UNITED STATES PATENT AND TRADEMARK OFFICE

_____

BEFORE THE PATENT TRIAL AND APPEAL BOARD

_____

FORTINET, INC.
Petitioner
v.
SOPHOS, LLC
Patent Owner

_____

Case No. IPR2015-00618
Patent 8,261,344

_____

VIDEOTAPED DEPOSITION OF SETH NIELSON, Ph.D.

Baltimore, Maryland
Wednesday, October 21, 2015

Reported by: John L. Harmonson, RPR
Job No. 98827

October 21, 2015

8:49 a.m.

Videotaped Deposition of SETH NIELSON, Ph.D., held at the offices of DLA Piper LLP, 6225 Smith Avenue, Baltimore, Maryland, pursuant to Notice, before John L. Harmonson, a Registered Professional Reporter and Notary Public of the State of Maryland, who officiated in administering the oath to the witness.

A P P E A R A N C E S

On Behalf of the Petitioner:

QUINN EMANUEL URQUHART & SULLIVAN
50 California Street
San Francisco, CA 94111
BY: JASON LIU, ESQ.

On Behalf of Patent Owner:

DLA PIPER
11911 Freedom Drive
Reston, VA 20190
BY: NICHOLAS PANNO, ESQ.
JAMES HEINTZ, ESQ.

ALSO PRESENT:

KRISHNA SHARMA, Legal Video Specialist

S. NIELSON

P R O C E E D I N G S

8:49 a.m.

THE VIDEOGRAPHER: Good morning. This is the start of media labeled number 1 in the videotaped deposition of Mr. Nielson, taken in the matter of Fortinet, Inc. v. Sophos, Inc., in the United States Patent and Trademark Office, before the Patent Trial and Appeal Board, IPR Number IPR2015-00618.

This deposition is being held at 6225 Smith Avenue, Baltimore, Maryland, on October 21, 2015, and the time on the video monitor is 8:48 a.m.

My name is Krishna Sharma from TSG Reporting, Inc., and I'm the legal video specialist. The court reporter today is Mr. John Harmonson, also in association with TSG Reporting, Inc.

Will counsel please identify yourselves for the record, and after that our court reporter will swear in the witness

S. NIELSON

and we can begin.

(Counsel placed their appearances on the video record.)

* * *

Whereupon,

SETH J. NIELSON, Ph.D.,

after having been first duly sworn or affirmed, was examined and did testify under oath as follows:

EXAMINATION

BY MR. PANNO:

Q. All right. Good morning.

A. Good morning.

Q. Would you please state your name for the record?

A. Seth James Nielson.

Q. And you are aware that your testimony today is under oath?

A. Yes.

Q. Okay. Have you ever been deposed before?

A. Yes.

Q. In any IPR proceedings?

A. No.

S. NIELSON

Q. Any other patent --

A. Yes.

Q. -- proceedings? Okay. Great.

So you probably are aware, then, of these ground rules that I'm just going to lay out real quickly right now. But I would like to ask that when I ask a question that requires or suggests a yes or no answer from you that you answer yes or no rather than "uh-huh" or "huh-uh" or head nod or something, just to help out the court reporter. Do you understand?

A. Yes.

Q. And we'll try not to talk over one another. So if I ask a question, please wait for me to finish to begin talking and I will do the same, wait for you to finish your answer before I ask again.

Do you understand that?

A. Yes.

Q. Okay. If I ask a question and you answer it, I assume you understood the question. If you don't understand the question, please ask me to clarify or rephrase. Is that all right?

A. Yes.

S. NIELSON

Q. Okay. And Mr. Liu may object at some points. If he does, are you still required to answer the question unless he explicitly instructs you otherwise.

Do you understand that?

A. Yes.

Q. Okay. So I know you're a little under the weather. Do you have any -- I trust that won't affect your ability to testify fully and truthfully today?

A. No.

Q. And you're not under the influence of any medications or anythings that might affect your ability to testify?

A. No.

Q. Any other ailments, physical, mental, anything going on that might affect your ability to testify?

A. No.

Q. Okay, great.

So in preparation for this deposition, did you meet with counsel prior to the deposition?

A. Yes.

S. NIELSON

MR. LIU: Objection; privileged.

I'll caution the witness not to reveal any attorney-client privileged information.

BY MR. PANNO:

Q. Right. I won't ask what you spoke about, just whether you met with counsel.

A. Yes.

Q. Great.

And did you discuss it with anyone -- discuss the deposition with anyone else before coming?

A. No.

Q. Okay. And did you review any documents to prepare yourself for this deposition?

A. Yes.

Q. What documents?

A. The declaration that I submitted. The patent that is the subject of the IPR. The petition. The institution decision -- am I calling that by its right name?

Q. Sure.

A. And the prior art patents.

Q. Okay. And did you do anything else to

prepare for the deposition?

A. No.

MR. PANNO: Okay. So I guess to get started I would like to ask the court reporter to hand the witness a copy of marked Exhibit 1001.

(Fortinet Exhibit 1001 marked for identification and attached hereto.)

BY MR. PANNO:

Q. Dr. Nielson, do you recognize this document?

A. It appears to be the '344 patent that is the subject of this IPR.

Q. Okay. And I will refer to this as the '344 patent for this proceeding. Is that all right?

A. Sure.

Q. Okay, great.

MR. PANNO: And then I'll also at this time like to ask the court reporter to give the witness a copy marked Exhibit 1007.

(Fortinet Exhibit 1007 marked for identification and attached hereto.)

BY MR. PANNO:

S. NIELSON

Q. Dr. Nielson, do you recognize this document?

A. Yes. This appears to be the declaration I submitted for this IPR, including Appendix A and B.

Q. All right. And so we'll just refer to this as your declaration. Is that all right?

A. Yes.

Q. Okay. So, Dr. Nielson, do you consider yourself an expert in any field?

A. Yes.

Q. What field or fields?

MR. LIU: Objection; vague.

BY MR. PANNO:

Q. In what field are you an expert?

A. Computer science generally. More specifically, computer security but also programming languages, software development and architecture.

Q. All right. Could you describe the basis of your expertise in computer security?

A. Yes. I have a Ph.D. from Rice University. My research area was security. In addition to my academic training therein, I was

also working as a consultant during my Ph.D. for Independent Security Evaluators, creating secure software libraries for a client and evaluating software for security vulnerabilities.

After I graduated with my Ph.D., I went to work for Independent Security Evaluators full time. As part of that process I -- excuse me. As part of my employment, I continued to develop applications that dealt with cryptography, hardware accelerated cryptography, cryptography on the file system.

I also worked with -- it's a system for finding errors in code that could lead to a security problem. And I also did evaluations where people submitted systems to be evaluated to see if there were vulnerabilities or flaws therein.

After ISC split and I was with the Harbor Labs people that split off from that, I have done additional security evaluations for major medical device manufacturers. I'm in the middle of a review right now of -- a-year long review for a major medical device manufacturer where I am evaluating their devices which are in

hospitals for risks to attackers or malware.

I've also in my work as a consultant reviewed source code dealing with antivirus software and developed kind of like an inert virus but to test it against triggering on antivirus software.

I guess one other thing I left out during my Ph.D. time period, I also did an internship at Google in their security group.

Q. All right. And then I also noticed you listed on pages 50 to 51 of your declaration three patents that you're an inventor -- on which you're an inventor.

A. Yeah.

Q. Can you describe the subject matter of those three patents just generally?

A. Yes. So these patents -- what I know about them is a little bit tangential because the work I did was at the end of my time at ISE, and after I left they took the work that I had been contributing to and submitted patents.

But my understanding is that it is based on some design work and implementation that I did before I left ISE where we developed a

system for a secure communication library that could tolerate loss of trust in a certificate authority.

Q. Okay. And you mention on page 50 you teach network security classes at I believe Johns Hopkins?

A. Yes.

Q. What is the subject matter of those courses, course or courses that you teach?

A. The course is network security. It's a graduate level class. I teach it primarily to students in what's called the MSSI program. It's masters of science and security informatics.

The class is a fairly broad course. We cover topics even as wide as, say, user psychology when it comes to passwords and some of those kind of issues, all the way down into cryptographic protocols at the network level. If it has to do with computer security and involves networks, we try to cover it.

Q. Okay. So I would like to move on and direct your attention to Paragraph 21 of your declaration. And I'll give you a minute.

A. Yes, I'm there.

Q. Okay, great.

So Paragraph 21 includes your definition of a person having ordinary skill in the art. Could you please just review that definition and let me know when you've done that.

A. Yes.

Q. Okay. So what do you mean by a person of ordinary skill in the art relevant to the '344 patent?

MR. LIU: Objection; vague.

BY MR. PANNO:

Q. What do you mean by "relevant" in that statement?

MR. LIU: Same objection.

BY MR. PANNO:

Q. Relevant to the '344 patent.

A. Well, somebody who would be a person of ordinary skill in the art in terms of being able to understand and implement, I guess is the word, the patent.

Q. Okay. And what do you mean by "at the time the '344 patent was effectively filed"?

A. Well, I understand that the relevant time period is June 30, 2006.

Q. So that statement could be summarized as a person who is able to understand and implement the '344 patent at the time of the filing date?

A. I don't -- so let me -- let me say that I understand that a person of ordinary skill in the art is a legal concept, and I have applied the standard as given to me by counsel. So I'm not commenting on the legal aspect of it.

Q. Okay.

A. My understanding of a person of ordinary skill in the art is it's representative of some kind of a hypothetical standard that you use in interpreting how somebody would read the patent and have knowledge to understand what it's saying.

So to my understanding of that standard, then, yes, at the time, June 30, 2006, this is where I place that standard.

Q. Okay. And you say that that person would have a master's degree in electrical engineering, computer engineering or computer science or a bachelor's in one of those fields with at least two years experience working for

network security, hardware, software.

How did you arrive at that definition?

A. Using my experience in the field and looking over what concepts are covered in the '344 patent, I gave my opinion as it's presented here about the approximate level you would need in terms of experience and education to be somebody of ordinary skill in approaching this patent.

Q. All right. So someone with a master's degree in electrical engineering, let's just say, to pick one example, would be someone who could read the '344 patent and understand it?

A. As a general matter, I would say yes.

Q. Would they be able to implement it?

A. Yes.

Q. So -- Okay. So without having taken any special courses on computer security, is just someone with a master's in EE?

A. The reason that I think a master's degree makes such a difference is because usually with a master's degree we are expecting students to be more mentally flexible and have some extra breadth even outside of their specific subject

area.

Q. All right. How far outside one's specific subject area would you expect that breadth to extend?

A. I don't know that I can give a perfect bounds, but I do think that I could take a master student at Johns Hopkins who has completed his master's degree in electrical engineering, hand him this patent, and have him understand what it's laying out.

Q. And understand how to implement it?

A. Well, usually when you implement something, it's implemented as a team. It's implemented in a group. I would believe he could be a contributing member of that implementation group.

Q. In what way could he contribute?

MR. LIU: Objection; vague.

THE WITNESS: I think that he would be able to -- I think there wouldn't be anything that he wouldn't be able to participate with in that group.

BY MR. PANNO:

Q. Would it be as a member of the team

who is being guided by someone who understands the patent more fully?

MR. LIU: Objection; vague. Incomplete hypothetical.

THE WITNESS: Again, given that I'm working with a hypothetical person that's -- I don't know quite how to answer that. I would say that in general a person with a master's degree in electrical engineering would understand the patent and would be able to implement it.

BY MR. PANNO:

Q. What about someone whose focus in electrical engineering is something like power systems or control systems or something that's kind of far afield from the more software-oriented disciplines within electrical engineering?

MR. LIU: Objection; incomplete hypothetical. Compound question.

THE WITNESS: I can certainly imagine there would be some person with a master's degree in electrical engineering out there somewhere who could not do this. But I do

give this as a general guideline for what I think this hypothetical person would be like.

BY MR. PANNO:

Q. Okay. And then just to kind of finish the thought, in Paragraph 22, the following paragraph --

A. Yes.

Q. -- you state that you have these capabilities of a person of ordinary skill at the time the patent was effectively filed, correct?

A. Yes.

Q. Okay. So that was June 30, 2006?

A. Yes.

Q. So we discussed your background somewhat, but just to kind of pin that down, what was your -- what were you doing professionally and -- you know, with your education in June of 2006?

A. So in June of 2006, I was still in my Ph.D. program, and I was working as a consultant at Independent Security Evaluators.

Q. All right. And this was -- so you had completed a bachelor's and master's in --

A. Computer science.

Q. -- computer science.

And your work at that time would have been -- could you describe just basically the work you had completed up to that time?

A. Up to 2006?

Q. Yeah, June 2006.

A. Just to clarify, you want me to repeat what my professional background was --

Q. Only the portions that are -- that you had -- I believe, correct me if I'm wrong, that your previous recitation of your professional background was sort of a complete summary of your experience.

A. It was a summary of my experience.

Q. Well, a summary of your experience up till today?

A. Yes.

Q. Okay. So can you just point out what parts of that were -- you had completed by June 2006?

A. Sure. So I had completed my master's degree in computer science in 2004.

Q. Okay.

A. I graduated with my bachelor's degree in 2000. I worked for three and a half years in industry before -- well, some of it overlapped with my master's degree but before and overlapping with it.

I had started my Ph.D. program in 2004. In 2005 I had already published a paper about computer security. I had an internship with Google, and in 2006 I was already one year into consulting for Independent Security Evaluators.

Q. All right. So clearly you may have been someone who was a person of ordinary skill or maybe someone with more skill than that. But would you say that a person of ordinary skill, by your definition, would have had the same sort of experience that you had at that point, the same level of experience?

A. Well, I think that -- I think that the definition I gave here in Paragraph 21 was a master's degree. So obviously I was already a few years past that.

Q. Okay. Great.

So I think we can move on to -- Let's

go to Paragraph 42 of your declaration, please. And if you would like to review that, please feel free to do so.

A. Yes, I've reviewed it.

Q. Okay. So you say in this paragraph that genes is not a term of art ordinarily used in computer science, correct?

A. Correct.

Q. All right. How did you come to that conclusion?

A. When I wrote this sentence, I could have been more clear. I should have probably said computer security rather than computer science. I can think of at least one place in computer science where they do use the term.

Q. Okay. Could you explain where in computer science they use the term?

A. So the term is used in artificial intelligence. There is a concept of a learning algorithm that uses genes.

So I've written a program like this myself. You create a mathematical structure. It represents some type of computation but it's called a gene because there are different pieces

of it, just like in your DNA. And then when you're solving a problem, you create a large number of these with variations and then kind of analogous to evolution, you take these genes and you kind of crossbreed them. You take these mathematical structures, you split them up and recombine them and see if it moves you closer to a better solution.

Q. But that is -- that is -- as you laid it out, however, you have not encountered the term "gene" for the use of -- in the field of computer security?

A. Correct.

Q. And so your assertion that it's not a term of art ordinarily used in computer security is based upon your experience in the field?

A. Yes.

Q. Okay. And you state also in Paragraph 42 that you were asked to apply the construction of a piece of functionality or a property of a program to the term gene. Is that correct?

A. Correct.

Q. All right. You were asked -- Okay.

So is that construction consistent with your reading of the '344 patent?

A. What do you mean, "consistent"?

Q. So you were asked to apply the term. Was that by counsel?

A. Yes.

Q. Okay. So you've also reviewed the '344 patent yourself, correct?

A. Yes.

Q. So that definition -- You agree with that definition based upon your reading of the '344 patent?

MR. LIU: Objection; assumes facts not in evidence.

THE WITNESS: I have previously reviewed this patent for a claim construction in a different proceeding, and I submitted an opinion in that one with a slightly different claim construction. And that was my opinion then.

For this particular analysis that I'm doing here, which is an IPR, it's a little bit different with broadest reasonable interpretation, as I understand that term

from counsel, I've been asked to apply this. I don't necessarily see anything -- I don't -- I haven't seen anything with it that I think is wrong with that for this particular matter.

BY MR. PANNO:

Q. Okay. So could you find -- and please feel free to look over the '344 patent. But could you point to a point -- a teaching in the '344 patent that supports that construction, please.

MR. LIU: Objection; foundation.

THE WITNESS: Okay. In the '344 patent, in Column 5, line 32, it says, "A gene is a piece of functionality or property of a program."

BY MR. PANNO:

Q. So now we're here. I'll have to keep this open. But in the following paragraph in your declaration you discuss a piece of functionality or property -- well, you discuss a piece of functionality of a program. And you can review your paragraph.

A. Which, 43?

Q. 43, yes.

A. Okay.

Q. So what would a person of ordinary skill understand the term "piece of functionality of a program" to mean?

A. Well, I think that the next line in the '344 patent kind of explains it a little further. I think a person of ordinary skill in the art reading it would then read, "each piece of functionality is described using sequences of APIs and strings," and that that along with the examples given in 6 would help them to understand what that means.

Q. In 6?

A. I'm sorry, Column 6.

Q. Column 6, thank you.

Okay. So could you describe what a person of ordinary skill in the art would understand a sequence of APIs to be?

A. Well, so the examples that are given in Column 6 are -- obviously, they're examples. But the examples given here are things like system calls, for example. That would be one type of sequence of APIs.

S. NIELSON

Q. Okay. And the string?

A. String is well known in the art. It's just text.

Q. Okay. So the text -- what would the text -- What purpose does the text serve?

MR. LIU: Objection; foundation. Vague.

THE WITNESS: I'm not sure because -- Let me ask a clarifying question here.

BY MR. PANNO:

Q. Absolutely.

A. So with a gene being a piece of functionality or property of a program, and then pieces of functionality described using APIs and strings, you're asking what purpose --

Q. Well, what purpose -- Yeah. What purpose would a string serve in that context? You know, examples.

MR. LIU: Same objection.

THE WITNESS: If I understand your question correctly, the strings here -- I mean, we're talking about the genes, right?

BY MR. PANNO:

Q. Right.

A. The genes are used for classification.

Q. Right.

A. So these strings would be used in classifying the genes as they're defined.

Q. Okay. And the APIs would be also used in classifying the genes?

A. Let me rephrase that. So the gene is described in this example using sequences of APIs and strings. So the gene would be some sequence of APIs and strings.

Q. Okay.

A. And then the gene is compared against know -- you know, is checked for known genes in the library.

Q. So the gene -- the gene, which is a sequence of APIs and strings, describes a piece of functionality of the program; is that correct?

A. A piece of functionality or property of the program.

Q. Okay. So in the event that it describes a piece of functionality, what would that piece of function -- what would a piece of functionality be that's being described?

MR. LIU: Objection; vague.

S. NIELSON

Incomplete hypothetical.

THE WITNESS: I think so. For example, in Column 6, it shows one on line 40, and it gives this gene a name and calls it exec, which would be short for execute. And it executes other programs and includes as an example the following function, or I would call it an API or a system call, create process A.

BY MR. PANNO:

Q. Okay. And so these other -- and these other examples would be similar. So run key, which starts on line 35?

A. Yes.

Q. And then the following functions are -- how would you describe, for example, the following -- like the function RegCreateKeyExA and RegSetValueExA, are they also --

A. Those would be APIs and system calls I think for both of them in this example.

Q. Okay.

A. And then above them is a string.

Q. The line 37?

A. Yes.

Q. Okay. Is a string?

A. Yes.

Q. Okay. And then as you noted, a gene can also be a property of a program. What would someone of ordinary skill in the art understand property to mean?

A. Well, there's not a lot of discussion about it in the patent. It does mention in Column 5 at line 4 that when the properties are extracted, that can include file size or name. But those wouldn't fit into what we're talking about here.

Q. Right. But just as an understanding of the term, what would someone of ordinary skill in the art understand the term "property" to mean?

A. Sure. So I think somebody of ordinary skill in the art, when they would look at this, would look at a sequence of APIs and strings and they could either say well, that's going to be a piece of functionality as in, you know, it performs something or it's a property as in it defines something about that program, or it could be both.

S. NIELSON

Q. Okay. So how would a -- the sequence of APIs and strings illustrate a property of a program?

A. Right. So, for example, if you have a sequence of APIs that have to do with network access, then you can say that that program has the property of being a network accessing program, which matters in a computer security term because that may be an unauthorized or suspicious aspect no matter how it's doing it.

Q. Okay. So I guess would you mind using as an example -- let's talk the gene CopySys which is in Column 6 starting at line 28.

A. Sure.

Q. So that -- So this gene includes the functions listed from lines 30 to 34.

A. Yes.

Q. So could you -- I guess could you step through those functions and explain what they describe about the gene?

MR. LIU: Objection; calls for a narrative.

THE WITNESS: Well, so the names of the functions here -- I'll just read through

it starting in line 30.

GetSystemDirectoryA. GetModuleHandleA. GetModuleFileNameA. The string .exe. And CopyFileA.

So this CopySys gene I would think somebody of ordinary skill in the art would see that primarily as functionality. But I can imagine somebody maybe seeing GetSystemDirectoryA all by itself as a gene that is a system access or something like that.

BY MR. PANNO:

Q. So you say GetSystemDirectoryA as a gene.

A. It could be.

Q. But in this example, perhaps this -- you mean to say GetSystemDirectoryA by itself could be a gene?

A. Sure.

Q. All right. This example includes GetSystemDirectoryA as well as other APIs and strings, that could also be a gene?

A. Yes.

Q. And the functionality of this gene is

affected by each of the functions listed in this column, in this section of the disclosure?

A. Yes.

MR. LIU: Objection; vague.

Go ahead.

THE WITNESS: Yes. So again, this one describes the gene. It describes the functionality. It's representing, and it says -- copies itself to a system folder. And so it's identifying multiple steps that would be involved in a copy to the system folder.

BY MR. PANNO:

Q. Okay. And then in Paragraph 43 of your declaration, if you would turn back there, you also discuss functional blocks.

A. Yes.

Q. Okay. So what would one of ordinary skill in the art understand the term "functional block" to mean?

A. Well, again pulling from the '344 patent, somebody reading the '344 patent in Column 5 starting at line -- I can't tell if that's 16 or 17. I think it's --

S. NIELSON

Q. I think let's call that --

A. I think it's 17. "At step 250, the function blocks are extracted from the file. These functional blocks include sequences of application program interface...calls, string references, et cetera, which illustrate the function and execution flow of the program."

Q. Okay. So a functional block, then, illustrates a function and execution for the program, correct? That is its purpose? It includes -- I'm sorry, go ahead.

A. I believe that that is a construed term.

Q. It is. Actually, Paragraph 51 of your declaration I believe recites the construction.

A. Yes, correct.

Q. Right. So yeah, okay, this is a segment of the program that illustrates the function and execution flow of the program according to this construction?

A. Yes.

Q. So that's how one of ordinary skill in the art would understand the term?

A. I think that is a very reasonable way

for one of ordinary skill in the art to understand that term.

Q. Okay. So in this context, what would a person of ordinary skill in the art understand the term "program" to mean?

A. Program?

Q. Program, yeah.

A. I would say any piece of software.

Q. Okay. And what about the function of the program?

A. The way it behaves, what it does.

Q. And what about execution flow of the program?

A. Well, so functional block is actually a term that I have seen used in the art, and it's completely in line with what we're describing here, the execution flow and function of the program.

But let's say that you've got like something that's not binary, like a script. Even with a binary it's true, though, that a function block could be a chunk of code that has recognizable boundaries, a loop for example, or a call to a -- even a binary has the equivalent of

function calls. And so you could identify those chunks as a functional block, and that would definitely illustrate the flow of the program.

Q. Okay. So could you elaborate, how would a chunk like that illustrate the flow of the program?

A. Well, for example, if you've got a loop, a loop changes the program flow. A branch instruction in assembly, the binary, would change the flow. Function call changes the flow.

Q. Okay. All right. Let's see where should we go next.

MR. PANNO: Actually, why don't we take a quick break right now. It's a little before the hour but...

THE VIDEOGRAPHER: We're now off record at 9:27.

(Recess taken.)

THE VIDEOGRAPHER: We are now back on the record at 9:34.

MR. PANNO: So I would like to ask the court reporter to mark this and give it to the witness. This is the petition for inter partes review for this case.

S. NIELSON

(Exhibit 2001 marked for identification and attached hereto.)

BY MR. PANNO:

Q. Dr. Nielson, do you recognize this document?

A. Yes. It appears to be the petition for this IPR.

Q. Okay. Turning quickly back to your declaration, in Paragraph 57 you state that "In connection with the above, I have reviewed, had input into, and endorse as set forth fully herein the claim charts in the accompanying petition showing that each element of Claims 1 and 2 are disclosed by Bodorin."

A. Yes.

Q. That you -- Excuse me.

MR. PANNO: I would like to also ask the court reporter to give the witness Exhibit 1003.

(Fortinet Exhibit 1003 marked for identification and attached hereto.)

BY MR. PANNO:

Q. Dr. Nielson, do you recognize this?

A. Yes. This appears to be the Bodorin

reference that I cite to in my declaration.

Q. Okay. And let's refer to this as Bodorin for this proceeding. Is that all right?

A. Yes.

Q. Okay. Could you briefly describe the subject matter of this patent?

MR. LIU: Objection; vague.

THE WITNESS: As a brief overview, I would say that it's a system for classifying malware.

BY MR. PANNO:

Q. All right. Could you elaborate a little further? How does this system classify malware?

MR. LIU: Objection; vague.

THE WITNESS: Well, in some of the embodiments disclosed within the patent, it starts by doing a dynamic evaluation, is I think the words it uses, and the art would probably either just call it emulation or -- it could be a complete execution, either way. But it starts to dynamically evaluate the piece of software.

And as it is examining how the

software is behaving, which it does by looking at more or less API calls, system calls, then it selects from those ones that are interesting, is the word Bodorin uses, to create a signature, and then it compares that to a library.

BY MR. PANNO:

Q. All right. So you mentioned that it looks at API calls, system calls, and selects interesting ones from among those.

A. Yes.

Q. All right. So how does it look at the API call?

A. Well, because it's doing this dynamic evaluation, it's basically seeing every API call as they happen.

Q. Could you explain how it sees the API calls as they happen?

MR. LIU: Objection; asked and answered.

THE WITNESS: Well, if you have an emulator, for example, the emulator is receiving the calls directly. Bodorin suggests in one embodiment that you would

have some kind of secure virtual machine. You would still have the same basic access to the API calls as they happen.

BY MR. PANNO:

Q. Could you point out where in Bodorin it talks about that?

A. Yes. Column 4, starting in line 39, it says: "Each dynamic behavior evaluation module, such as dynamic behavior evaluation module 204, represents a virtual environment, sometimes called a sandbox, in which the code module 212 may be 'executed.' To the code module 212, the dynamic behavior evaluation module 204 appears as a complete, functional computer system in which the code module may be executed. By using a virtual environment in which the code module 212 may operate, the code module may be executed such that its dynamic behaviors may be evaluated and recorded, while at the same time any destructive behaviors exhibited by the code module are confined to the virtual environment."

Q. All right. So how does the -- Excuse me.

How does the -- How are the dynamic

behaviors evaluated and recorded?

A. So as you continue reading into the next paragraph, it says: "As the code module 212 executes within the dynamic behavior evaluation module 204, the dynamic behavior evaluation module records 'interesting' behaviors exhibited by the code module. Interesting behaviors are those which a user or implementer of the malware detection system 200 has identified as interesting..."

And so if you look over at Table A, it lists the sorts of behaviors, as an example only it says, that may be interesting. These would be examples of system calls.

Q. All right. So this table is a table of behaviors that are observed; is that correct?

A. Well, so -- again, these are -- these are examples of system calls. You'll notice there are some even that we saw in the '344 patent, like GetModuleFileNameA. These are a standard call for Windows, for example.

And so what you've got is you have a virtual environment where it's executing, and because you control the virtual environment, you

see every system call. So as system calls are going by, when you see one that is interesting, you pull it out and record it.

Q. All right. You mentioned that this is one way that Bodorin teaches observing the interesting behaviors.

A. Yes.

Q. Does Bodorin disclose observing interesting behaviors in any other way, or is this the only embodiment?

MR. LIU: Objection; vague. Assumes facts not in evidence.

THE WITNESS: So Bodorin talks about a dynamic behavior evaluation module. And so I'm not aware of any way other than dynamic behavior evaluation. But this particular example given here -- I'm sorry. This particular example given here would actually cover -- it's a -- a sandbox is a broad term. Let me show you what I'm talking about.

If you look in Column 4 at line 30, it points out that there are different types of code modules that can be evaluated. And so

it mentions as an example a Microsoft Windows executable. And what it's showing in Table A would be the sort of things that you would find in a Microsoft Windows executable. But then it lists a Microsoft.net executable and a Java applet or application. Those wouldn't have the same type of API calls. Java programs use what's known as a Java virtual machine, and their access to the system looks different.

So in any of these examples you can have a virtual machine -- I'm sorry, a sandbox, a virtual environment in which you can observe the behaviors dynamically because the software has to interact with the system somewhere. And as long as you can see how it's interacting with the system, then you can see its execution.

BY MR. PANNO:

Q. Okay. So that's an example there. So you have Windows executable which has one type of command -- one command for one action. You have a Java applet or application that has a different command for the same action; is that correct?

S. NIELSON

MR. LIU: Objection; compound. Mischaracterizes the testimony.

THE WITNESS: So the API calls that you would see on Windows would even be different on maybe a different operating system, but definitely different to a Java application. But you would be able to -- in all cases you would have something like open a file, write to a file, execute a file; they might just have different names and formats.

BY MR. PANNO:

Q. Okay. So would Bodorin be able to detect open a file for different languages having different commands for open a file?

MR. LIU: Objection; assumes facts not in evidence. Vague.

THE WITNESS: Well, I would assume that one of skill in the art, reading the Bodorin patent, would be able to take the example described for the Windows executable and be able to do that exact same thing for a Java applet.

BY MR. PANNO:

S. NIELSON

Q. Okay. And so if the user considered open file to be an interesting behavior --

A. Yes.

Q. -- the user would be able to configure a system based on the teachings of Bodorin to identify open a file in a Java program, a Java applet?

A. I believe so, yes.

Q. And does Bodorin explain what the behavior signature for Java or anything other than the example of Table A would look like?

MR. LIU: Objection; vague.

THE WITNESS: So Table A isn't actually the behavior signature. Table A gives examples of APIs that may be interesting. It does give an example of what a behavior signature might look like in Tables 5A and 5B.

BY MR. PANNO:

Q. Okay. So the signature, for example, of Figure 5A --

A. Yes.

Q. -- could you point out what part of the table illustrates the interesting behavior in

Figure 5A?

A. Yes. So what you have is -- You see the columns there?

Q. Yes.

A. So on the left-handmost column, it refers to it as a behavior token but it maps to one of these APIs. And so you can see there Internet read file or Internet open URL and write file. Those correspond to the interesting behaviors.

Q. Okay. So if Bodorin is used with a Java program that has a command that causes a file to be written, would the behavior signature look like the third behavior signature in the third line in Figure 5A?

MR. LIU: Objection; incomplete hypothetical.

THE WITNESS: I think so. The text in Bodorin, if we can turn back to the text for a minute, it talks about these figures in Column 7. So -- Excuse me.

So what it says here is it says that these figures starting in line 16, "are block diagrams illustrating exemplary

behavior signatures of hypothetical code modules, such as the behavior signature 210 described above."

Let me skip down to 23. "As illustrated in both behavior signature 500 and behavior signature 512, each behavior includes three elements; a behavior token, such as behavior token 502..." And let me just stop there.

So the behavior token was what we were looking at before, the InternetOpenUrl, Internet read file and write file.

And here's the interesting part. If we skip down to 32, it says: "A behavior token, such as behavior token 502, is used to identify the particular 'interesting' behavior recorded by the dynamic behavior evaluation module..."

So what this says is that your dynamic evaluation module would find the stuff like what's described in Table A and it would say oh, look, I have a Windows write file API command. And then when it goes into the Figure 5A, it gets replaced by a more

generic behavior token called write file.

So in answer to your original question, yes, I would imagine a Java signature could conceivably look just like this because whatever Java's call for write file might look like, it could be replaced with the write file behavior token that you see here.

BY MR. PANNO:

Q. Okay. So I would like to just really quickly step through the claim charts in the petition that I handed you --

A. Sure.

Q. -- as you have indicated that you have endorsed these.

A. Yes.

Q. Okay. So let's see. We're on page 23 of the petition. We don't need to do the entire chart.

A. All right.

Q. Page 23 includes claim element 1B.

A. Yes.

Q. Identifying at least one functional block and at least one property of the software

and cites to Bodorin.

A. Yes.

Q. Okay. So could you please identify the functional block in this portion of Bodorin?

A. Yes. So I'm sorry, which exhibit is it again? I closed it.

Q. Bodorin?

A. Bodorin.

Q. Bodorin is Exhibit 1003.

A. Got it. Thank you.

So for Bodorin, the functional block which again is defined to be for this matter a segment of the program that illustrates the function and execution flow of the program would be the complete set of APIs that are observed by the dynamic behavior module.

Q. So where is that in this passage?

A. So it says: "As the code module 212 executes within the dynamic behavior evaluation module 204, the dynamic behavior evaluation module records 'interesting' behaviors exhibited by the code module."

So the entire set of behaviors, just not the interesting ones, are the functional

block.

Q. Okay. And could you identify the property in this passage?

A. So again, as we were discussing with the '344, you could identify a property from an API as well.

Q. And could you point out in this passage what the teaching is that covers that?

A. So in this passage I have the citation from Bodorin, but I explain more about the individual pieces in my declaration.

Q. Okay, sure.

But which citation in Bodorin would your explanation apply to?

A. So to these behaviors that are exhibited by the code module.

Q. So the interesting behaviors?

A. No. The behaviors.

Q. The behaviors?

A. Yes.

Q. Okay, thank you.

So why don't we turn to element 1C, which is pages 24 to 25.

A. Yes.

Q. And same thing. I would just like to ask you, and you can review it, of course, before I do, but to identify a couple of terms in the Bodorin reference.

So one or more genes. Can you point to what in this citation describes the one or more genes?

A. Yes. So the interesting behavior or the one that's highlighted here is API function would be -- Well, so the call and its parameters would be the gene.

Q. The API function call?

A. The interesting behavior, yes.

Q. All right. Could you read directly which portion of this passage is the gene?

A. So do you want me to just read the whole --

Q. Just read the portion that is the gene only.

A. "...and wherein each dynamic behavior evaluation module records interesting API function calls that the code module makes as it is executed, wherein the interesting API function calls are specified by a user and comprise a

portion of all API function calls...that the code module makes..."

Q. Okay. And Element C also includes -- teaches that the genes describe one or more of the at least one functional block and at least one property.

Can you specify which portion of this passage is directed to the at least one functional block described by the gene?

A. Yes. So the part where it says "not all API function calls," the interesting behaviors are a subset of that. So the entire set is the functional block.

Q. Okay. And same thing for property, the at least one property. Can you point out in here?

A. Well, first of all, the construed meaning of gene is that it needs to describe at least one functional block or one property. And so to meet this limitation, I don't have to have the gene describing a property.

However, with that said, because these -- as I pointed out, these particular APIs would also represent properties, I think that it

applies as well.

Q. Okay. And then the claim, Element C also requires that the genes ascribe one or more of at least one functional block and at least one property of the software as a sequence of APIs and strings. So it's just a little redundant, but could you point out what in this passage is the sequence of APIs?

A. It refers to Figure 5A and 5B, and if you look at each line, you will have an API call and strings. So you have a sequence of APIs and strings.

Q. Okay. Is each line, then, a sequence of APIs and strings?

A. Yes.

Q. Okay. And -- Okay. I think that's good for Paragraph 1C.

I would like to jump around again. If you would turn back to your declaration briefly.

A. Sure.

Q. Paragraph 94. So again in Paragraph 94, you state that you have reviewed, had input into and endorse as if set forth fully herein the claim charts in the petition related

to obviousness of Claim 16 and 17 over Bodorin in view of Kissel and Apap.

A. Yes.

MR. PANNO: So I would like to ask the court reporter to hand the witness Exhibit 1004.

(Fortinet Exhibit 1004 marked for identification and attached hereto.)

MR. PANNO: And also please hand the witness Exhibit 1006.

(Fortinet Exhibit 1006 marked for identification and attached hereto.)

BY MR. PANNO:

Q. All right. So do you recognize, Dr. Nielson, Exhibit 1004?

A. Yes. It appears to be the reference referred to as Kissel in my declaration.

Q. And we can refer to that as Kissel here today. Is that all right?

A. Yes.

Q. Okay. So could you briefly describe the subject matter of Kissel?

A. Kissel deals with tracking down malware found in e-mail. It can look at details

of the e-mail itself as well as features of script or other attachment.

Q. Okay. Could you describe in a little more detail how it does that?

A. So Kissel has this thing it calls a feature vector, and then once it's created the feature vector, it compares that against known vectors or signatures of known bad things.

Q. Okay. And then I think we can -- if you would take a look, please, at Exhibit 1006.

A. Yes.

Q. Would you tell me if you recognize this.

A. Yes. This appears to be the reference I referred to as Apap in my declaration.

Q. All right. We'll refer to this as Apap. Is that all right?

A. Yes.

Q. Okay. Could you, again, for this one same thing, describe the subject matter?

A. Yes. Apap is a disclosure dealing with what I would generically call intrusion detection. And it monitors a system for probabilistic anomalies to try and detect -- So

backing up.

When a person uses a computer, they have a pattern, and when an intruder, it is assumed that the intruder will have a different pattern. And so Apap is directed towards using a probabilistic model to detect when an intruder is using the system.

Q. Okay. So just sort of -- I know you touched on this in your declaration, but would you please describe what would motivate one of ordinary skill in the art to combine these three references, Bodorin, Kissel and Apap?

A. Sure. So first of all, I did an analysis on Kissel and Apap separately, which even though that's not instituted, that's relied on for combining with Bodorin. And so what I talked about with the motivation to combine Kissel and Apap is that Kissel already had a system for updating its library of bad signatures, so to speak.

Let me just point you to what I said in my report. So in Paragraph 85, when you have one of these detection systems, if something new comes along, of course you won't have a signature

for it, because a signature can only be something you've seen before.

So Kissel disclosed that if you come across something that doesn't match in the library but you now have determined that it's bad, you can add it into your library if it's crossed a certain threshold of badness. So that now if you see that in the future, you won't do a crossing of a threshold; you'll just say it's bad.

So Kissel also mentioned that you would -- Let me double-check and make sure I'm right on this. Yes, okay. So Kissel also explicitly mentions that it's going to check for false positives. But what Kissel didn't explicitly say is to check for false positives against a reference file.

But what Kissel does say within its reference, when it talks about checking for false positives -- Let me turn you to Kissel.

And it says, if I can find this without search.

Q. Take your time.

A. Yes. Kissel, Column 5, starting in

line 61 I think it is, if I'm getting that split right, it says: "The determination of a false positive can be accomplished by any conventional means..."

And then it lists a number of conventional means, but it does not mention reference files.

However, it was well known in the art that when creating signatures, you wanted to avoid false positives, and that there was suggestions even just within the art that you would compare it against a corpus of known good references, basically reference files.

And so I believe that one of skill in the art would be motivated to combine Apap, which discusses using reference files for reducing false positives, with this idea of any conventional means in Kissel.

Q. Okay. So that covers the combination of Kissel and Apap?

A. Yes. Yes.

Q. So why Bodorin?

A. Right. Okay. Right. So Bodorin has as part of its specification that it's important

to update the signature store. This is something I refer to in Paragraph 92.

So it's already motivating somebody who is reading Bodorin to look for ways to keep the security store updated because that's the only way to keep the system effective. In fact, Bodorin has a section in there already that discusses checking for a probabilistic score. Let me find that reference for you.

It maybe didn't use the word "probabilistic," but I think he uses the word "threshold." Let's see.

So in Column 6, the last paragraph starting at 58: "If there was not a complete match between the behavior signature 210 and the behavior signatures in the malware behavior signature store 208, at decision block 318, a further determination is made as to whether there was at least a partial match. If there was a partial match between the behavior signature 210 and a behavior signature in the malware behavior signature store 208, at block 320, the malware detection system reports that the evaluated code module 212 may be malware."

It talks about some mechanisms for doing that. But given that somebody reading Bodorin would be looking for ways to expand their storage system and given that Bodorin already is recognizing things outside of its system, I think it would have been motivation to combine with Kissel that takes those less than perfect matches and figures out how to add them into the system.

Q. All right. And then one would, moving on from there, look to Apap?

A. Yes.

Q. All right. So I just once again would like to, for ease of Claim 16, just step through a few of the chart entries.

And let's see. Let's try I think page 46 --

A. Sure.

Q. -- of the petition.

A. I'm there.

Q. All right. And excuse me. 16B. And some of these may be the same answers as they were for the other chart, but I would just like to go through them again if I may.

A. Yes.

Q. So again, 16 teaches identifying one or more genes, and I would like to ask, Dr. Nielson, if you would show me where in this cited passage the genes can be found.

A. So I'll point out the same sequence of text that we discussed earlier for genes. It's on the next page in the -- it's citing from Claim 1. "...and wherein each dynamic behavior evaluation module records interesting API function calls that the code module makes as it is executed."

So the genes would be the interesting API function calls which are called elsewhere in the patent interesting behaviors.

Q. All right. And functionality of the software, could you find that in this passage, please?

A. Sure. So I think we've already discussed that API function calls are defined functionality of the software.

Q. All right. And that is in here where?

A. Oh. Well, even the part I read mentioned API function calls.

Q. And then is property in this passage?

S. NIELSON

MR. LIU: Objection; asked and answered.

THE WITNESS: Sure. So the way I've referred to it before is that an API function call could also describe a property of a program.

BY MR. PANNO:

Q. All right. Then the sequence of APIs and strings.

MR. LIU: Objection; asked and answered.

THE WITNESS: So I see that Figure 5A and B are not cited in this element. But here in the first paragraph it talks about -- oh, where was it? Hold on a second.

Yes. Okay. So it mentions in -- starting about halfway through that paragraph it says Table A. "Table A includes an exemplary, representative list of 'interesting' behaviors, including parameters that are also considered interesting..."

Like we saw in the table, you get the

API call and parameters as strings, and those would represent a sequence of APIs and strings.

BY MR. PANNO:

Q. And actually this element talks about identifying one or more genes driving a functionality or property of the software. Could you just point out what in this text illustrates the "of the software"?

A. Yes. That would be code module 212.

Q. Code module 212 is the software?

A. Yes.

Q. Okay. And then we can jump, I guess, to Element 16D, which starts at the very bottom of 47 and continues on to 48, 49. And this element says testing a set of genes for false positives against one or more reference files using a processor.

A. Yes.

Q. So again, if you would, could you please identify where false positives is found in this -- in these citations, I suppose.

A. So in the second citation from Exhibit 1004, the bolded section discusses

optimizing the system using false positive thresholds appropriate for the given computing environment.

Q. Okay. And the reference files, could you show me where those can be found?

A. In Exhibit 1006, so the anomaly detection algorithm is trained over the normal data. So the normal data would represent your corpus of reference material, reference files.

Q. Okay. Could you just please explain how the normal data represents a reference file?

A. Sure. So at the very least the data is going to be stored in a file. And so that would be a reference file of parameters and other pieces of information.

Another example would be that because we're talking about an intrusion system, you have to have -- your normal data would include different elements about the system in terms of activity, but also in terms of files. And so you would have reference files be part of normal data.

Q. So could you expand on that a little bit? I just want to make sure I understand what

normal data means. Just normal.

A. Sure. It's actually tied to normalization in statistics. The normal -- so you have to have -- you have to know what a baseline is in order to be able to detect bad behavior. You have to -- you have to know -- because all activity could look abnormal under certain circumstances. Every file could look like a bad file given, you know, a certain set of criteria.

And so the idea here of having normal data or normalized data is that you're establishing what your baseline for the system is.

Q. And that is something that would be understood by one of ordinary skill?

A. Yes.

Q. And could you -- and you can turn to Apap if you need to. Show me how that's taught by Apap. That's Exhibit 1006.

A. Yeah, I've got it here.

Sure. So here on Column 17 -- So here in Column 17, starting in 43, it talks about a system that does not use a registry like Windows

does. And so it talks about in this case having a file system sensor. And so accesses to the file system provides an audit source of data. And then you would build your normal baseline model from monitoring the file system accesses.

Q. Okay. Does it explain how to build this normal baseline model?

A. The patent covers the statistics of this system pretty extensively.

Q. So one of ordinary skill would be able to build a normal baseline model from the teachings of Apap?

A. I believe so, yes.

Q. Okay. So going back to our chart once more.

A. Yes.

Q. So again, 16D is testing a set of genes for false positives against one or more reference files using a processor.

A. Yes.

Q. So could you just please point out the portion of these passages that teaches using a processor?

A. It would be inherent.

Q. Could you explain how?

A. You can't run software without a processor.

Q. But can you test a set of genes that you've already built against one or more reference files without using a processor?

A. No.

Q. Why not?

A. Well, the file is electronic.

Q. Could you print the file to a piece of paper and print the reference file and go through it manually?

A. I haven't thought through that super carefully. I would imagine that you could make an argument that that's still using a processor.

Q. How so?

A. Well, you had to use the processor to print the file.

Q. All right. But is that using the processor to test the set of genes?

A. It was part of the test.

Q. All right. Would you say that testing a set of -- Okay.

So how are the set of genes tested for

false positives against the reference files? I mean, what is the procedure for doing that?

A. Well, so what you would do -- Let's take the standard '344 example of a Windows binary. You would have to take the file and decompile it, have a long list of all the API calls and then you have to go through and see which genes that lines up with, and then compare it against the signatures.

Q. Okay. So is it possible to do that portion without a processor?

MR. LIU: Objection; vague.

THE WITNESS: You could not decompile the program without a processor.

BY MR. PANNO:

Q. Okay. So could you take the decompiled program, or a printout thereof or whatever, and compare it against the reference file without a processor?

MR. LIU: Objection; incomplete hypothetical.

THE WITNESS: I suppose that's theoretically possible.

BY MR. PANNO:

S. NIELSON

Q. All right. So in that case, you say you still need to use a processor to print the file?

A. Sure.

Q. Okay. So that is the basis for, in that example, the inherency of using a processor?

A. Maybe I should step away from the word "inherent." That's a loaded word. I would say one of ordinary skill in the art, looking at these disclosures, would assume a processor, otherwise it's too error-prone and too time-consuming for a human to do and have any reasonable success.

MR. PANNO: I have no further questions, Dr. Nielson. I'll pass the witness.

MR. LIU: Can we take maybe a two-minute break?

MR. PANNO: Absolutely.

THE VIDEOGRAPHER: Off the record at 10:22.

(Recess taken.)

THE VIDEOGRAPHER: We are now back on the record at 10:24.

S. NIELSON

MR. LIU: Just a few questions.

EXAMINATION

BY MR. LIU:

Q. Dr. Nielson, you've been previously handed previously marked Exhibit 1007.

A. Yes.

Q. That's your declaration, correct?

A. Yes.

Q. Does this exhibit, this declaration, accurately reflect your opinions on the '344 patent and the prior art as you sit here today?

A. Yes.

MR. LIU: I have nothing further.

THE VIDEOGRAPHER: We are now off the record at 10:25.

(Deposition adjourned at 10:25 a.m.)

ACKNOWLEDGMENT OF DEPONENT

I, SETH NIELSON, Ph.D., have read or have had the foregoing testimony read to me and hereby certify that it is a true and correct transcription of my testimony with the exception of any attached corrections or changes.

_____

SETH NIELSON, Ph.D.

[ ] No corrections

[ ] Correction sheet(s) enclosed

SUBSCRIBED AND SWORN TO BEFORE ME, the undersigned authority, by the witness, SETH NIELSON, Ph.D., on this the _____ day of _____, _____.

C E R T I F I C A T E

STATE OF MARYLAND

I, JOHN L. HARMONSON, a Notary Public within and for the State of Maryland, do hereby certify:

That SETH NIELSON, Ph.D., the witness whose deposition is hereinbefore set forth, was duly sworn by me and that such deposition is a true record of the testimony given by such witness.

I further certify that I am not related to any of the parties to this action by blood or marriage; and that I am in no way interested in the outcome of this matter.

IN WITNESS WHEREOF, I have hereunto set my hand this 23rd day of October, 2015.

_____

JOHN L. HARMONSON, RPR

My commission expires: 08/02/17

---------- I N D E X ----------

---------- EXHIBITS ----------

ERRATA SHEET

CASE: Fortinet, Inc. v. Sophos, LLC

DATE: October 21, 2015

WITNESS: SETH NIELSON, Ph.D.

Reason Codes:

1. To clarify the record.
2. To conform to the facts.
3. To correct transcription errors.

Pg.	Ln.	Now Reads	Should Read	Reason
___	___	_____	_____	_____
___	___	_____	_____	_____
___	___	_____	_____	_____
___	___	_____	_____	_____
___	___	_____	_____	_____
___	___	_____	_____	_____
___	___	_____	_____	_____
___	___	_____	_____	_____
___	___	_____	_____	_____
___	___	_____	_____	_____

_____

Signature of Deponent

SUBSCRIBED AND SWORN BEFORE ME

THIS ___ DAY OF _____, _____

_____