

UNITED STATES DISTRICT COURT
CENTRAL DISTRICT OF CALIFORNIA
HONORABLE DAVID O. CARTER, JUDGE PRESIDING

- - - - -

ECHOSTAR SATELLITE CORPORATION,)	
et al.,)	
)	
Plaintiffs,)	
)	
vs.)	No. SACV 03-950 DOC
)	Day 5, Volume III
NDS GROUP PLC, et al.,)	
)	
Defendants.)	
_____)	

REPORTER'S TRANSCRIPT OF PROCEEDINGS

Jury Trial

Santa Ana, California

Wednesday, April 16, 2008

Debbie Gale, CSR 9472, RPR
 Federal Official Court Reporter
 United States District Court
 411 West 4th Street, Room 1-053
 Santa Ana, California 92701
 (714) 558-8141

EchoStar 2008-04-16 D5V3

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

APPEARANCES OF COUNSEL:

FOR PLAINTIFF ECHOSTAR SATELLITE CORPORATION, ET AL.:

T. WADE WELCH & ASSOCIATES

BY: CHAD M. HAGAN

CHRISTINE D. WILLETTS

WADE WELCH

Attorneys at Law

2401 Fountainview

Suite 700

Houston, Texas 77057

(713) 952-4334

FOR DEFENDANT NDS GROUP PLC, ET AL.:

O'MELVENY & MYERS

BY: DARIN W. SNYDER

DAVID R. EBERHART

Attorneys at Law

275 Embarcadero Center West

Suite 2600

San Francisco, California 94111

(415) 984-8700

-AND-

HOGAN & HARTSON

BY: RICHARD L. STONE

KENNETH D. KLEIN

Attorneys at Law

1999 Avenue of the Stars

Suite 1400

Los Angeles, California 90067

(310) 785-4600

ALSO PRESENT:

David Moskowitz

Dov Rubin

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

I N D E X

WITNESSES	DIRECT	CROSS	REDIRECT	RECROSS
RUBIN, Avi				
By Mr. Hagan	6		126	
By Mr. Stone		44		

EXHIBITS

EXHIBIT NO.	IDENTIFICATION	IN EVIDENCE
225	Excerpt of Dr. Rubin's report	47
800	Report	70
802	Copy of presentation	74

1 SANTA ANA, CALIFORNIA, WEDNESDAY, APRIL 16, 2008

2 Day 5, Volume III

3 (1:01 p.m.)

4 (Outside the presence of the jury.)

5 THE COURT: All right. We're on the record. All
6 counsel are present.

7 I've gone back to tell Mr. Bender that we have a
8 call in to Kathy Peel (phonetic) at Orange County
9 Transportation Authority seeing if we can reschedule working
10 with the County the date of his presentation until Monday
11 afternoon.

12 Kathy Peel has not called back. I wanted to
13 assure Mr. Bender that the Court had followed through with
14 that effort. That's why I was back there for a moment.
15 There's been no other conversation about the case, except I
16 always take my clerk with me. And a note was handed to the
17 clerk by one of the jurors:

18 "There was the lady on the NDS side, second row
19 from the back, that I find very distracting because it looks
20 like she's recording the trial."

21 I don't know if that's true or not. I'm going to
22 not inquire any further. I don't see the lady, second row
23 from the back, at the present time.

24 There's no recording of the trial. Thank you very
25 much.

1 (In the presence of the jury.)

2 THE COURT: All right. Thank you. The jury's
3 present. All counsel are present.

4 Let me state that I received a note from the jury
5 saying that one of the jurors was concerned about a lady in
6 the second row on the NDS side -- and, of course, that
7 doesn't mean they're on the NDS side, but the NDS side of
8 the court -- second from the back row, that's distracting
9 because it appears that she is recording the trial.

10 Let me remind all parties that there's to be no
11 recording of this trial. Thank you very much.

12 Counsel, if you would like to continue with your
13 direct examination.

14 MR. HAGAN: Thank you, Your Honor. Chad Hagan on
15 behalf of the EchoStar and NagraStar plaintiffs.

16 AVI RUBIN, PLAINTIFF'S WITNESS, PREVIOUSLY SWORN

17 RESUMED THE STAND

18 DIRECT EXAMINATION (Continued)

19 BY MR. HAGAN:

20 Q. Good afternoon, Dr. Rubin. Before we broke for lunch,
21 you identified for the ladies and gentlemen of the jury the
22 four key components of the hack methodology for EchoStar's
23 system that the defendants developed. And you discussed
24 briefly with them and gave us a demonstration about the I/O
25 buffer overflow vulnerability.

1 I want to change focus now and talk about the three
2 other key components. And the second one you identified was
3 the RAM ghost effect, or address alias. Can you explain to
4 the ladies and gentlemen of the jury what you're referring
5 to there?

6 A. Sure. And I already did this to some degree before.
7 It's basically the notion that memory is finite. Right. So
8 it only goes so far. So what happens if you give an address
9 that's beyond the bounds of memory?

10 And in the Smart Card it's a little different from a
11 computer because you have so little memory. So when you
12 only have a little bit of memory and if you try to address
13 something, like you asked for a location, try to store
14 something at a location, or read something from a location
15 that's larger than the amount of memory you have, this
16 particular chip had an unusual property, which is this
17 wrapping around, of ignoring the most significant digit
18 there.

19 Q. And if the RAM ghosting effect was not inherent in the
20 chip used by EchoStar, would the I/O buffer overflow
21 vulnerability have made a hack possible in the way the
22 defendants developed it?

23 A. This particular attack relied on the RAM ghost effect,
24 on this wrapping around. As I'll show you when I walk you
25 through the attack, if that property had not been there,

1 this attack would not have worked.

2 Q. Let's focus briefly on the third key component, and
3 that is sophisticated knowledge and usage of the index
4 variable. Can you explain for us what you mean by that.

5 A. Sure. So again, there was a particular byte in
6 memory -- and I'll show it to you on a chart where that was
7 located approximately -- that had a value. And that value
8 was important in determining when messages were read in on
9 the communication, where they were stored in the
10 communication buffer.

11 So to back up a little bit, when -- you've got your
12 set-top box, right? That's the thing that you put on top of
13 your TV. And it has a Smart Card in it. The way that the
14 system functions is through messages. The set-top box,
15 which is called the IRD, will send a message, just bits, to
16 the Smart Card, which will then process them.

17 And the first thing that happens when a message comes
18 in from the set-top box is that the bits of that message are
19 copied into a buffer, into a section of memory that is
20 reserved for that purpose.

21 How do they get there? Well, that's through this index
22 available. The index available is incremented. Every time
23 a byte is copied from the IRD into the memory buffer, that
24 value increases by one. And the reason for that is so that
25 the next byte will come right after the last byte. So this

1 index available in memory keeps being incremented, which
2 causes the next byte to be written consecutively.

3 Now, this attack relies on maliciously and
4 intentionally changing that index variable.

5 So what you're doing is you're going to overflow the
6 buffer, you're going to wrap around. And you're going to
7 craft the message so that the index variable is not what
8 it's supposed to be. What is it? Well, it's going to be
9 controlled so that the next byte read in from the set-top
10 box will go wherever the attacker wants it to go. And I'll
11 show you how that plays into the attack.

12 If you didn't understand the working of that index
13 variable, you could not have written this attack.

14 Q. Then, finally, the fourth key component that you
15 identified through analyzing the defendant's hack
16 methodology was the behavior of the exception handler. And
17 I think you mentioned that briefly that this morning and how
18 that related to an invalid checksum error. Can you walk us
19 through that really quickly?

20 A. Sure. And I'm trying not to repeat the part I already
21 told you.

22 So when the set-top box sends a message to the Smart
23 Card in a legitimate operation, the message is computed as
24 it's supposed to be. There are all kinds of messages sent
25 for all kinds of reasons. One of them might be, you know, I

1 changed the channel. I want to start decrypting that
2 channel now. And so that causes messages between the
3 set-top box and the receiver.

4 The designers of the system will program a whole bunch
5 of different messages for operation. This is just normal
6 use.

7 Now, they're worried, of course, as I said before, that
8 somebody might have not intentionally maybe but through a
9 glitch or communication error caused the message to be
10 received incorrectly. And it's not that uncommon for there
11 to be problems in hardware or software. So as a security
12 blanket, they put in this checksum. The checksum ensures
13 that when you receive the message, you have this way of
14 knowing that it actually arrived intact.

15 If it doesn't arrive intact, there are different
16 options. One of the things you can do is send a message
17 back saying I didn't get it intact. Please retransmit it.

18 Another thing that you could do is just do nothing.
19 Right. So if you ever push your remote and nothing happens,
20 well, maybe that's what happens and you'll push it again.
21 So there are different things you might do in response to
22 the checksum being bad.

23 Because of that, they couldn't hard-program in the
24 exception handling. Because sometimes you do one thing,
25 sometimes you do another thing. So what they allowed for is

1 flexibility in the code.

2 And so the way the card designers did it was they said
3 let's just go to a particular address on the top of the
4 stack, and we'll see what instructions that points to and
5 then we'll just do whatever it says there.

6 And again, this is something like the other property
7 where if you didn't understand that the card did that when
8 there was a bad checksum and then -- then, you know, the
9 attack just doesn't work. It's completely based on these
10 four properties.

11 Q. Thank you, Dr. Rubin. And I know we'll get to the
12 particular attack and the Nipper attack in a little bit.

13 But for right now I think it's important to talk a
14 little bit about reverse engineering.

15 A. Sure.

16 Q. You have significant experience in reverse engineering
17 products; is that correct?

18 A. That's right.

19 Q. Both as part of your company, Independent Security
20 Evaluators, as well as through your work for Johns Hopkins
21 University; is that correct?

22 A. That's right.

23 Q. And can you tell us about one of those
24 reverse-engineering processes?

25 A. Sure. So there have been several of these that we've

1 done over the years. Let me talk about the Exxon Mobil
2 speed pass.

3 If you're familiar with the Exxon Mobil speed pass,
4 it's a product which is a little device that you can put on
5 your key chain. And you get this from Exxon Mobil so that
6 you can go to a gas station and just hold it up to the gas
7 pump. And you don't have to run your credit card or
8 anything, and it will perform a cryptographic exchange.

9 And that's important because you don't want to be able
10 to get someone else's credit card number used. So it's got
11 some keys on it. And that happens really quickly. It
12 literally takes an eighth of a second for the message to be
13 transmitted and transmitted back.

14 So you hold the thing up. It's a convenience feature.
15 And you buy your gas. And on the back end system, it looks
16 up your credit card number, it performs some encryptions;
17 and if you're the authorized person, then you get to buy the
18 gas.

19 Interestingly, we discovered that in many Ford
20 vehicles, like a hundred thousand of them, they were using a
21 chip in the key that's exactly the same chip in the Exxon
22 Mobil speed pass. And so we set out in my lab at Johns
23 Hopkins -- and I think it basically happened over lunch one
24 day when somebody said, "I have one of these. I wonder how
25 secure it is" -- to make that determination.

1 And so we got some equipment, we set up a lab, and we
2 basically tried hard to break this thing.

3 One of the justifications for doing this -- because
4 whenever we set out to do one of these projects, which can
5 often become controversial -- is to figure out if there will
6 be enough benefit to society and the people for us doing
7 this, or a net loss. And we have had projects where we
8 start out all excited and then decide not to do it.

9 So I called the general counsel of the university and
10 said we are interested in studying the Exxon Mobil speed
11 pass. We plan on breaking it. If we're successful breaking
12 it, we will call Texas Instruments and call Exxon Mobil.

13 And the reason Texas Instruments was because they're
14 the manufacturers of the chip. They told me it would take a
15 while, they would think about it. And then they got back to
16 me and said, "We'll support you in that effort."

17 We then thought about, well, what if they're not very
18 happy with us if we're successful, and they sue us?

19 So I called up the lawyers at the Electronic Frontier
20 Foundation, which is a nonprofit organization funded, I
21 think, to the tune of about \$10 million. And their purpose
22 is to answer these kind of questions for researchers. And
23 they take the cases pro bono.

24 So I retained their lawyers, and they did a study and
25 wrote a brief about whether or not it was ethical and

1 whether or not it was legal to do what we were interested in
2 doing. I had worked with them before on the electronic
3 voting machines, and so they were familiar with my lab and
4 how we worked, as was the Johns Hopkins general counsel.

5 So we received the brief back, and the brief was -- was
6 encouraging: We wouldn't be breaking any laws.

7 In fact, they agreed with us that there was enough
8 benefit to consumers when they make a decision whether or
9 not to use this product, to understand how secure it is.

10 And so we set to work. And we -- it took us three
11 months of my students working around the clock. Actually,
12 they get pretty excited about these projects. We were
13 sending test messages to the thing. It's basically what's
14 called an RFID chip, a radio frequency identifier.

15 And if you follow the technical press, these are really
16 up-and-coming technologies. And we -- it took us a while to
17 be able to understand the communication. But eventually we
18 were able to get the devices to respond to us.

19 And we built what we called a cracker, which is a
20 parallel array, meaning a whole bunch of these -- and this
21 is a very technical term, FPGA, field-programmable gate
22 arrays. What these are is a fancy way of saying little
23 bitty computers. And we got these little bitty computers
24 and lined them up 16 in a row. And their purpose was to
25 search for cryptographic keys.

1 And because of the nature of FPGA's, they can do it
2 very, very fast. And so we set these things to work. And
3 once we had understood the communication between our
4 equipment and the speed pass, all we really needed was a big
5 breakthrough on our cracker.

6 So we set up little lights on each of the crackers, and
7 we put a webcam on it. And the thing was, once we found a
8 key, a light would turn on, we'd see it on the webcam. So
9 about five of us involved in the project, we're constantly
10 looking at our computers all over the country, constantly
11 looking at our web browsers. I was -- I remember being at a
12 conference and sitting there with my laptop -- so I remember
13 being at a conference and looking on my laptop all the time
14 at the webcam to see if that light went on.

15 It happened one day at about 2:00 in the morning, and I
16 got a call from one of my grad students, who woke me. He
17 said the light went on. I waited till the next morning and
18 went into the lab. And we had obtained the key that was in
19 that Exxon Mobil speed pass.

20 Without going into a lot of the technical details, once
21 we had that breakthrough, it gave us a lot of information,
22 and so we were able to then reproduce that a lot faster.
23 And we got to the point where we could crack a key in about
24 20 minutes.

25 So we felt, you know, pretty proud of ourselves. We

1 wrote up our results. And I called the lawyers again and I
2 said, "Well, we did it."

3 And they asked me what my plan was. And I had been
4 involved in these kinds of situations before where we had
5 something like this, and now what do we do with it.

6 So I called up Exxon Mobil and I said, you know, "I
7 need to talk to somebody because we've cracked the Exxon
8 Mobil speed pass. I'm a professor at Johns Hopkins."

9 And my call was returned pretty quickly. And they
10 asked me if they could see it. And so we invited them out
11 to the lab. We hadn't told anybody about this except the
12 Hopkins lawyers. And they also brought Texas Instruments
13 with them. And they showed up with a little bag full of
14 chips. Some were Exxon Mobil speed pass, some were car key
15 chips. We had already tested our attack on the car key
16 chips made by Ford.

17 Okay. So they showed up with a bag of chips -- not a
18 bag of chips but a bag of computer chips. And they
19 challenged us and they said, "We're gonna have this meeting
20 with you. We want you to tell us what you did and how you
21 did it. But meanwhile we want you to crack these chips. We
22 really want to see if you can do it."

23 So one of my grad students took it into the lab and put
24 the cracker to work and was able to break them a lot faster
25 for a reason that I think you guys can now appreciate, which

1 is that they had manually entered some keys into them, but
2 they hadn't bothered to enter any of the hex letters A, B,
3 C, D, E, F. It was all numbers. They entered them in
4 decimal, but the computers considered them to be in hex.
5 And so a lot of values were impossible. And we found the
6 keys very, very quickly.

7 So while we're having the meeting where we're walking
8 them through how we broke the system, they -- you know, my
9 student came into the room with a piece of paper with all
10 the keys written on them. And they pulled out a piece of
11 paper from their pocket and compared them and said, "Yes,
12 you guys did it. Now, what?"

13 We said, "This is our objective. We're going to
14 release this information, but we're going to do it
15 responsibly. We believe in responsible disclosure." And I
16 outlined our plan which we had come up with in advance for
17 them. Which was, first, we were going to write a technical
18 paper. The technical paper would describe the new science
19 we had invented in order to perform this attack because it
20 was things that no one had ever done before. That's what
21 grad students are supposed to be doing. And we were going
22 to submit it to a technical conference.

23 But we were going to omit the specific details of the
24 attack. We were not going to write something that would
25 enable somebody else to go and do it. We're simply going to

1 show the steps that we carried out.

2 The next step that we were going to do is -- we knew
3 that this would be very interesting for the media. And we
4 didn't want to do this in an ad hoc way. And I had a
5 relationship with a reporter at the New York Times who I had
6 worked with on a lot of the electronic voting stories. We
7 were going to give him an exclusive, invite him down to our
8 lab, show him what we did, and then he would write a story
9 in the New York Times. And then whatever happened happened.
10 But we would not release our paper until the conference six
11 months later.

12 Exxon Mobil wasn't very happy about this. They said,
13 "Well, we'd really like it if you don't release this."

14 We said, "We know that you would be, but we're going
15 to. What we want is for you guys to fix this before we
16 released it. And here" -- and we had worked on this.
17 "Here's how to fix it. It's a very simple fix. Use a
18 longer key."

19 So that's how it transpired. The New York Times story
20 ran. The media really picked up on this. The show
21 20/20 came to our lab and filmed an interview with me. And
22 then they had a reporter walk into the lab with an Exxon
23 Mobil speed pass in their pocket that they brought with
24 them, and asked us to drive them to a gas station and buy
25 gas and have that reporter pay for it. We knew that was the

1 setup.

2 My students had the scanning equipment, and we asked
3 the person to walk over here. We scanned them, got the
4 information we needed from their FID chip. We then went
5 into our cracker, we broke the key. We had a simulator we
6 had built on a laptop. And we drove to a gas station with
7 the reporter in the back seat. And we held up an antenna
8 that we had connected to the laptop up to the gas pump. And
9 sure enough, the tiger lit up. And we bought gas and we
10 left.

11 In the story that ran on 20/20, they showed --

12 THE COURT: Did you get to pay for it?

13 THE WITNESS: No, that reporter paid for it.

14 At this time gas prices were getting a lot higher,
15 and so we were inviting a lot of reporters in to do this.

16 In the story that ran on 20/20, the reporter
17 showed their bill, that they had actually been the one who
18 had paid for the gas.

19 So the point of this is that my specialty is
20 really breaking systems like this. And over the years I've
21 developed a methodology that I call responsible disclosure,
22 of how you go about deciding whether or not to take on a
23 particular project and then undertaking the project,
24 informing all the parties responsible, making sure a fix can
25 get out there, and then publishing.

1 And I should mention that our technical paper won
2 the Best Paper Award at the USENIX Security Conference. I
3 know the term USENIX has come up in this trial before. It's
4 the premier systems organization.

5 BY MR. HAGAN:

6 Q. Thank you, Dr. Rubin.

7 If I understood the process correctly, you and your
8 team at Johns Hopkins set out to determine whether or not
9 you could reverse-engineer and crack this Exxon Mobil speed
10 pass.

11 Before you did that, you thought there may be
12 implications, so you contacted the general counsel and you
13 contacted lawyers for the electronic -- what was it, EFF?

14 A. Yes, Electronic Frontiers Foundation.

15 Q. And who's the EFF?

16 A. It's a nonprofit organization which was created by
17 donations of some very wealthy individuals to provide legal
18 advice and support to technologists.

19 Q. And you notified them of the efforts that you and your
20 team at the university intended to engage in, and they gave
21 you a written report saying that it was okay to go down that
22 path; is that correct?

23 A. That's right.

24 Q. And then after you determined that there was a
25 vulnerability in this speed pass, you contacted the

1 manufacturer of the chip.

2 A. Right.

3 Q. And you contacted Exxon Mobil.

4 A. Right.

5 Q. And then you set up a meeting where they could come
6 into your lab, and you could show them what the
7 vulnerabilities were, correct?

8 A. Right.

9 Q. And then you went a step further. You actually
10 developed a patch. You and your team developed a software
11 patch or a way to fix that vulnerability; is that correct?

12 A. Right. Well, in this case it was simply advice. In
13 other cases we've developed patches.

14 Q. Did you charge Exxon Mobil anything for that advice?

15 A. No.

16 Q. Did they take that advice and try to improve the
17 robustness of their product, or did they take steps to try
18 to improve the robustness of their product?

19 A. They told us that they were.

20 Q. If I understand your testimony at your previous
21 deposition, you and your team at ISE also engaged in efforts
22 to reverse-engineer the Apple iPhone.

23 A. That's right.

24 Q. And did you follow similar procedures with respect to
25 the Apple iPhone project?

1 A. Yes, we followed the exact same procedures.

2 Q. And that was contacting the EFF attorneys?

3 A. We started with our own general counsel, in this case
4 my wife. We told her what it was we were planning on doing.
5 And she did some research. And then we talked to the same
6 EFF lawyers that we had talked with during Exxon Mobil speed
7 pass.

8 Q. And were you and your team at ISE able to effectively
9 reverse-engineer the Apple iPhone?

10 A. Yes. We were able to crack it in a week.

11 Q. Did you determine whether or not there were
12 vulnerabilities in that technology?

13 A. Yes, we did.

14 Q. After you made that determination -- can you tell the
15 ladies and gentlemen of the jury the steps that you took?

16 A. Yeah.

17 So once we had figured out how to break the security on
18 the iPhone -- and I should point out that this was a few
19 weeks after the iPhone came out. We were planning on, you
20 know, looking at it -- everyone. There was a lot of buzz in
21 the media about the iPhone. And we thought, let's see what
22 they did with security and how good it is. And once we saw
23 that it wasn't really that good and that we could break it,
24 we contacted Apple.

25 They have a web page that's devoted to people informing

1 them of vulnerabilities in their products. And that should
2 give you an idea that this is not something that happens
3 with infrequency. So we submitted a vulnerability report to
4 them, telling them exactly where in their system the problem
5 was and providing them with a patch that would fix it and
6 telling them what our plans were for publication and for
7 releasing the information, pretty much the same as we did
8 with Exxon Mobil.

9 Q. And did you follow those same steps that you called
10 responsible disclosure with respect to the Apple iPhone
11 project?

12 A. Yes. I recall a meeting that I called my staff in to
13 discuss how we were going to release the information and
14 what our steps were going to be.

15 In my mind it's a pretty formal process, when you
16 engage in this kind of research, to have a laid-out game
17 plan for what you're going to do with respect to the release
18 before you even really start to break something.

19 Q. And were you able to develop a software patch or fix
20 for the Apple iPhone?

21 A. We did.

22 Q. And did you notify Apple of that and provide them with
23 that information?

24 A. We did. Like I said, we actually sent them the patch.

25 Q. And did Apple eventually end up improving their product

1 and improving the robustness of the Apple iPhone?

2 A. Yes. Apple released their own patch before the date
3 that we said we were going to publish. And they didn't use
4 our code, but they did thank the engineers from my company
5 in the patch, basically credited us with being the ones who
6 found it.

7 I should point out that this is a relationship that
8 large companies, software companies often have with
9 researchers who find vulnerabilities: You show us the
10 vulnerabilities, and in exchange, when we fix it, we'll
11 credit you. It's kind of an unwritten rule.

12 Q. Dr. Rubin, are you also familiar with the term "ethical
13 reverse engineering"?

14 A. Yes.

15 Q. Can you describe for us what that term means?

16 A. The way that term is used is basically to contrast the
17 kind of reverse engineering that we do where -- I'm not
18 talking about the iPhone now; I'm talking about for our
19 customers where somebody has asked you to reverse-engineer
20 something for them and has hired you to do that versus
21 nonethical, which would be where you are doing it with some
22 malicious intent, as a malicious hacker.

23 Q. Now, do you always practice ethical reverse engineering
24 and responsible disclosure protocols in the work that you
25 engage in with your team at Johns Hopkins and with ISE?

1 A. That's a fundamental goal of ours.

2 Q. Why is it important to practice those protocols?

3 A. You know, it's a very, I think, controversial and
4 loaded activity that we partake in. And if you're not
5 careful to first talk to lawyers and to have the studies
6 done, you may find yourself in a precarious position where
7 you may have done something that could get you into hot
8 water. So we err on the side of caution.

9 I can give you an example. I will give you an example
10 now of a project that my guys thought would be really cool
11 to undertake and that we explored undertaking and that the
12 lawyers told us not to do and that we didn't do.

13 You're familiar with iTunes. And iTunes has a movie
14 rental capability now. And wouldn't it be cool to be able
15 to kind of break or hack the iTune system so that when you
16 rent a movie, instead of only getting it for 30 days, like
17 is their policy, you could get it for an unlimited amount of
18 time? That was something that occurred to one of my guys in
19 my company. We're always thinking like that: How can we
20 defeat this or that.

21 I asked the lawyers about that.

22 They said, "Well, there are several reasons why you
23 shouldn't do this."

24 Number one, they said it was a violation of a law
25 called the Digital Millennium Copyright Act because the

1 content -- DMCA, for short. The content in these movie
2 rentals is copyrighted. And reverse-engineering a
3 copyrighted system is much more serious in the U.S. legal
4 system than reverse-engineering something that doesn't
5 protect copyrighted material. And this has to do with laws
6 such as the DMCA, which were specifically designed for
7 copyrighted material.

8 And so I told the guys this project's not a go and
9 we're not going to do this. And we didn't do that one.

10 Q. Now, have you ever had any of your clients on behalf of
11 ISE come to you and ask if you will engage in efforts to
12 reverse-engineer and hack their competitors' technology?

13 A. Yes, we have had that happen.

14 Q. And have you undertaken those projects?

15 A. No.

16 Q. Why is that?

17 A. We're not comfortable doing that. The first name of my
18 company is "Independent," Independent Security Evaluators.
19 And we feel that being hired to hack someone's competitor
20 kind of aligns us with them a little bit. And we didn't
21 feel that we would truly be independent if we did that.

22 Q. Now, you were in the courtroom when David Mordinson,
23 the defendant's engineer, one of their employees that worked
24 on the EchoStar hack, testified; is that correct, sir?

25 A. Yes, I was.

1 Q. And you heard Mr. Mordinson testify that he didn't tell
2 or NDS didn't tell anyone at EchoStar or NagraStar or
3 NagraVision or Kudelski that they were engaging in the
4 efforts to hack EchoStar's security system; is that correct?

5 A. That's right.

6 Q. And you were also here when he testified that the
7 defendants didn't even notify the chip manufacturer that
8 they were engaging in these efforts to reverse-engineer and
9 hack EchoStar's security system; is that correct?

10 A. That's right.

11 Q. Did you have an opportunity to read in detail
12 Mr. Mordinson's report, the Headend Report?

13 A. Yesterday.

14 Q. Did you find any actions in that report that discussed
15 how the defendants could improve the robustness of their
16 product or their security system by hacking EchoStar's
17 technology?

18 A. No, that wasn't in there. It was really a description
19 of how to hack it and an actual attack.

20 Q. Do you have an opinion as to whether or not the
21 defendants followed protocol for ethical reverse engineering
22 and responsible disclosure when they hacked EchoStar's
23 security system?

24 MR. STONE: Objection. Assumes facts not in
25 evidence, and outside the scope.

1 THE COURT: There's a caveat to this.

2 I'm going to allow his opinion, but remember, this
3 is his protocol. Whether there's an industry-wide protocol
4 or whether this turns out to be a violation of the law when
5 I give you those instructions may be a far different
6 question than the limited opinion I'm going to let him voice
7 at this time.

8 Overruled.

9 You can answer that.

10 THE WITNESS: I don't believe that they followed
11 ethical reverse engineering or responsible disclosure in
12 that.

13 BY MR. HAGAN:

14 Q. Thank you Dr. Rubin.

15 I want to turn your focus now to Exhibit 998.

16 MR. HAGAN: Christine, if you could give him a
17 copy of that.

18 And, Clint, if you could publish the first page.

19 (Document displayed.)

20 MR. HAGAN: Your Honor, this has already been
21 admitted into evidence.

22 BY MR. HAGAN:

23 Q. Dr. Rubin, can you identify this document for the
24 ladies and gentlemen of the jury?

25 A. Yes. This is the Nipper post from December 21st that

1 was reposted on December 23rd.

2 Q. And is this the hack methodology that you referred to
3 earlier in your testimony as the recipe?

4 A. This does contain the recipe, yes.

5 Q. And as part of your work in this case, did you conduct
6 an analysis of this recipe?

7 A. I did. I dissected this and looked at it very, very
8 carefully.

9 Q. And based on your analysis, were you able to determine
10 whether or not there were any similar fundamental components
11 of the Nipper hack methodology that was posted on
12 Mr. Menard's website and the hack methodology developed by
13 the defendants?

14 A. Yes, I was.

15 Q. And can you identify those for us?

16 A. Sure.

17 So when I looked through this code and basically
18 through this hack, it was really kind of deja vu because it
19 has the exact same four components that I listed earlier
20 that are in the Haifa attack.

21 It exploits a buffer overflow in the exact same way
22 that the Haifa report does. It relies on this unknown
23 property of the ghosting effect to wrap around memory to get
24 to the index variable, relies on the index variable and
25 demonstrates a complete understanding of the operation of

1 the index variable. And also has an incorrect checksum at
2 the end of it which causes a jump to the address that's on
3 the end of the stack, which happens to be the right place in
4 the attacker's code.

5 So as far as I could tell, there was a blueprint that
6 both of those programs were following. And the thought that
7 came to my mind was that it's the same DNA. These are a
8 strain of the same thing.

9 Now, the code in the two programs is different. You
10 know, it's really the structure that's the same. These --
11 the methodology, these four things are the same. And so in
12 my mind it was as though the program was written twice maybe
13 by different people or maybe by someone who had learned more
14 about the chip when they wrote the second one.

15 Q. Now, as part of your work, did you compare this hack
16 methodology, the Nipper hack methodology, with the
17 defendant's hack methodology?

18 A. Yes, I did.

19 Q. And can you describe in terms that we can understand
20 and get our arms around -- can you describe how you went
21 about that analysis and comparison?

22 A. Sure. At this point it might help for me to use the
23 demonstrative.

24 MR. HAGAN: Your Honor, may I publish the
25 demonstrative?

1 (Document displayed.)

2 THE WITNESS: Okay. What I have here, this is the
3 code in Mordinson's report, the Haifa attack. As I told you
4 earlier, it has the 0X and then hex values throughout.

5 And over here is the code from the Nipper posting.
6 We've looked at that posting several times. And this is
7 what was in there as how to attack the card.

8 What I want to do is walk you through. On this
9 chart I have memory. And I have two identical copies here.
10 And this would be the RAM of the Smart Card. And what I'd
11 like -- the purpose that I have in showing you this is to
12 demonstrate this DNA property, the similarity of these two
13 attacks. And it's technical, and I'm going to go slowly.

14 So the IRD sends this message to the Smart Card.
15 Right. Just to show you -- so when it sends hex 21, right,
16 that's 0010, 0001. It's our shorthand. Right. And so each
17 one of these corresponds to a whole bunch of zeros and ones,
18 and we're not gonna say that again.

19 So what happens is, in RAM there's this one byte
20 here which is just hanging in the middle there. And this --
21 I haven't seen something like this before. It's unusual.
22 And this is the index variable. And it's got a value.

23 And what happens -- now I'm going to be the
24 computer, okay. I'm the computer inside the Smart Card, and
25 I'm going to do what the computer inside the Smart Card is

1 gonna do. I'm the Smart Card and I received this message
2 and I'm going to take the first byte that I'm receiving and
3 I'm going to consult the index variable, say, "What's the
4 value in the index variable?" "Thank you."

5 And now I'm going to use a formula that I have in
6 my code to calculate, using the index variable, where should
7 I put this byte that I just read.

8 And it comes right here. 21. In fact, this last
9 box here is what's called the I/O buffer. Input/output
10 buffer. This is where the Smart Card places things that it
11 reads from messages.

12 And the next thing I'm going to do after I read in
13 this byte is I'm going to increment the index variable. I'm
14 going to add one to it. Why? So that I'll copy the next
15 byte there. Let me do that. Don't worry, I'm not going to
16 do all of them.

17 Zero, zero. And on and on.

18 And so what I'm going to do is I'm going to copy
19 all of these bytes into here. But guess what? There's a
20 problem.

21 The problem is that when I get to this byte right
22 here, this zero, zero -- and I'm the one that put these
23 lines here. This is just all the numbers.

24 When I get to here, I'm now here. I'm at the end
25 of the buffer. Now, someone sending a message this long is

1 intentionally overflowing the buffer. Right. This is gonna
2 overflow the buffer.

3 And so what's gonna happen to this byte? Well,
4 the developers of this -- of this card assumed that when
5 something got copied past the end of the buffer, it would be
6 written off here. It would fall out of memory, right.
7 Because they didn't know -- and we heard testimony yesterday
8 that they didn't know about the RAM ghost effect.

9 So, in fact, these bytes don't get copied over to
10 here because of the RAM ghost effect. They get copied over
11 to here.

12 Now, I'm going to tell you something about this
13 card. The first hex 20 bytes, which is 32 in people talk,
14 the card does not actually let you write to the memory.
15 It's protected. So what happens when these zero bytes are
16 written to here? Nothing. Nothing changes in the memory.
17 It's just an operation that the attacker had to perform to
18 get to the index variable. This is like a race to the index
19 variable. The attacker's goal is to change that index
20 variable to make the attack work.

21 So I've marked off this section of zeros. The
22 attacker knew that these wouldn't actually have any impact
23 whatsoever, so they just put zeros there.

24 And when I get to this point, I'm now at the end.
25 I just didn't write zero, zero there. And those bits go off

1 into thin air.

2 Now, once I'm done with that, this area of memory
3 is no longer protected. So now we're actually gonna copy.
4 And remember, we keep incrementing the value in the index
5 variable.

6 So this zero, zero will show up here, and these
7 values will be consecutively copied into memory up until
8 this point.

9 And notice that they're not all zeros. There's
10 81, there's a 5, right? So in here somewhere you're getting
11 81, you're getting 5.

12 The attacker was very clever, and he had all the
13 code to the chip, so he knew how it worked and realized that
14 there were actually valuable things in memory over there.
15 You can't just go around overwriting memory. Okay.

16 So in order to preserve the functionality of the
17 card so the hack would work, the attacker, David Mordinson
18 in this case, had to put legal values there that the card
19 would be able to operate with.

20 Now, as an attacker this can be done through trial
21 and error. Or you can sit down with a microscope and
22 tweezers and look at the code, figuratively. And look at
23 the code and figure out that needs to be an 81. And I'm not
24 sure how he did it. Maybe some combination of those.

25 But anyway, it turns out that these values that

1 were in the Haifa report are the correct values for this
2 attack to work.

3 The next thing is DF. And this is where it gets
4 clever, and this is the trick. DF gets written into the
5 index variable.

6 What that does is -- this was figured out using
7 the complex formula. With basic algebra you change the
8 unknown, right. We're trying to get something to a
9 particular spot here. And we know that DF is going to come
10 in and get us there. Right. So that's basic algebra.

11 And where the attacker wanted to be was right
12 here. Let me see which I drew as the stack. This is the
13 stack.

14 Now, for the purposes of this case, it's not
15 important for you to know what a stack is, and I'm not going
16 to tell you. But it's a section of memory used for things.

17 And what the attacker wanted is for 019C to be at
18 the top of the stack. And so he crafted DF so that the next
19 byte that could be copied, which was here, would get you
20 right there. Okay.

21 Now, why did he put it in twice? I wondered about
22 that. I looked at it a lot, and I said he wasn't sure about
23 this value, so he wanted to make sure, if he overshot, he'd
24 still get it. That's my theory. Anyway, it doesn't matter.
25 You put it in twice, belt and suspenders, you're gonna get

1 there. In fact, it does work.

2 Now, what is 019C? Well, interesting, that's the
3 address of the I/O buffer. And that is where all of this
4 code was copied.

5 So I actually misspoke. This first line here is
6 the header, and it doesn't get copied. So we start copying
7 here at 9D, and we copy all the way to 00. That's this.

8 Okay. So you're getting a feeling, I hope, for
9 how intricate this attack is, how much it relies on the
10 index variable, the ghost effect, the buffer overflow.

11 There's a signature here to this attack. And it
12 gets you over here to 019C. And then, guess what, we hit a
13 bad checksum. I'm the computer again. I say, well, gosh,
14 the checksum for this message is not FF. What am I supposed
15 to do? Well, I'm programmed to look at the top of the
16 stack, grab that address and start executing code there.
17 Right here. That code came from right here.

18 And so what is this? Well, if you look at the
19 Haifa report, the Headend Report, this is code that dumps
20 the EEPROM, okay. This is software code that could have
21 been written on the card originally. We call it shell code.

22 And this should also help you understand that this
23 is a generic attack. Once this is known, an attacker can
24 put whatever program code they want here, and it will run on
25 that card because of this attack.

1 Okay. If you have trouble following that, you're
2 gonna see it again now because the Nipper attack is very,
3 very similar to the Haifa attack.

4 It is also different in some ways. And the
5 differences, to me, demonstrate that it was either somebody
6 else who looked at this attack and said, "I'm gonna do the
7 say thing," or the same person learning a little bit more
8 about this chip took some shortcuts. And you'll see the
9 shell code is shorter. It's actually better written, it's
10 more compact, but it does the same thing.

11 Okay. So I'm going to shift gears now, and I'm
12 gonna talk about this posting which was on this website on
13 the Internet.

14 So let's start off. We have the same header. And
15 we start copying bytes. Now, what's kind of interesting is
16 notice that these bytes don't look like code. 1, 2, 3,
17 4, 5, 6, 7 -- it's wasted.

18 Okay. There is no objective to these bytes except
19 to get you to the index variable. Okay. Clearly the
20 programmer knew about that.

21 So what I've drawn here is fill the I/O buffer
22 basically with garbage, and you get to the 04 at the end.
23 And so far the attacker has done nothing except get prepared
24 to overflow the buffer.

25 All right. Well, now the attacker's gonna

1 overflow the buffer, and the attacker knows that it's gonna
2 wrap around and start writing things here. Only the
3 attacker also knows that these values don't matter. They
4 actually won't be written to the card because the first
5 20 hex bytes don't write. The card's protected. So you
6 basically can see zero, 1, 2, 3, 4, 5. The attacker doesn't
7 really care.

8 So we've gotten all the way down to here in this
9 message, and all we've done is try to find that index
10 variable.

11 Now, once we get to here, we start writing into
12 memory here. So we're going to write 0001, et cetera. None
13 of these values really matter either. Except some of them
14 do. Some of them have to overwrite legitimate values just
15 like Haifa did. Right. And so he didn't use 81 and 5
16 because there are other values that will not crash the card
17 either that are legal values.

18 And so those other values are written here. And
19 then the index variable is written.

20 So at this point the attacker's actually done
21 something really important.

22 Now, just like DF, actually where the attacker
23 wanted to be, C3 is gonna get you to where the attacker
24 wants to be. And where does the attacker want to be? This
25 attacker decided -- this code decided to be on the stack.

1 And again, it doesn't matter what the stack is. But C3 was
2 done very, very cleverly so that the next byte would be
3 written in the middle of the stack somewhere in a very
4 precise location.

5 Just to remind you that the index variable tells
6 the computer on the chip where to write the next byte.

7 Okay. So once we overwrite this, we no longer
8 keep writing over here. We're jumping over to here. And we
9 write the 9B, 9C. And this was actual code, so it's gonna
10 have more structure to it. It's not 1, 2, 3, 4, 5.

11 And this code was written so that right when you
12 get to the end of it, you're going to write 0060, and that's
13 gonna show up at the end of the stack.

14 Okay. So the attacker counted backwards to get
15 the shell code written to a place where once it was all
16 copied, the last byte would be at the top of the stack.

17 And then an incorrect checksum. The checksum for
18 this message is not 55. So what happens when an incorrect
19 checksum is reached? When an incorrect checksum is reached,
20 we're gonna go to the top of the stack and start executing
21 at that address. Well, what does that address point to?
22 Right there.

23 And so it's gonna start executing this code.

24 Now, let me point out why I believe that these are
25 pretty much the same program and why they're so similar.

1 Both of the programs require you to overflow a
2 buffer. Both of the programs rely on the ghosting effect
3 which wraps you back around to the top, which was a property
4 of this card that was not known.

5 Both of these attacks target the index variable.
6 They race to the index variable so that they can put the
7 precise value that they need to, to get exactly where they
8 need to be in the code. And both of them orchestrate an
9 address at the top of the stack, and they give an incorrect
10 checksum so that you will go to that address and start
11 executing the shell code.

12 Okay. I'm not a biologist, but this is the same
13 DNA to me. These two things, to me. This could not have
14 been done without access to the Mordinson report.

15 BY MR. HAGAN:

16 Q. Thank you, Dr. Rubin.

17 Now, Dr. Rubin, you were also present when
18 Mr. Mordinson testified, and we created a little
19 spreadsheet, a little matrix; is that correct, sir?

20 A. Yes.

21 Q. And in the matrix or the table that Mr. Mordinson and I
22 created, we identified several similarities in the Haifa and
23 Nipper hack. The ones that Mr. Mordinson identified were
24 the reference to Nipper, the ATR software application that
25 the defendants developed for this project, the application

1 of this to a ROM 3 EchoStar card.

2 Those are not important to your DNA analysis; is that
3 correct?

4 A. That's right.

5 Q. The bottom four are the ones that Mr. Mordinson
6 identified. Dumping the EEPROM, overflowing the buffer,
7 using the RAM ghost effect and executing code in RAM.

8 Mr. Mordinson did not talk about the index variable.

9 Did you find any reference in the defendant's Headend
10 Report that talked about the index variable and how that
11 worked with EchoStar's security system?

12 A. Yes, I did. He actually in his report identifies that
13 particular byte as the index variable and explains how it's
14 used to form the attack. So it's an integral, necessary
15 component in his report.

16 Q. Mr. Mordinson also testified that both of these attacks
17 use the ability to execute code in the RAM portion of the
18 card. Is that consistent with the demonstration that you
19 just gave us?

20 A. Right. Remember that the address that's obtained at
21 the top of the stack jumps to some code which is in RAM and
22 starts executing it.

23 Q. Now, you've had an opportunity to review the report
24 submitted by the defendant's software expert in this case,
25 Nigel Jones; is that correct?

1 A. That's right.

2 Q. And in his report Mr. Jones identifies -- he does the
3 opposite as you. He tries to identify a lot of technical
4 differences between the two hack methodologies. Did you
5 review that portion of the report?

6 A. I did. I read his whole report.

7 Q. And can you explain to the ladies and gentlemen of the
8 jury why those technical differences are not significant in
9 your opinion and in your analysis?

10 A. Sure. So let me give you a few of the things that he
11 points out. He has a page where he talks about that the
12 code in the Nipper attack and in the Haifa attack is of
13 different lengths, which, if you looked at these, you would
14 notice that.

15 The reason for that is that the Nipper one puts in a
16 whole bunch of nonimportant data before it gets to where it
17 needs to be; whereas, the Haifa one puts the shell code
18 right in the I/O buffer. So they're obviously not the same
19 program. They do it a little bit of a different way. So
20 they're not going to be the same length.

21 Another thing that he points out is that in the shell
22 code itself, the Haifa methodology writes actual assembly
23 language code to do the writing of the data from the -- when
24 it's writing the EEPROM.

25 And the Nipper methodology uses a built-in function of

1 the card, so the code is more compact and more efficient.
2 And so he uses that to try to make the case that these
3 programs are different.

4 In my opinion, if you are developing a program like the
5 one in the Headend report, which is clearly for
6 demonstration purposes because of the detail in which the
7 report explains things, you might do things more
8 pedagogically.

9 And when you're building an attack that you expect to
10 work in the real world, now you've had two years since the
11 Haifa report was written to improve on it. So that's why I
12 think it's either the same programmer learning more and
13 writing a better attack or someone else looking at the Haifa
14 report and then building their own from scratch.

15 Q. But besides these technical differences, the
16 fundamental components of each of the hack methodologies is
17 materially identical; is that correct?

18 A. That's right.

19 Q. And I understand from Mr. Jones' report that there are
20 different code lengths and different styles of coding. If I
21 understand your testimony correctly, that's no different
22 than if two people wrote the same sentence with the same
23 words but some inverted nouns and verbs and some used
24 different types of penmanship; is that correct?

25 A. That's right. It's not inconsistent with two different

1 people writing the program from the same source to have
2 different programming styles.

3 Q. Now, Dr. Rubin, after comparing the defendant's hack
4 methodology for EchoStar's security system and then the
5 Nipper hack methodology posted in December of 2000, is there
6 any doubt in your mind, based on all of your years of
7 experience and analysis in this case, that the Nipper hack
8 methodology was derived from the defendant's Headend Report?

9 A. I don't have any doubt. In fact, if two students in
10 one of my classes were to turn in these two messages as
11 their assignments, I would give them both an F because
12 there's no way that that wasn't collaboration or copying.

13 MR. HAGAN: Thank you, Dr. Rubin.

14 Pass the witness, Your Honor.

15 THE COURT: This would be cross-examination by
16 Mr. Stone on behalf of NDS.

17 MR. STONE: Thank you, Your Honor.

18 CROSS-EXAMINATION

19 BY MR. STONE:

20 Q. Good afternoon, Dr. Rubin.

21 A. Good afternoon.

22 Q. We've met before a couple times, right?

23 A. Right.

24 Q. Now, you sat through the entire trial so far, correct?

25 A. Pretty much. I missed a few minutes here and there.

1 Q. You were here when Mr. Nicolas testified, correct?

2 A. Yes.

3 Q. And you heard the suggestion that because someone
4 posted using the name Nipper in late 1998 -- I think it was
5 December of '98 -- that that was somehow proof that Nipper
6 could only have come from the Headend Report which preceded
7 that. Do you recall that?

8 A. I don't remember him saying that that was proof of
9 that, but I do remember the discussion.

10 Q. And you know from your own report that such a
11 suggestion would be incorrect, right?

12 A. What's the suggestion?

13 Q. That Nipper could only have followed the Headend Report
14 and no other source because the Headend Report preceded it.

15 A. Okay.

16 Q. Would you like to see the report?

17 A. No. I just want to understand the question.

18 Q. Let me show you Exhibit 799, which is your report.

19 A. Okay.

20 Q. That might help. What I'm focusing on is page 799-19
21 for the time being.

22 MR. STONE: And, Your Honor, I'd like to publish
23 just that portion of it.

24 THE COURT: Well, I'm not certain, Counsel. As
25 long as this is going to be, you know, published by both

1 sides, different portions of different doctor's reports, and
2 you reach that agreement, I have no concern. Normally
3 reports don't go up.

4 MR. HAGAN: That was my understanding, Your Honor.
5 I thought we had already discussed this, and each side's
6 expert reports were not going to come in or be published.

7 THE COURT: I did, too.

8 MR. STONE: I was just going to show a portion.

9 Let me do this: I can focus him on a reference,
10 and we can go from there. That will work.

11 BY MR. STONE:

12 Q. Dr. Rubin, looking at page 19 of your report, you have
13 a reference there to the first known case of an EEPROM dump
14 attributed to Swiss Cheese Productions on October 24th,
15 1998.

16 Do you see that?

17 A. Yes.

18 Q. And you have an NDS number, which is a Bates stamp
19 number for a document produced in this case, correct?

20 A. Right.

21 Q. 1760?

22 A. Right.

23 Q. And that would have been a document that you relied
24 upon to prepare your report?

25 A. Yes.

1 Q. Okay. If I could show you Exhibit 225, which is that
2 document. If you go to page 225-02, that's the -- pardon
3 me -- 225-03. That's the Bates stamp number you were
4 referencing, correct?

5 A. Right.

6 Q. And this was attached to the e-mail that is at
7 page 225-01, correct --

8 A. Yes.

9 Q. -- if you look at the document?

10 A. Yes.

11 Q. And this is a document you relied upon in reaching your
12 opinions, correct?

13 A. Right. I looked at this, yes.

14 MR. STONE: Your Honor, I would ask to publish
15 Exhibit 225 and move it in at this point.

16 THE COURT: Any objection?

17 MR. HAGAN: No, objection.

18 THE COURT: Received, and you may publish the
19 document.

20 MR. STONE: Thank you, Your Honor.

21 (Exhibit No. 225 received in evidence.)

22 BY MR. STONE:

23 Q. Looking at the page that is 225-01, this was an e-mail
24 from Ted Rose to Reuven Hasak and another gentleman, Yoni
25 Shiloh, on October 25th, 1998.

1 Do you see that?

2 A. Yes, I do.

3 Q. In reviewing the documents in this case, did you learn
4 that Mr. Rose was responsible for monitoring the Internet?

5 A. I don't remember that, but I -- sure looks like it.

6 Q. And it shows down at the bottom, No. 3, EchoStar CAM
7 dump and public keys by Swiss Cheese Productions.

8 Do you see that?

9 A. I do.

10 Q. All right. And that was the file that was attached
11 that you looked at for the purposes of your report, correct?

12 A. Yes.

13 Q. And if you could go to the next page, 225-02, please.

14 A. (Complies.)

15 MR. STONE: Could we get 225-02.

16 (Document displayed.)

17 MR. STONE: If we could zoom in a little on the
18 top.

19 (Technician complies.)

20 BY MR. STONE:

21 Q. And this was part of the text file that was posted on
22 the Internet, correct?

23 A. Right.

24 Q. And it says, "Dump from running CAM REV3.13." You
25 understood that to be from a ROM 3 version card?

1 A. That's right.

2 Q. And then if you go to the next page, please, 225-003,
3 that's the EEPROM dump that appeared on the Internet
4 sometime before October 25th, correct?

5 A. That's right.

6 Q. And do you see a phrase in there that looks familiar to
7 you from the code that EchoStar and Nagra use?

8 A. Yes, I do.

9 Q. And what is that phrase?

10 A. You're going to make me say it. All right. "Nipper is
11 a butt licker."

12 Q. And who do you understand to have put that in the code?

13 A. The creators of the card.

14 Q. The creators of the card or EchoStar? Do you know?

15 A. Well, I heard testimony yesterday that EchoStar asked
16 Nagra to put it in.

17 Q. Do you know how many employees at EchoStar had
18 knowledge of that phrase?

19 A. I do not.

20 THE COURT: Well, which portion?

21 MR. STONE: The Nipper --

22 THE COURT: "Nipper is a butt licker"?

23 MR. STONE: That phrase. That pass phrase.

24 THE COURT: Is that correct?

25 MR. STONE: That's correct.

1 THE WITNESS: What was the question? I thought
2 you asked me how many people knew, and I said I didn't know.

3 BY MR. STONE:

4 Q. And the Headend Report was completed on November 1st,
5 1998, correct?

6 A. That was the final draft, yes.

7 Q. Well, the previous draft was October 27th, 1998?

8 A. Right.

9 Q. And we know that sometime prior to October 25th, 1998,
10 there was on the Internet by the Swiss Cheese Productions a
11 dump of the EEPROM with the phrase, right?

12 A. That's correct.

13 Q. So anyone visiting that site prior to October 25th,
14 1998, would have seen that phrase in the code that was
15 posted, right?

16 A. That's correct.

17 Q. Now, is it correct that -- it's your understanding the
18 lawyers for plaintiffs found you as an expert in this case
19 by doing a Google search?

20 A. That's my understanding.

21 Q. And prior to this lawsuit, had you ever evaluated the
22 security of a conditional access system for satellite
23 television?

24 A. No, I had not.

25 Q. And the Smart Cards in question, the ROM 3 cards, are

1 considered embedded chip systems?

2 A. That's right.

3 Q. And have you ever written a software program for
4 embedded chip systems?

5 A. No, I have not.

6 Q. Or have you ever written software for any embedded
7 chip?

8 A. No.

9 Q. And am I correct, you haven't done a code review for an
10 embedded chip?

11 A. Not -- not -- no.

12 Q. And you've never been called upon to design a Smart
13 Card, correct?

14 A. That's right.

15 Q. And the ROM 3 card had an ST microchip with a Motorola
16 6505 processor?

17 A. I thought it was a 6805.

18 Q. 6805. I'm sorry, you're right. Is that correct?

19 A. Yes.

20 Q. Have you ever written a program for an ST microchip?

21 A. No, I have not.

22 Q. Have you ever written a program for a 6805 processor?

23 A. No.

24 Q. And I believe you testified that you're not even
25 familiar with the assembly language that's used in the chip

1 in the ROM 3 card, correct?

2 A. I wasn't before this case.

3 Q. In fact, the first time you ever reviewed assembly
4 language code was part of your review of the code on
5 March 28 -- 26th, excuse me, of 2008, correct?

6 A. That's not correct. I've been working on assembly code
7 for years, and I taught a course with students working on
8 assembly code.

9 Q. Was that the first time you'd ever reviewed assembly
10 code for a Motorola 6805 processor?

11 A. Yes.

12 Q. And that has different operations codes than other
13 processors, correct?

14 A. Yeah.

15 I should explain that every chip has its own flavor of
16 assembly language. The hard part is understanding
17 conceptually how to do assembly code. That's what we teach.

18 Once you've worked with a particular assembly,
19 understanding a different one is the matter of the syntax.
20 Like are you copying things from the first parameter to the
21 second, or the other way around?

22 But the question is -- in IBM chips, for example, if
23 you see loader register, for example, and then there are two
24 values, you copy the second value to the first. And in the
25 Motorola assembly language, you copy the first to the

1 second.

2 So when you work under an assembly language, you need
3 to get the reference manual and you need to understand what
4 the syntax is. But once you've done a lot of assembly
5 language programming -- and in my case I taught assembly
6 language programming -- looking at a new one is just a
7 matter of looking a few things up.

8 Q. Now, when you did the code review, though, you spent
9 about an hour and a half reviewing the ROM 2 code before you
10 realized you were reviewing the wrong code, correct?

11 A. I don't think it was that long, but it was a bit of
12 time where I was looking at the ROM 2.

13 Q. And that's because you didn't have familiarity with the
14 code used in this particular chip, correct?

15 A. No. Actually the reason that I started looking at
16 ROM 2 was that I was given a machine with all the code on a
17 hard disk but no information about the file system
18 structure. So I didn't know which code was where.

19 And once I started looking at it -- and I think I
20 noticed a comment in one of the files -- or something
21 triggered that I might be looking at the wrong code. And
22 then I went back and figured out where the ROM 3 code was.

23 Q. Now, you've never set out to determine how cheaply or
24 how efficiently one could extract the code from a NagraStar
25 Smart Card, have you?

1 A. No.

2 Q. Am I correct, you did not read the deposition of
3 Christopher Dalla in this case?

4 A. That's right.

5 Q. Do you know who Christopher Dalla is?

6 A. I don't.

7 Q. And so therefore you don't know what Mr. Dalla
8 testified to, about when he obtained the ROM 2 and ROM 3
9 code, correct?

10 A. I have no knowledge of that.

11 Q. Have you ever heard of the DISH Plex piracy group?

12 A. Just in court.

13 Q. Before coming to court here this week and last week,
14 had you ever heard of that group?

15 A. I think you asked me that same question in one of my
16 depositions, but other than that...

17 Q. Have you ever heard of the EROM hacking forum that was
18 hosted by the DISH Plex piracy website?

19 A. Same answer.

20 Q. Did you read any of the testimony about the DISH Plex
21 piracy lab in Thunder Bay, Ontario that was used to extract
22 code from EchoStar access cards?

23 A. No.

24 Q. Did you read the deposition testimony of Mr. Pilon,
25 who's one of the plaintiff's informants, about the DISH Plex

1 piracy lab?

2 A. No.

3 Q. Did you review any documents that plaintiffs provided
4 you about information they had about a piracy lab in Thunder
5 Bay, Ontario?

6 A. No.

7 I should mention that my scope of my work was to look
8 at the Haifa report and the Nipper posting and to draw a
9 comparison, and not to read a lot of depositions that aren't
10 related to that.

11 Q. Did you find out information about ROM dumps of the
12 ROM 3 card that were available on the Internet prior to
13 December 2000?

14 A. I found information that there were some fragments
15 available.

16 Q. Did anyone provide you documents that showed you that
17 there was an entire commented disassembly of the ROM 2 code
18 on the Internet going back to early 2000?

19 A. No. Everything that I looked at in this case related
20 to the ROM 3.

21 Q. And did you analyze any piracy devices before being
22 retained in this lawsuit?

23 A. No.

24 Q. And did you conduct an investigation to specifically
25 determine how much of the ROM code of either ROM 2 or ROM 3

1 were on the Internet prior to December 2000?

2 A. No.

3 Q. And you understand that defendant's experts undertook
4 that exercise?

5 A. Yes.

6 Q. And that's not something that you attempted to do or to
7 rebut; is that correct?

8 A. No, that's right. I actually relied on Michael Barr's
9 report, B-A-R-R, which deals with that.

10 Q. Now, you're the founder and president of Independent
11 Security Evaluators, correct?

12 A. Yes.

13 Q. And your firm is expert in software, reverse
14 engineering of secure products?

15 A. Yeah, any software.

16 Q. And occasionally you've been hired to do third-party
17 evaluations, which means breaking into someone else's system
18 to evaluate it, right?

19 A. No. We have been hired before to do third party, and
20 we only do that work with the stipulation that it's with the
21 cooperation of the third party. So we've done it in the
22 case where somebody wants to buy a product and they want us
23 to make a judgment on how secure it is.

24 Q. Well, your company has done some reverse engineering
25 that is unknown to the party whose product or system you

1 were reverse-engineering, correct?

2 A. I should point out that's getting into the government
3 work that I'm not supposed to talk about.

4 Q. And haven't you done that also in the private sector,
5 sir?

6 A. It's under circumstances that I'm, again, not supposed
7 to talk about.

8 Q. Have you ever attacked a product just to see if you
9 could do it?

10 A. Yes.

11 Q. And you've done that more than once, correct?

12 A. Yes.

13 Q. And you consider yourself a hacker, right?

14 A. I consider myself a White Hat hacker. There's a
15 distinction between a malicious hacker, who is what's been
16 called in this case a pirate, versus a tinkerer who likes to
17 play with computers and look at where things break.

18 Q. Are there other companies that you're aware of that
19 offer hardware or reverse-engineering services in the
20 private sector?

21 A. Is it "or"? Hardware or reverse engineering?

22 Q. Hardware or software.

23 A. Yes.

24 Q. Have you heard of Semiconductor Insights?

25 A. No.

1 Q. Are you familiar with Michael Strizich,
2 S-T-R-I-Z-I-C-H?

3 A. No.

4 Q. And have you ever or anyone at your firm ever presented
5 at the Black Hat conference?

6 A. Yes. Dr. Charlie Miller, who works for me, has
7 presented there.

8 Q. And have you attended Black Hat conferences?

9 A. I've never been.

10 Q. Does the term "Black Hat" refer to malicious hackers or
11 pirates?

12 A. The term Black Hat sometimes is used to refer that way.

13 It was pointed out earlier in this case that the term
14 Black Hat has broadened over time. The Black Hat conference
15 is as likely to include FBI agents, for example, as it is
16 pirates. It's a place where all kinds of people get
17 together to discuss issues around breaking system and
18 hacking.

19 Q. Well, it's a conference of folks who are involved in
20 the security business, right?

21 A. Right.

22 Q. And didn't you testify that the Black Hat conference
23 was the opposite of what the name implies, that it is not
24 malicious hackers?

25 A. No. Malicious hackers go there. It's not intended for

1 malicious hackers. I don't think those kind of meetings are
2 held in the public.

3 Q. Why is the name Black Hat associated with a conference
4 of security researchers?

5 A. I think its history. I think that there used to be a
6 bunch of people that you would be more likely to call real
7 hackers, malicious hackers, that would get together. And I
8 think that evolved over time into a legitimate conference.

9 Q. Do you know where the name Black Hat originates with
10 respect to computer security?

11 A. I do not.

12 Q. And the Black Hat conferences would be where people
13 gather to discuss hacking techniques and reverse engineering
14 and countermeasures, correct?

15 A. Yes.

16 Q. And so if somebody suggested that the term Black Hat in
17 conjunction with hardware or software security engineers
18 necessarily means something malicious, that would be
19 incorrect, right?

20 A. Are you talking about that specific conference?

21 Q. Yes.

22 A. Yes.

23 Q. Now, let me show you, if we could, Exhibit 809.

24 MR. STONE: This is in evidence.

25 (Document displayed.)

1 BY MR. STONE:

2 Q. Okay. Now, you've seen this article before, I take it?

3 A. I believe you showed it to me at my deposition.

4 Q. And looking at the first page, you're familiar with
5 Mr. Kuhn, right?

6 A. Yes. Marcus Kuhn is a well-known security researcher.

7 Q. And he works at the laboratory at the University of
8 Cambridge, the Smart Card laboratory?

9 A. I'm not sure if he's still there, but he used to.

10 Q. And another professor at Cambridge was Ross Anderson?

11 A. Right.

12 Q. And Mr. Kuhn came up with techniques that were
13 publicized to work on the ST micro family of
14 microcontrollers or chips, correct?

15 A. I believe that's right. I'm hedging because I'm not
16 sure if it's that particular chip or not.

17 Q. Well, the ST microchips are the ones used in the
18 EchoStar access card, correct?

19 A. That's right.

20 Q. And you respect Mr. Kuhn's opinions in the area of
21 extracting code from embedded chips, right?

22 A. I what?

23 Q. You respect --

24 A. I respect, yes, definitely.

25 Q. -- his opinion about extracting code from embedded

1 chips?

2 A. Yes.

3 Q. And if I could direct your attention to the first page,
4 at the bottom, it's in small type.

5 A. Yes.

6 Q. This paper was presented at a USENIX workshop?

7 A. Right.

8 Q. And that was an organization you were a member of?

9 A. Yes. I still am.

10 Q. And were you on the board of USENIX as well?

11 A. Yeah. I spent four years on the board of directors.

12 Q. Were you on the board when this article was published?

13 A. I think so. I think I was on the board until 2000.

14 Q. And you would agree, USENIX is a legitimate
15 organization?

16 A. It's the top academic organization for systems
17 security.

18 Q. And were you present when this paper was presented at
19 the USENIX proceedings in May of 1999?

20 A. No. USENIX has about 12 workshops every year in
21 various different areas. And I never miss the USENIX
22 security one, but I don't go to all the workshops.

23 Q. If you look at the paragraph entitled Abstract.

24 A. Right.

25 Q. It says, "We describe techniques for extracting

1 protected software and data from Smart Card processors.
2 This includes manual microprobing, laser cutting, focused
3 ion beam manipulation, glitch attacks and power analysis."

4 Do you see that?

5 A. I do.

6 Q. And those were all well known as of May 1999, correct?

7 A. I would imagine they became well known after this paper
8 was published.

9 Q. Did Mr. Anderson and Mr. Kuhn publish papers before
10 this that discussed some of those similar methods?

11 A. I don't know.

12 Q. And it says, "Many of these methods have already been
13 used to compromise widely fielded conditional access
14 systems, and current Smart Cards offer little protection
15 against them."

16 Did you notice in this article that they went on to
17 provide countermeasures to protect Smart Cards against these
18 types of attacks?

19 A. I would have to look again. I'm only familiar with
20 this paper from my deposition.

21 I imagine that USENIX's paper would require you to talk
22 about that in it, so I have no doubt that they talk about
23 that. I just can't say without looking.

24 Q. Okay. Do you know whether Kudelski Nagra implemented
25 any of the countermeasures identified in this article?

1 A. I don't know which countermeasures were identified in
2 the article.

3 Q. Did you read any of the references at the end of the
4 article as part of your work in this case?

5 A. No. I only looked at this briefly in my deposition.

6 Q. Now, your company hacked the Apple iPhone, correct?

7 A. That's right.

8 Q. And it was three weeks within its release?

9 A. Something like that.

10 Q. And your company was the first to hack the iPhone,
11 correct?

12 A. Well, other people had been able to unlock it, which is
13 different from what I consider hacking it. But we were the
14 first to be able to run shell code on it.

15 Q. And it was your idea to hack the iPhone, right?

16 A. Yes.

17 Q. And you thought it would be good for your company if
18 your employees were able to hack the iPhone, right?

19 A. Yes.

20 Q. And one of your purposes in hacking the iPhone was so
21 that your company would get credit for doing it, right?

22 A. Yes.

23 Q. And that included publicity, right?

24 A. Right.

25 Q. And publicity generates business, doesn't it?

1 A. It does.

2 Q. And when your company hacked the Apple iPhone, what was
3 the vulnerability you exploited?

4 A. There was a buffer overflow vulnerability.

5 Q. And that's one of the most common forms of attack in
6 computer systems, correct?

7 A. That's right.

8 Q. And anybody with any expertise in computer security
9 would know about buffer overflow attacks, correct?

10 A. Absolutely.

11 Q. In fact, a lot of them were publicized in the 1990's,
12 correct?

13 A. That's right. They get publicized almost every day.

14 Q. Is that why your company first thought of using that
15 kind of an attack on the iPhone, to crack it?

16 A. If you're going to try to break a system, I think
17 buffer overflow is the first thing you might try.

18 Q. And that was, in fact, the first thing that Mr. Miller
19 of your company tried on the iPhone, right?

20 A. That's right.

21 Q. And he was able to extract the binaries for the
22 application code from the iPhone as part of his hack
23 methodology, right?

24 A. Actually, he used other tools to do that.

25 Q. What tools did he use to extract the binaries?

1 A. I believe -- oh, I'm not supposed to say that.

2 I think he used the Jailbreak program.

3 Q. And what was Jailbreak?

4 A. It was a tool that ran on the iPhone that would allow
5 you to take binaries off of it.

6 Q. Was it a malicious program?

7 A. Depends how it was used. It was a tool, and all it did
8 was let you take binaries off.

9 Q. Was it an official Apple program?

10 A. No.

11 Q. Was it written by hackers?

12 A. I don't know.

13 Q. Did you use any of that work or any of that program in
14 your work?

15 A. Dr. Miller did use Jailbreak, if that's what you're
16 asking. I already said that. Do you mean me or ISE?

17 Q. You personally, sir.

18 A. No.

19 Q. And the binaries are maintained in the firmware of the
20 Apple iPhone, correct?

21 A. Right.

22 Q. Could you explain what firmware is?

23 A. Sure. In a device firmware is a layer between the
24 software and the hardware. So the -- if you think about an
25 iPhone, you've got your e-mail application, your phone

1 application, stocks; these are applications. You've got the
2 hardware, which is physical wires. And then what's in
3 between is kind of -- in a computer you would call the
4 operating system, is called firmware in a device like the
5 iPhone.

6 Q. And getting the machine code out of the iPhone was the
7 first step to disassembling it, to understanding the code,
8 correct?

9 A. That's right.

10 Q. And that was the process that your company did to hack
11 the iPhone?

12 A. Right. You have to learn something about it before you
13 can hack it.

14 Q. And how was your company able to extract the machine
15 code from the firmware of the iPhone to discover the buffer
16 overflow vulnerabilities?

17 A. That's not exactly how we did it.

18 Q. Well, didn't you use fuzzing to extract the binaries?

19 A. I'm going to help you out a little bit. No. We used
20 fuzzing in a different way. But we extracted the binaries
21 simply using Jailbreak.

22 Q. And how is the buffer overflow vulnerability discovered
23 by Mr. Miller?

24 A. He discovered it with fuzzing.

25 Q. Fuzzing is where you send invalid inputs to the

1 computer and you get the results back and you're able to
2 determine whether the buffer will overflow, correct?

3 A. It's a -- fuzzing is a trial-and-error concept. It's a
4 little broader than just what you described.

5 In general, anytime you -- like, remember that website
6 that I showed you when I taught you what a buffer overflow
7 was? If you were to take a web page and just start putting
8 in all kinds of crazy random things and see if something
9 good happens -- like you get a big raise -- then that's
10 fuzzing.

11 But in order to determine whether or not there's a
12 buffer overflow, there's a very directed way that you would
13 fuzz. And that's what Dr. Miller did when he was hacking
14 the iPhone.

15 Q. And so fuzzing can be used to discover a buffer
16 overflow vulnerability, right?

17 A. Yes.

18 Q. And your company uses fuzzing to evaluate secure
19 products, right?

20 A. It's a tool that we use.

21 Q. And fuzzing has been around since at least 1989,
22 correct?

23 A. Yeah. Longer than that. I mean, the term fuzzing
24 hasn't been around longer than that, but the concept has
25 been.

1 Q. And in opening statement, counsel said your firm had
2 been retained by Apple. Did Apple retain your firm to hack
3 the iPhone?

4 A. No.

5 Q. Did you have Apple's permission to hack the iPhone?

6 A. No.

7 Q. Let me be clear, then. Before you hacked the iPhone,
8 you did not call up Apple's general counsel and ask them if
9 it was okay to do that; is that right?

10 A. That's right.

11 Q. And you didn't have a brief prepared before you hacked
12 the iPhone, did you?

13 A. Yeah, we did.

14 Q. Did you contact anyone at Apple for their permission
15 before you published any results?

16 A. Not for their permission. We basically told them about
17 it before we published the result.

18 Q. And at the time your company hacked the iPhone, there
19 were lots of malicious hackers trying to hack the iPhone,
20 correct?

21 A. That's right.

22 Q. And so after you hacked the iPhone, you didn't maintain
23 the results in some encrypted form on a secure computer; is
24 that correct?

25 A. No. That's how we keep all of our information.

1 Q. And how long did you keep it encrypted on the computer?

2 A. Basically -- what do you mean by "information"? The
3 codes that we wrote for the hack or the report that we
4 wrote?

5 Q. All the information relating to how you conducted the
6 hack and the code that you wrote to execute the hack.

7 A. So we never released the code except Charlie gave a few
8 snippets of it in the talk that he gave. And the report we
9 released I think three weeks -- I don't remember the timing
10 exactly, but we gave Apple three weeks before we went public
11 with this.

12 Q. And was Apple happy that you had hacked the iPhone when
13 you had first told them?

14 A. No.

15 Q. Let me show you Exhibit 800, please. Okay. Do you
16 recognize Exhibit 800?

17 A. Yes, I do.

18 Q. Is this the first report that your company published on
19 hacking?

20 A. I think this was our only report about it. I don't
21 remember another one.

22 Q. Well, do you recall a presentation at the Black Hat
23 conference that Mr. Miller gave?

24 A. Right, yes.

25 MR. STONE: Your Honor, I would move Exhibit 800

1 at this time.

2 THE COURT: Any objection?

3 MR. HAGAN: No objection.

4 THE COURT: 800 is received.

5 (Exhibit No. 800 received in evidence.)

6 BY MR. STONE:

7 Q. All right. Now, this report is entitled "Security
8 Evaluation of Apple iPhone," correct?

9 A. That's right.

10 Q. And that's the same thing as hack?

11 A. No. That's not the same thing as a hack.

12 Q. Did you ever use the term "hack" with respect to what
13 you did to the iPhone in any publications?

14 A. I would imagine. I don't have any problem calling it
15 that.

16 Q. Now, look at the bottom of the second page, please,
17 sir -- 800, Page 2 at the bottom.

18 It says: "Apple was notified of these findings
19 including detailed technical documentation on July 17th.
20 While this paper serves to highlight our findings, we will
21 not release the remaining technical details until
22 August 2nd. This delay is provided in order to give Apple
23 sufficient time to produce patches so that hackers cannot
24 take advantage of these vulnerabilities." Right?

25 A. Right.

1 Q. You gave Apple two weeks, approximately, to get a patch
2 together?

3 A. There was more than that. We actually wrote a patch
4 and gave it to them as well.

5 Q. Isn't it true, Dr. Rubin, that your company was going
6 to publish this information about hacking the iPhone whether
7 Apple issued a patch or not?

8 A. That's right. We were going to put the patch on our
9 website and direct people to it and then release it. Part
10 of the --

11 Q. Would that be true if Apple objects?

12 A. Yes. Let me elaborate a little bit. Can I?

13 Q. Well, your counsel will have a chance --

14 A. Sure.

15 Q. -- to go into it more. But I wanted to find out if, in
16 fact, you would have done it even if Apple had objected.

17 A. Yes, you often have to do this in order to be able to
18 work in this area because everybody is going to object to
19 having you release information, even if you do it
20 responsibly.

21 Q. We'll get to that in a minute. But if you would go to
22 Page 800-5. Now, Apple did not approve this publication,
23 correct, that we're looking at, Exhibit 800?

24 A. They did not.

25 Q. And looking at Page 5 where it says "vulnerability

1 analysis"?

2 A. Right.

3 Q. That's describing the vulnerabilities that were found
4 in the iPhone that enabled the hack, correct? If you go
5 down at the bottom of that paragraph, it even identifies how
6 you found the vulnerability, right?

7 A. Actually, what I was doing is identifying the software
8 in which we found vulnerabilities, but it's not identifying
9 the vulnerabilities.

10 Q. And it's saying, "The vulnerability we discovered and
11 exploited was found in Mobile Safari using fuzzing,"
12 correct?

13 A. Correct.

14 Q. So you identified the method that was used?

15 A. Right.

16 Q. And what are attack scenarios underneath that?

17 A. Oh. So these are -- attack scenarios, basically when
18 there's a vulnerability, you talk about when the attack
19 happen. So in our example, we talked about since the iPhone
20 has WiFi, which is wireless networking, if you were to go to
21 a cyber cafe and get on that cyber cafe's WiFi network and
22 the owner of that network was malicious, they could hack
23 your iPhone. So we're describing different scenarios in
24 which the attack is possible.

25 Q. And if you go to Page 800-6, there's a paragraph

1 entitled, "Black Box Exploitation."

2 A. Right.

3 Q. At the top it says: "Once a vulnerability has been
4 identified, the next step is developing a functioning
5 exploit." What is a "functioning exploit"?

6 A. A functioning exploit is when the attack actually
7 works. Let me differentiate between two phases. So the
8 first phase of doing a security analysis like this is to
9 find out where there's a vulnerability; for example, in this
10 case, the RAM ghost effect, the index, all of these
11 vulnerabilities that we talked about. You can stop there.
12 That could be okay. Now, you know the thing is vulnerable,
13 or you can take it all the way and actually build an attack
14 that exploits that, and that would be the message that you
15 guys saw.

16 Q. And that was part of your security evaluation, as you
17 called it, right?

18 A. We did build the attack.

19 Q. What is the paragraph entitled "black Box Shell Code
20 Development" refer to?

21 A. Well, when you attack a system, you have to write code
22 that will run on that system. So after you discover a
23 vulnerability and exploit it, the next step is to run the
24 shell code. In the Haifa Report, the shell code was what
25 piece that's in the I/O buffer that you have to jump into

1 later. In our case, we also wrote shell code to build an
2 example exploit.

3 Q. If you could go to Exhibit 802, please. Is 802 a copy
4 of the presentation Mr. Miller gave at the Black Hat
5 conference on August 2nd, 2007?

6 A. Looking at the first few pages, I would say that's what
7 it appears to be.

8 Q. And did Apple approve of this presentation?

9 A. No.

10 MR. STONE: I would move Exhibit 802 at this time,
11 Your Honor.

12 THE COURT: Any objection?

13 MR. HAGAN: No objections, Your Honor.

14 THE COURT: Received.

15 (Exhibit No. 802 received in evidence.)

16 BY MR. STONE:

17 Q. Okay. If we could look at the first page, the talk was
18 titled, "Hacking Leopard: Tools and techniques for
19 attacking the newest Mac OS 10," right?

20 A. Right.

21 Q. And if I recall correctly, originally Mr. Miller was
22 gonna do a talk on hacking Leopard, which was the new
23 operating system, not the iPhone?

24 A. That's right.

25 Q. And you folks had obtained a prototype or beta copies

1 of the Leopard software under a developer's agreement,
2 correct?

3 A. Well, actually, that's what I thought in my deposition.
4 And after you asked me this, I went back and talked to
5 Dr. Miller. And he told me that while he did click through
6 the agreement, he never downloaded it.

7 Q. Downloaded which agreement?

8 A. Leopard. He never downloaded the Leopard code.

9 Q. I'm sorry?

10 A. Dr. Miller told me that he never actually downloaded
11 the Leopard code.

12 Q. Okay. But your testimony was that you had signed up
13 under the developer's program so you get access to the code
14 to hack it, right?

15 A. You asked me about that, and at the time I thought that
16 he had done that, and I asked him, and he told me that he
17 went to the site to download Leopard, clicked through the
18 copyright agreement page, and then when he saw that he had
19 to download it and burn it to a disc and he didn't have a
20 disc drive that he could use to burn it; he didn't download
21 it.

22 Q. But that was his intention, correct?

23 A. Yes.

24 Q. So he was going to pretend to be a developer so he
25 could get the software so that he could hack it, unbeknownst

1 to Apple, right?

2 A. No, he was not going to pretend to be a developer. He
3 was a developer, and he had a lot of reasons why he was
4 interested in loading an early version of Leopard, including
5 wanting to run it himself as his operating system.

6 Q. But his intent was to hack it and write about it?

7 A. When Charlie works at something, he's always very
8 curious. He wanted to load it, and I'm sure he would have
9 taken a whack at it because that's what he likes to do. But
10 he didn't actually download it.

11 Q. Let's go to Page 8 of Exhibit 802.

12 This has two quotes from Apple marketing materials,
13 correct?

14 A. Oh. My Page 8 is not that.

15 Q. Should say "Apple" at the top?

16 A. Oh, that's my Page 7.

17 Q. I don't know why that would be.

18 A. So there's a "7" here, but 802-8.

19 Q. The one at the bottom?

20 A. My bad. Um -- right.

21 Q. And was one of the reasons you hacked the iPhone to
22 keep Apple honest in its representations in the marketplace?

23 A. Yes.

24 Q. Now, if you could go to Page -- I think it's 20 -- if
25 you give me one second, I'll tell you -- Page 20.

1 Page 802-20, sir.

2 A. Right.

3 Q. Okay. It says, "They make exploitation fun." Who's
4 the "they"?

5 A. I would imagine he was referring to Apple.

6 Q. And Mr. Miller is an employee of your company?

7 A. Yes, he is.

8 Q. And do you supervise his work?

9 A. I don't supervise him very closely, but he reports to
10 me.

11 Q. You're the president of the company?

12 A. I am.

13 Q. And you have the ability to stop presentations like
14 this if you want to stop them?

15 A. Yes, I do.

16 Q. Now, on the slide that says "They make exploitation
17 fun," Mr. Miller revealed that Apple doesn't randomize
18 anything. What he's talking about is they don't randomize
19 certain portions of memory, correct?

20 A. That's correct.

21 Q. That makes it easy to hack the Apple product. Is that
22 what he's saying?

23 A. Right. So basically one of the countermeasures that's
24 commonly adopted for buffer overflow attacks is to take
25 memory and randomize it, 'cause if you randomize the memory

1 and you're trying in our example to get to the index
2 variable, you're not going to be able to go sequentially and
3 get there. So this is a countermeasure in our cat-and-mouse
4 game against hackers in security.

5 Q. But Apple hadn't made that countermeasure at the time
6 this was published, correct?

7 A. What do you mean? In what?

8 Q. In their products. Where he says Apple doesn't
9 randomize, that's an indication they had not done it,
10 correct?

11 A. In some products. They did it in some.

12 Q. Where Mr. Miller wrote, "The heap is executable," code
13 can be executed in the heap, H-E-A-P, correct?

14 A. The heap is in RAM.

15 Q. And that's similar to the stack or RAM that we looked
16 at for the ROM 3 card, correct?

17 A. Right.

18 Q. Now, if you could go to Page 23, please, of
19 Exhibit 802 -- actually, I'm sorry -- if you go to
20 Page 22, it will show the cover slide.

21 A. Yes.

22 Q. This is the page -- 802-22 is the iPhone details,
23 correct?

24 A. Right.

25 Q. And then the rest of this talk gave the details of the

1 iPhone hack, right?

2 A. Right.

3 MR. STONE: If you go to the next slide, please.

4 BY MR. STONE:

5 Q. This slide is entitled, "How to find a Mac OS 10
6 0-day." Do you see that?

7 A. Yes.

8 Q. And a 0-day is a vulnerability that is not generally
9 known, correct?

10 A. That's right.

11 Q. And a 0-day can also mean a vulnerability for which no
12 patch has been issued, correct?

13 A. Right.

14 Q. And this slide shows one way to find a vulnerability in
15 the Mac operating system, correct?

16 A. That's correct.

17 Q. So that Mr. Miller was saying, find some open source in
18 the Mac product, read the change log for that software, find
19 a good bug, and then, he says, "profit" exclamation point.

20 A. Right. This information here is not news to the Black
21 Hat community. This is a standard way. This is kind of a
22 set-up slide that he's setting up the rest of it.

23 And in a deposition I was asked about this word
24 "profit." Did Dr. Miller want to profit from this? And
25 afterwards I remembered that this is a reference to the

1 classic all-time security paper called "Smashing the Stack
2 for Fun and Profit," which is the first paper to kind of
3 show how a buffer overflow can work, and it's the one that
4 we always teach from and cite. So he put the "profit" in
5 this, speaking to an audience of people who have read this
6 paper, to try to get a laugh. He wasn't actually looking to
7 profit.

8 Q. Didn't you say at your deposition that you wished
9 Mr. Miller had not put that in the presentation?

10 A. That's right. Like I said, after my deposition I
11 remembered that that referred to that paper.

12 Q. But at your deposition you didn't recall that from the
13 time that Mr. Miller gave his report, right?

14 A. That's right.

15 Q. And that was in August 2007?

16 A. The deposition was in October, I think.

17 Q. No -- and the presentation was in August.

18 A. Right.

19 Q. So a few months apart?

20 A. Right.

21 Q. Now, you can't guarantee that no malicious hacker would
22 follow these steps to exploit a vulnerability in the Leopard
23 or Mac's OS X operating system, correct?

24 A. I can pretty much guarantee that they would. This is
25 the standard way that people do it.

1 Q. Let's go to Page 802-25, please.

2 And again, this is a presentation that Apple did not
3 want you to give, right?

4 A. I don't know that. First, I didn't give it, but you're
5 referring to ISE -- I'm assuming that he's referring by
6 "you" to plural, my company, as opposed to me. There were
7 people from Apple in the audience, and Dr. Miller told me
8 that after his presentation they came up and had a friendly
9 conversation, so I don't know that Apple necessarily was
10 upset. Their iPhone was more secure when we were done.

11 Q. Let's talk about that. Look at Page 802-25. This is
12 entitled "The Vulnerability."

13 A. Right.

14 Q. And this is the process where Mr. Miller explains how
15 one could use the heap to overflow and write an exploit to
16 hack the iPhone, right?

17 A. Right.

18 Q. And he gives a number of total bytes that can be
19 overflowed, correct?

20 A. That's correct.

21 Q. And he discloses that he found it the old-fashioned
22 way, fuzzing, correct?

23 A. That's right.

24 Q. If you go to the next page, 802-26.

25 Do you know what Mr. Miller meant by his title at the

1 top? "Another change log entry, another Safari 0-day"?

2 A. Let's see. Change logs are things in software that
3 will show you when things change. It's what it sounds like.
4 It's a log of a change. And they're often used by hackers
5 to figure out what the bug was that caused that change. If
6 you get your access to the change logs for a product, you
7 can then pretty easily figure out how to attack previous
8 versions of the product. And then people who haven't
9 upgraded to the latest release are going to be vulnerable to
10 that.

11 Q. Where it says "Another heap overflow" and it has some
12 script language, that's Mr. Miller of your company giving an
13 example of an unchecked input that overflows memory in the
14 iPhone, correct?

15 A. Right.

16 Q. Go to the next page if you will, please, Page 27 of
17 Exhibit 802.

18 A. Okay.

19 Q. This is entitled, "Black Box Exploitation of the
20 iPhone." What is a black box exploitation?

21 A. So the point of this slide is that you -- when you go
22 to attack something -- sometimes Mr. Mordinson, for example,
23 had access to the ROM code. So we call that a white box
24 evaluation because you get all of the code in the system.
25 So it's a lot easier because when you have the code, you can

1 figure out where all the bytes go, and you can perform your
2 attack.

3 We only had an Apple iPhone. We had to treat it as a
4 black box where we couldn't see inside it. We didn't know
5 what it was doing, so that was a high-level description of
6 the kind of attack that this was.

7 Q. And then you gave the details of the exact process,
8 correct?

9 A. I'd have to look at the slides to see if he gave all
10 the details.

11 MR. STONE: Let's look at Page 802-30, please.

12 (Document displayed.)

13 BY MR. STONE:

14 Q. This is entitled, "A good" -- and "good" is in
15 quotes -- "crash." Do you see that?

16 A. Yes.

17 Q. What Mr. Miller meant by that is by causing the iPhone
18 to crash, you can obtain data that can be used to hack it,
19 correct?

20 A. That's right.

21 Q. And --

22 A. Some crashes give you no information. And what he was
23 saying here is that this particular crash was useful to him.

24 Q. In this publication that Mr. Miller made, he described
25 how to cause the phone to crash, correct?

1 A. That's right.

2 Q. And what do you understand Mr. Miller to have meant at
3 the bottom where he said "old school heap overflow"?

4 A. What he was kind of -- this was very much a
5 tongue-in-cheek presentation where he was preaching to the
6 choir at Black Hat, and he was basically saying that nothing
7 is fancy here. This isn't any new technique. We're using
8 heap overflow.

9 Q. And did he have Apple's permission to disclose that
10 vulnerability prior to this?

11 A. At this point, Apple had already acknowledged us and
12 issued a patch.

13 Q. But they had not approved him disclosing these details;
14 isn't that true, sir?

15 A. That's right.

16 MR. STONE: Now, go to Page 802-31, please.

17 (Document displayed.)

18 BY MR. STONE:

19 Q. This is entitled, "Controlling the inputs," right?

20 A. That's right.

21 Q. And this gives some pretty specific details that would
22 give one the ability to write four bytes anywhere in the
23 memory of the iPhone, correct?

24 A. Right.

25 Q. And then if you go to the slide at Page 33 entitled,

1 "Getting PC"?

2 A. Okay.

3 Q. What does that mean, "Getting PC"?

4 A. PC is the program counter, and that shows you where you
5 are in the program.

6 Q. Is that similar to an index variable?

7 A. No.

8 Q. Is it similar to a stack pointer?

9 A. Yes.

10 Q. It says, "We chose to overwrite a saved return address
11 on the stack." What does that mean?

12 A. Okay. So I guess I'm not gonna get away with not
13 telling you what a stack is. Sorry.

14 When a program is in memory in a computer -- or in a
15 device for that matter -- the program is composed of
16 subroutines and functions. These are -- think of them as
17 modular pieces of code that do one thing.

18 So, for example, let's say that you're gonna write a
19 program to calculate, you know, the trajectory of an
20 airplane, and one of the things that you need to be able to
21 do is compute a factorial. That's a function that you need.
22 So instead of embedding all the code for the factorial,
23 every place that you need the factorial function you're just
24 gonna write a factorial function, and you're gonna call it
25 whenever you need it.

1 Now, the way the computer works is, when you call a
2 function when you're running, say, okay, it's time -- I need
3 a factorial, call factorial, it takes what's called a stack
4 frame. It takes a bunch of memory and allocates it on the
5 stack. And the stack is just a place in memory where you're
6 going to put some things every time you call a function.

7 And a computer is a very simple device. It just does
8 what it's told.

9 And after you call the factorial, now you want to
10 go back to what you were doing before you called the
11 factorial. How are you going to know where to go in the
12 program to get back to where you were before you called the
13 factorial? One of the things you put in the stack frame in
14 memory is a return address. That's the address in memory
15 right after where you previously were before you called the
16 function.

17 And so factorial runs, gets the result, and then looks
18 in the return address to see now where do I need to go back
19 to.

20 Now, a common technique when you're hacking is to say,
21 "Let's write a function." It will get put on the stack, and
22 if one of the variables or buffers in that function -- say
23 that factorial has X and Y, right? If X doesn't have bounce
24 checking on it, you can write a really long value into it --
25 just like I did earlier -- and start overriding frame

1 pointers on the stack. And you can change the return
2 address back to your shell code instead of going back to the
3 program and doing legitimate things.

4 Q. So was Mr. Miller acting maliciously in using this
5 shell code and a return address on the stack?

6 A. No.

7 Q. Now, when you overwrite a safe return address, that's
8 not the normal function of the program, is it?

9 A. That's a hack. That's what hackers do.

10 Q. And this slide goes on to reveal that it was found by
11 fuzzing from SP. Does that mean "stack pointer"?

12 A. Yes.

13 Q. What does that mean where it says "setting the other
14 register to a stack value"? That is where you want it to
15 return to your shell code?

16 A. I'm not sure what he was doing with the other stack
17 value.

18 Q. And those are instructions below that provide detail on
19 how to do that on the Apple iPhone?

20 A. I think that this is data. It's a dump of the crash
21 log.

22 Q. Which would help you in hacking the iPhone?

23 A. Right.

24 Q. And then executing on the stack down below. Do you see
25 that?

1 A. Right.

2 Q. That's disclosing that code is executable, the stack,
3 which is similar to the RAM in the ROM 3 card, right?

4 A. Right.

5 MR. STONE: Now, go to Page 34, please.

6 (Document displayed.)

7 BY MR. STONE:

8 Q. By the way, did the Nipper attack also overwrite part
9 of the stack?

10 A. Yes, it did.

11 Q. And it had a shell code that had a return address on
12 the stack?

13 A. No, it didn't use a return address in the same sense
14 that most programs do. But it had an exception handling
15 address which pointed back to the middle of the stack.

16 Q. Now, go to Page 34 where it says, "Search for your
17 shell code." What is shell code? Is that the malicious
18 code?

19 A. Shell code is the code written by the attacker to do
20 whatever it is that the attacker wants to do.

21 Q. And did the Nipper attack have shell code in it?

22 A. Yes. The Haifa and the Nipper -- the Haifa put the
23 shell code in the I/O buffer, and Nipper put the shell code
24 on the stack.

25 MR. STONE: Now go to Page 38 of

1 Exhibit 802, please.

2 (Document displayed.)

3 BY MR. STONE:

4 Q. Is this -- this is entitled, "Make it happen," right?

5 A. Right.

6 Q. Is this the actual exploit code?

7 A. I believe this is part of it.

8 Q. And so your company disclosed specific strings of code
9 that could be used to hack the iPhone in its presentation at
10 the Black Hat conference, right?

11 A. No. They presented code that could be used to hack the
12 previous version of the iPhone; but by the time this was
13 presented, the iPhone had been patched. This would not have
14 worked anymore.

15 Q. You've testified that you cannot guarantee that no
16 malicious hacker could use the methods and steps in this
17 presentation to create some new hack of iPhone, right?

18 A. Right. They'd have to make their own attack.

19 Q. Well, you can't guarantee that they wouldn't use these
20 methods and steps to do that. Isn't that what you testified
21 to, sir?

22 A. They wouldn't use these exact steps, but they might
23 increase their knowledge through looking at this and be able
24 to develop hacks.

25 Q. And your company was the first to hack the iPhone and

1 publish the results, right?

2 A. Again, there were people who knew how to unlock it but
3 not how to run shell code on it.

4 Q. So is it your testimony that your company would not be
5 responsible for every hack of the iPhone that comes after
6 your publication of the details of the hack?

7 A. That's right.

8 Q. And it's your testimony you're not responsible for
9 every hack of the iPhone after this presentation because
10 whether you had done it or not, it all would have happened,
11 correct?

12 A. I'm certain that this would have happened.

13 Q. And that's because the iPhone had vulnerability in it,
14 right?

15 A. That's correct.

16 Q. Now, it's your testimony that the details in this
17 presentation would not allow somebody to create a hack of
18 the iPhone once it was patched, correct?

19 A. That's right.

20 Q. So it's your testimony that once Apple patched the
21 iPhone, the details could not be used for any hacks in the
22 future?

23 A. Yeah.

24 Q. Now, if for any reason Apple chose not to issue a
25 patch, do you believe your company would be responsible for

1 all future hacks of the iPhone?

2 A. No. We were gonna issue the patch, but I think it
3 would have been very irresponsible of them not to issue a
4 patch for this.

5 Q. How was Apple able to issue a patch for the iPhone?

6 A. They have a mechanism: When you synchronize your
7 iPhone with your computer, it will download any patches that
8 have occurred to the phone.

9 Q. You're aware that patch codes for the EchoStar access
10 cards can be delivered through the satellite signal,
11 correct?

12 A. I am aware of that.

13 Q. And you agree that the attack that was posted in the
14 Nipper posting in December of 2000 would not be possible
15 unless the buffer in the ROM 3 card could be overflowed in
16 the first instance, right?

17 A. If the buffer could not have been overflowed, then that
18 particular attack would not have worked.

19 THE COURT: Tell us when is a good time for a
20 break. Now?

21 MR. STONE: Yes.

22 THE COURT: You're admonished not to discuss this
23 matter among yourselves nor to form or express any opinion.

24 Take a recess. We'll come and get you in about
25 20 minutes.

1 (Recess held at 3:00 o'clock p.m.)

2 (Proceedings resumed at 3:18 p.m.)

3 (In the presence of the jury.)

4 THE COURT: We're back in session. All parties
5 are present. The jury is present.

6 This is still cross-examination by Mr. Stone on
7 behalf of NDS.

8 MR. STONE: Thank you, Your Honor.

9 CROSS-EXAMINATION (Resumed)

10 BY MR. STONE:

11 Q. So, Dr. Rubin, backing up for a second, it's clear that
12 the Nipper posting in December of 2000 would not be possible
13 unless the buffer could be overflowed in the first instance,
14 right?

15 A. That's right. The Nipper posting required the ability
16 to overflow the buffer.

17 Q. And the black box pirate device that EchoStar and
18 NagraStar acquired also used a buffer overflow attack very
19 similar to the Nipper attack, right?

20 A. That's right.

21 Q. And the black box device would not work unless the
22 buffer could be overflowed in the first instance, as well,
23 correct?

24 A. Is that what you just asked me?

25 Q. I'm talking now about the black box.

1 A. Right.

2 Q. The black box device would only work if the buffer
3 could be overflowed in the first instance?

4 A. That's right.

5 Q. And you testified that buffer overflows are very easy
6 to fix and easy to prevent, correct?

7 A. In general they are, yes.

8 Q. And you testified in this case that it would take two
9 assembly language programming statements to protect one
10 communications buffer against an overflow, correct?

11 A. That's right.

12 Q. And the ROM 3 card had one communications buffer,
13 correct?

14 A. Correct.

15 Q. And you were asked at your deposition, don't you
16 recall, that if you were asked to write code to prevent
17 buffer overflow for the ROM 3 card, how would you write that
18 code? Do you recall that?

19 A. I remember that, yes.

20 Q. And didn't you say you would write code so that when
21 the buffer was being read in, you would simply count the
22 bytes, and when it got to the length of the buffer, it would
23 stop reading?

24 A. Right. That's what I would do.

25 Q. And it was your testimony that that code that you just

1 described would effectively prevent anyone from using a
2 buffer overflow attack; is that correct?

3 A. From using that particular buffer overflow attack, yes.

4 Q. Well, from overflowing that buffer that's protected by
5 that code is what you testified to.

6 A. Right.

7 Q. So there could be no attack on that buffer if it had
8 those two lines of code that counted bytes that you just
9 described, correct?

10 A. That's right.

11 Q. And you testified, didn't you, that you have no reason
12 to disagree with Mr. Jones' opinion that the patch code used
13 on the ROM 3 card effectively completely precluded the
14 buffer overflow vulnerability from being used?

15 A. That's right.

16 Q. You also testified you had no reason to dispute
17 Mr. Jones' testimony that the patch code for the ROM 3 card
18 actually checked twice for buffer overflow and precluded it?

19 A. That's right. Although after listening to
20 Christophe Nicolas today, I think it's much more complicated
21 than that.

22 Q. Well, you had plenty of time to study this issue from
23 when you were retained in July of 2007 and March 26, 2008,
24 when you saw the source code, didn't you?

25 A. Right. Except that --

1 Q. Didn't you?

2 A. Well, that wasn't what I was tasked to do.

3 Q. Nobody asked you to look at that; is that right?

4 A. No.

5 Q. And when you came out and looked at the source code,
6 did you even go and look at the patch code?

7 A. No, I did not.

8 Q. Did somebody tell you not to?

9 A. No.

10 Q. Now, you've testified that the Nipper posting would
11 only be possible if the buffer could be overflowed in the
12 first instance, right?

13 A. Right.

14 Q. So if the ROM 3 cards had been electronically patched
15 before that posting, then the Nipper posting could not have
16 been used to dump the contents of the ROM 3 card, correct?

17 A. It depends what the electronic patch did.

18 Q. If it prevented buffer overflow?

19 A. Then the cards that received that patch would not be
20 vulnerable to that attack.

21 Q. And that would be true also of the black box attack as
22 well, correct?

23 A. What would be true -- are you asking me if the black
24 box would not be able to hack cards that had that patch on
25 it already?

1 Q. Yes, sir.

2 A. That's correct.

3 Q. Is it -- if your company had been advising Kudelski,
4 you would have recommended immediate development of a patch
5 upon discovery of the buffer overflow vulnerability; isn't
6 that correct?

7 A. That's correct.

8 Q. Now, you saw evidence that Kudelski and Nagra were
9 aware of the buffer overflow vulnerability prior to
10 October 2000, correct?

11 A. Not the vulnerability. I saw they were -- I saw
12 evidence that they knew they were not checking the buffer
13 bounds, but not that they considered that it was a
14 vulnerability.

15 Q. Well, the black box pirate device -- you used a buffer
16 overflow vulnerability, didn't you?

17 A. Right.

18 Q. And Kudelski had that no later than October 2000,
19 right?

20 A. Right. That's what I heard.

21 Q. You don't know of any reason, any technological reason,
22 why a patch could not have been issued to the ROM 3 cards at
23 that time, correct?

24 A. That's not correct.

25 Q. Well, let me -- I'll show you your deposition.

1 A. Let me say this was based on yesterday's testimony
2 where I heard information I was not previously familiar
3 with.

4 Q. Well, as of March 27th, 2008, your sworn testimony was
5 you weren't aware of any technological reason that a patch
6 could not have been issued at the time the back box was
7 analyzed, correct?

8 A. I was not aware of it then. And I am now.

9 Q. Well, had you bothered to talk to anyone at the client
10 prior to testifying twice in this case under oath?

11 A. No, I had all the information that I needed to form my
12 opinions in this case about one piece of code deriving from
13 the other, and that is outside of that scope.

14 Q. When was the first time you ever spoke to anyone at the
15 client?

16 A. I think it was the Saturday before -- no -- the Monday
17 before the trial started.

18 Q. Did anyone tell you that your testimony was incorrect
19 and that you should have changed it?

20 A. No. I realized when I was listening yesterday that I
21 had not understood everything about the constraints that the
22 site designers were operating under.

23 Q. It is your testimony yesterday that you did not see any
24 evidence from your review of the evidence explaining why
25 Nagra or Kudelski did not issue a patch after the black box

1 was obtained and before the December 2000 posting, correct?

2 A. Right. I didn't understand at that point the resource
3 constraints that they were operating under. And so I, you
4 know, without that information that I now have, I was --
5 that was my opinion then.

6 Q. Didn't you say that your company, had it been advising
7 Kudelski, would have recommended fixing the buffer overflow
8 vulnerability before the December posting and after the
9 acquisition of the black box?

10 A. But the December posting and after -- the black box was
11 acquired before the posting, right?

12 Q. Right?

13 A. I don't know what you mean before the December posting
14 and after the back box acquisition.

15 Q. You had advised Kudelski to fix buffer overflow
16 vulnerability as soon as they acquired the black box,
17 correct?

18 A. I would have advised that, but they would have
19 explained to me why that's a lot harder than I realize, and
20 I would have understood that explanation.

21 Q. Now, the resource constraints you testified to applied
22 only to the ROM 2 card, correct?

23 A. No.

24 Q. Was there room in the EEPROM for patch code to prevent
25 a buffer overflow on the ROM 3 card?

1 A. I don't know.

2 Q. Now, a patch was issued in this case after the
3 December 2000, posting?

4 A. Right.

5 Q. And you're saying that could not have been in
6 October 2000.

7 A. I haven't said, but I'm saying.

8 Q. My question is, is it your testimony that patch could
9 not have been issued in October of 2000?

10 A. No, I didn't say that.

11 Q. Well, isn't it true that if a patch had been issued
12 that prevented buffer overflowing before the Internet
13 postings, the memory aliasing feature in the ROM 3 card
14 could not have been exploited?

15 A. Not with that attack. I don't believe that it's true
16 that it could never have been exploited.

17 Q. How else do you exploit memory aliasing unless you
18 overflow the buffer?

19 A. You -- the memory aliasing would happen anytime an
20 address was accessed that was above the legal terminating
21 address for the memory. And that might have happened any
22 number of ways. This particular attack required overflowing
23 the buffer first, and so that attack wouldn't have been
24 possible.

25 Q. Now, isn't it true that if the patch to prevent buffer

1 overflow had been deployed prior to the postings, the ROM 3
2 card would have been a more secure and better product?

3 A. Yes.

4 Q. And wasn't that the reason your company hacked the
5 iPhone, to make it a more secure and better product, because
6 it forced Apple to patch the buffer overflow vulnerability?

7 A. It's one of the reasons. We already discussed the
8 publicity and, you know, educating security people, but that
9 was one of the reasons.

10 Q. Isn't it your claim that your company hacks products
11 and finds vulnerabilities to serve the greater goal of
12 having vendors fix the vulnerabilities as soon as the
13 vulnerabilities are disclosed to them?

14 A. I would say that that's too strong a statement. I
15 think it's -- we do it so that they can fix the
16 vulnerabilities as soon as they can.

17 MR. STONE: Your Honor, I would like to play from
18 Mr. Rubin's deposition at Page 172, Lines 12 through 18.

19 THE COURT: Beginning question, "Well"?

20 MR. STONE: Yes, sir, at Line 12 to 18.

21 THE COURT: That's denied, Counsel.

22 MR. STONE: Thank you, Your Honor.

23 BY MR. STONE:

24 Q. Mr. Rubin, did you testify that when you hacked the
25 iPhone, it was to serve the greater goal of having vendors

1 fix the vulnerabilities when the vulnerabilities -- didn't
2 you testify that it was for the reasons of greater
3 disclosure and fixing of vulnerabilities that the xbr21
4 disclosure might have led to a better product?

5 A. Can you say that again, please.

6 Q. Sure. Didn't you testify that having vendors fix
7 vulnerabilities as soon as they are disclosed was a reason
8 why the xbr21 disclosure might have led to a better product?

9 A. I'm assuming I said that in my deposition.

10 Q. Do you believe Nagra Kudelski as the vendor of the
11 ROM 3 card had a duty and responsibility to patch the buffer
12 overflow vulnerability as soon as they became aware it was
13 being used for pirate attacks?

14 A. I think they should have done it as soon as they could.
15 And there are some constraints on the card that would make
16 that really difficult to do very, very quickly.

17 Q. Do you know when they first began work on designing and
18 developing the patch?

19 A. I believe that Mr. Nicolas said that they did that
20 after they analyzed the black box and met with ST Thomson
21 and figured out what the problem was.

22 Q. Now, earlier you testified that the Headend Report, or
23 the Haifa Report, as you call it, and the Nipper posting
24 have the same DNA?

25 A. That's right.

1 Q. But you previously testified that the comparison you
2 made showed that the structure of the attack was similar,
3 correct?

4 A. The structure, yes.

5 Q. That's what you testified to earlier today, correct?

6 A. Right.

7 Q. Now, you and I have the same physical structure, don't
8 we?

9 A. Yes.

10 Q. I have two arms; you have two arms?

11 A. Yes.

12 Q. I have two legs --

13 THE COURT: Counsel, counsel...

14 BY MR. STONE:

15 Q. You and I don't share the same DNA, correct?

16 A. That's right.

17 Q. And did you compare the DNA between the black box and
18 the Nipper posting?

19 A. I did.

20 Q. And was it your opinion that the Nipper and the black
21 box shared the same DNA?

22 A. Yes.

23 Q. And if I could show you your report, Exhibit 799, at
24 Page 32 of 38. It's -- 799-32 is the bottom number?

25 A. I'm there.

1 Q. Okay. Now, in your report, you agreed with Mr. Jones
2 that as between the Nipper posts and the Headend Report, the
3 programming methodologies and styles are different, correct?

4 A. That's right.

5 Q. The programs are different lengths, correct?

6 A. That's right.

7 Q. The hexadecimal byte sequences are different, correct?

8 A. That's right. They would have to be if they were
9 different programs.

10 Q. Different addressing methodologies are used, correct?

11 A. Where does it say that?

12 Q. Right after the byte sequences are different.

13 A. How far down the page?

14 Q. About the fourth line.

15 A. Right.

16 Q. And then you agreed that many other specific aspects of
17 the two programs are different, correct?

18 A. That's right.

19 Q. Now, when you say "the programming methodologies are
20 different," what is a programming methodology?

21 A. So in this particular example, the Haifa Report, you
22 know, there are a lot of ways -- a methodology is just how
23 somebody goes about writing code. In this case, one example
24 I could use would be the using -- developing your own code
25 for certain functions that built-in functions exist for and

1 then otherwise using a built-in function.

2 Q. You also said the programming styles were different,
3 correct?

4 A. That's right.

5 Q. And in what way were the programming styles different?

6 A. It's kind of a hard thing to describe somebody's
7 programming style. It's like asking someone what their
8 writing style is like and how it's different. But I will
9 say that the Nipper code is much more compact, more cleverly
10 crafted.

11 Q. And then you also agreed, right, hexadecimal byte
12 sequences are different?

13 A. Right.

14 Q. What did you mean by that?

15 A. That it's not a carbon copy of the other one.

16 Q. And did you ever measure the degree to which there was
17 a difference in the hexadecimal byte sequences?

18 A. No, I did not. I don't see that that would be relevant
19 at all.

20 Q. You said that different addressing methodologies are
21 used?

22 A. Right.

23 Q. And is it true that Mr. Mordinson wrote his own, while
24 the Nipper posting uses a library call function?

25 A. Right. That's what I was calling a built-in function.

1 Q. You said many other aspects of the two programs were
2 different, correct?

3 A. That's right.

4 Q. Would that include the Headend Report as the shell code
5 in the communications buffer while Nipper places it in the
6 stack?

7 A. Exactly.

8 Q. And can you place the same number of bytes in the
9 communications buffer as you can in the stack in the ROM 3
10 card?

11 A. No.

12 Q. What is the difference?

13 A. So in the communications buffer, you can only put a
14 hundred bytes. And I don't recall the size of the stack.
15 But it's -- I remember it's not a hundred.

16 Q. Now, the Nipper posting uses a routine built into the
17 card to transmit data out of the card, correct?

18 A. That's right.

19 Q. And Mordinson wrote his own program, correct?

20 A. That's right.

21 Q. And the Nipper posting terminates by jumping into the
22 main processing loop, correct?

23 A. That's right.

24 Q. And how does Mordinson's code end?

25 A. It goes into an infinite loop.

1 Q. And didn't you testify that the Headend Report would
2 not have been sufficient to create the xbr21 posting?

3 A. There were -- what I said was that there were other
4 aspects about Nipper that didn't seem to be known or weren't
5 utilized by Mordinson in his code.

6 Q. Didn't you testify that the Headend Report would not
7 have been sufficient to create the xbr21 posting?

8 A. I don't think I said that.

9 MR. STONE: Your Honor, I would like to read
10 Page 21, Lines 7 through 12.

11 THE COURT: Overruled. You may not.

12 BY MR. STONE:

13 Q. Is it correct, Dr. Rubin, that there was programming
14 code in the xbr21 post that was not in the Headend Report
15 that NDS did not know was necessary for the attack to be
16 functional?

17 A. I'm not sure that NDS didn't know it. I don't know
18 that they put everything that they knew into their report,
19 but there was code in the Nipper posting that was not in the
20 Haifa -- in the Headend Report.

21 Q. Well, if somebody had used the code in the Headend
22 Report at the end of the program, it would simply hang,
23 correct?

24 A. That's right.

25 MR. STONE: And now if we could look at

1 Exhibit 511A, the Nipper posting?

2 This is in evidence.

3 (Document displayed.)

4 BY MR. STONE:

5 Q. Now, if you go to the second page of 511A, sir, in the
6 third line from the end of the code, there's --

7 A. Are you counting the incorrect checksum as a line?

8 Q. Yes.

9 A. So the first complete line from the bottom?

10 Q. Where it says "0x73, 0x81."

11 A. Right.

12 Q. Is it -- that's the so-called "73, 81 jump" identified
13 by Mr. Jones?

14 A. That's correct.

15 Q. And that's an address in memory that you need to jump
16 to, to continue to have the program function, correct?

17 A. That's right.

18 Q. And as part of that jump, the xbr21 code also passes
19 parameters, correct?

20 A. Right.

21 Q. And could you agree that the 73, 81 jump is nowhere
22 contained in the Headend Report?

23 A. That's correct.

24 Q. And you saw no evidence that anyone at NDS had figured
25 out the 73, 81 jump that is contained in the posting?

1 A. I saw no evidence of that.

2 Q. And the 73, 81 jump is nowhere contained in the Headend
3 Report, correct?

4 A. Yeah. I thought that's what you just asked.

5 Q. And so NDS' Haifa lab had the ROM code, but the Haifa
6 lab never derived this 73, 81 jump for the passing of the
7 parameters that are in the xbr21 posting?

8 A. I don't know. They didn't put it in the report. That
9 doesn't mean they didn't derive it.

10 Q. Did you see any evidence in any of the documents or
11 testimony that they had derived that?

12 A. No.

13 Q. And it was your hypothesis that xbr21 was able to
14 derive the 73, 81 jump and the passing of parameters by
15 having access to a dump of the ROM code, correct?

16 A. That would be one way to get it.

17 Q. And it was your testimony that it would have been
18 difficult for somebody to figure out the parameters of the
19 73, 81 jump, correct?

20 A. Yes.

21 Q. And I believe you testified you saw no evidence that
22 NDS figured out that jump, and you don't know how xbr21 was
23 able to derive that jump as part of the posting on the
24 Internet, correct?

25 A. That's right.

1 Q. So if I understand it correctly, in addition to these
2 differences in the code that we have talked about, the xbr21
3 posting had code that NDS did not know about, you saw no
4 evidence that they had figured out, and was necessary for a
5 functional buffer overflow attack, correct?

6 A. No. The 73, 81 is not necessary for a functional
7 buffer overflow attack. If the card goes into an infinite
8 loop at the end, as it does in the Haifa Report, you still
9 dump the EEPROM and have the ability to hack the card. This
10 is simply a way that's a little cleaner, which is why it's
11 my opinion that this is an improvement over the Haifa after
12 more information was learned.

13 Q. Information that you don't know how xbr21 derived,
14 correct?

15 A. No. Access to the ROM code would have been one way to
16 do that.

17 Q. Now, let's take up the similarities that you talked
18 about.

19 First, the buffer overflow. That's a vulnerability
20 that's in every ROM 3 card, correct?

21 A. That's right.

22 Q. And that would be every unpatched.

23 A. Yes, sir.

24 Q. And that's the weakest link in the card's security.

25 A. I don't know about that. I think taken together, all

1 of the factors are a weak link.

2 Q. And the only way you can exploit those other factors is
3 to have buffer overflow in the first instance, correct?

4 A. Every single one of them is necessary for the attack to
5 work, so I don't see that one of them is more important than
6 the others.

7 Q. Can you exploit the memory alias, then, if you cannot
8 overflow the buffer?

9 A. No.

10 Q. And the buffer overflow vulnerability can be determined
11 by sending invalid inputs through the process of fuzzing,
12 correct?

13 A. Right.

14 Q. And buffer overflow vulnerabilities are well-known and
15 common, and it's one of the first things an attacker would
16 think of, correct?

17 A. That's right.

18 Q. Anyone who had access to the ROM and disassembled it
19 would see that Nagra had deliberately failed to check for
20 overflow in the communications buffer in the ROM 3 card,
21 correct?

22 A. I disagree with that.

23 Q. Let's talk about memory aliasing. That's a property of
24 the particular ST microchip that was used in the ROM 3 card,
25 correct?

1 A. That's right.

2 Q. And did that exist in the ROM 2 card as well?

3 A. I don't know.

4 Q. How do you perform a buffer overflow attack on the
5 ROM 3 card without using memory aliasing?

6 A. You just send it more than a hundred bytes.

7 Q. And didn't you testify that you did not know whether
8 there was publicly available documents for the family of
9 chips using the EchoStar access cards that references memory
10 aliasing?

11 A. Yes, I don't know what all the documentation is for it.

12 Q. And you're aware that Mr. Jones looked at the data
13 sheet for this family of chips, correct?

14 A. That's right. I looked at it as well after my first
15 deposition.

16 Q. And you didn't dispute Mr. Jones' opinion that the data
17 sheet indicated there was memory aliasing?

18 A. I didn't see memory aliasing in the data sheet.

19 Q. You didn't dispute Mr. Jones' opinion that it discloses
20 memory aliasing, did you?

21 A. No. Before I looked at the time data sheet, you had
22 asked me if I had any reason to dispute Mr. Jones' statement
23 that there was aliasing. And at that point I didn't have
24 any reason to dispute it, but I do now.

25 Q. So if I understand correctly, are you saying that if

1 your company used fuzzing to overflow the ROM 3 buffer, the
2 bytes would not memory alias?

3 A. Well -- what?

4 Q. If your company were to use fuzzing to overflow the
5 buffer --

6 A. Right.

7 Q. -- would those bytes memory alias?

8 A. Sure.

9 Q. Why would that be?

10 A. Because that's what the card does when you send it too
11 many inputs.

12 Q. And that's because that is a feature of the card
13 itself?

14 A. It's a property of the card.

15 Q. Property of the card?

16 A. Yes.

17 Q. So anyone overflowing the buffer would end up utilizing
18 memory aliasing, correct?

19 A. That's right. I don't know about utilizing it, but
20 they would cause it.

21 Q. And they would become aware of it, correct?

22 A. No.

23 Q. Well, didn't you testify that you can test very simply
24 for memory aliasing?

25 A. If you're looking for it. You wouldn't probably

1 discover it on your own if you were just fuzzing it.

2 Q. Well, what would the test be to determine memory
3 aliasing that you said would be easy to devise?

4 A. This test assumes that you know about memory aliasing
5 and that that's what you're looking to verify whether or not
6 it exists. So the test is that you would send it
7 increasing-sized messages and you would have some way of
8 determining that you were overriding memory in the lower
9 regions of the card. They're many ways that you can do
10 that, but you're not likely to discover memory aliasing that
11 way unless you know that you're looking for it.

12 Q. Well, if you gave it escalating inputs and you saw
13 whether you got an error message or if you started to see
14 the same values again, that would tell you whether memory
15 aliasing was occurring, wouldn't it?

16 A. Again, if you started out with the problem of asking
17 yourself, "Does this card do memory aliasing," then, yes,
18 that would confirm or not for you that that happened. But
19 just because you saw that wouldn't necessarily lead you to
20 the conclusion that there was memory aliasing.

21 Q. One of the other elements you identified was the
22 incorrect checksum creating an exception. Do you recall
23 that?

24 A. That's right.

25 Q. And at your deposition you testified you could not

1 identify any other way to create an exception in the ROM 3
2 card, correct?

3 A. That's right.

4 Q. And didn't you testify also there was only one way you
5 could think of to get shell code to a place where it could
6 execute on the ROM 3 card?

7 A. Yeah. But I also since changed that opinion with new
8 information about the card.

9 Q. After March 27th, 2008?

10 A. Right.

11 Q. Now, the use of the index variable -- can you envision
12 a buffer overflow attack that does not use or make use of
13 the index variable?

14 A. Yes.

15 Q. And what would that example be?

16 A. Any buffer overflow attack that isn't crafted the way
17 these three that we've looked at are crafted, you would
18 overflow the buffer but not put in so many bytes that you
19 change the index variable. So say put 115 bytes in and
20 you've overflown the buffer, but you haven't touched the
21 index variable.

22 Q. And would that give you the proper return address for
23 the shell code?

24 A. No. You were asking me of an attack that didn't change
25 the address, but you didn't say that I was trying to do what

1 this attack was trying to do.

2 Q. Now, when hacking the iPhone, Mr. Miller of your
3 company took advantage of certain vulnerabilities in the
4 iPhone, right?

5 A. That's right.

6 Q. One of those is the fact that the iPhone uses a heap or
7 stack that allows the execution of code, correct?

8 A. That's right.

9 Q. Another was the fact that the iPhone was subject to a
10 buffer overflow, correct?

11 A. Correct.

12 Q. He also used shell code in conjunction with the buffer
13 overflow to get the iPhone to do what he wanted, correct?

14 A. Right. And the things that you're describing right now
15 are basically the -- the basic components of any buffer
16 overflow attack commercially in the field. Microsoft pretty
17 much once a week releases patches to their operating systems
18 and applications that are designed to address exactly these
19 kind of issues. So these are very common things. They're
20 not specific to the iPhone.

21 Q. Mr. Miller also found out that there was no
22 randomization of memory in the iPhone, correct?

23 A. That's correct.

24 Q. And then he used filler bytes to overflow the stack in
25 the iPhone, correct?

1 A. That's right.

2 Q. Did I miss anything that he utilized in the attack?

3 A. Let me think. I would guess you did, but I don't --
4 I'm not familiar with the attack at the level to say what
5 they are.

6 Q. So if a subsequent attack on the iPhone exploited those
7 same vulnerabilities, would you conclude that Mr. Miller
8 must have been the source of that attack?

9 A. I don't think that would give me enough information to
10 come to that conclusion. I would look at their attack and I
11 would look at his attack the way I looked at Haifa and
12 Nipper and saw the same DNA and same properties, and these
13 were all things not known. Whereas, all the things in the
14 iPhone were pretty much known, then I would conclude that.

15 Q. And Mr. Miller was giving a presentation because all of
16 that was already known?

17 A. The particular things that you described, that the
18 stack was executable, that there was a buffer overflow
19 vulnerability -- take that one back -- I'm not sure if it
20 was known, but anyone that would have fuzzed on the iPhone
21 would have known that there was a buffer overflow
22 vulnerability. And the other things you mentioned were
23 known.

24 Q. The four items or key components you identified
25 earlier -- the buffer overflow, the memory aliasing, use of

1 the index variable, and the exception from an incorrect
2 checksum -- you testified all could be deduced by an
3 appropriately skilled individual who had access to the ROM
4 code, correct?

5 A. What I said was that an individual at least as skilled
6 as David Mordinson, spending six months and under the
7 guidance of someone like Zvi Shkedy who had broken up the
8 ROM, with the ability to ask that person questions could, in
9 fact, given the ROM code, deduce those vulnerabilities. In
10 fact, he did it.

11 Q. And you also testified that there were people out there
12 equally capable as Mr. Mordinson?

13 A. That's right.

14 Q. Now, you disagreed with Mr. Jones' opinion that the
15 buffer overflow vulnerability was either intentional or due
16 to incompetence, correct?

17 A. Before I had looked at the code, I disagreed that that
18 buffer wasn't checked intentionally. And now that I've
19 looked at the code, I believe that they knew that they
20 weren't checking the bounds and that they had good reason
21 not to check the bounds.

22 Q. And didn't you testify that whether there is memory
23 aliasing or not, one should always check for buffer
24 overflow?

25 A. Yes, that's correct.

1 Q. And in reviewing the code, did you find that there were
2 two other buffers in the program for the ROM 3 code?

3 A. I did.

4 Q. Were those checked to make sure they did not overflow?

5 A. Yes.

6 Q. Could those two buffers be used for a malicious pirate
7 attack?

8 A. I don't believe so.

9 Q. The only buffer that can be used for a pirate attack is
10 the communications buffer, correct?

11 A. That's right.

12 Q. And at the time you testified originally, you had not
13 seen the source code?

14 A. That's right.

15 Q. And your testimony originally was that it would be
16 unbelievable that somebody would deliberately not check for
17 overflow, correct?

18 A. That's right.

19 Q. And did anyone at the client tell you before you gave
20 that opinion that it was not true that they had deliberately
21 failed to check?

22 A. As I mentioned earlier, I had not talked to anyone at
23 the client.

24 Q. Did you disclose your opinions to the attorneys for the
25 client before you testified?

1 A. No. That was a bit of an unusual question, so I think
2 the attorneys became aware of it in my deposition when you
3 asked it.

4 Q. And without having seen the code, your original
5 testimony was that from looking at all the evidence, the
6 conclusion you drew was they simply were not thinking about
7 buffer overflow attacks, correct?

8 A. Right. Before looking at the code, I was under the
9 impression that someone that wouldn't check the bounds on an
10 I/O buffer was just not thinking about it.

11 Q. And Mr. Jones, before seeing the code, had concluded
12 from the design decisions he had seen that it was more
13 likely than not that it was a deliberate decision, correct?

14 A. He did. I totally disagree with that, but that's what
15 he said.

16 Q. And it turned out he was correct, at least with respect
17 to it being a deliberate decision, correct?

18 A. That's right.

19 Q. Now, did you do anything to examine, inspect, analyze,
20 or verify the reason given in the code for the deliberate
21 failure to check for overflow?

22 A. Yes, I did.

23 Q. And didn't you testify that if we assume that the same
24 author wrote the code that explicitly checked with the two
25 buffers that cannot be used for piracy and deliberately

1 chose not to check for the one buffer that could be used for
2 piracy, your explanation would be that the person maybe
3 wrote one section of the code in a good mood and one section
4 of the code in a bad mood, correct?

5 A. No. I did say that, but not to that. You asked me how
6 could it be that the comments -- I said that the comments in
7 one place were written in a particular style, and in another
8 place they were written in another style, and that was one
9 of my reasons for speculating that perhaps they were written
10 by different people. And you asked me if they weren't
11 written by different people, if they were written by the
12 same person, how would I explain that, and I said that maybe
13 they were in a good mood when they wrote one part of it and
14 a bad mood when they wrote the other part.

15 Q. And the bad mood would be not checking the buffer part
16 of it?

17 A. I said that the comments in one section were kind of
18 dull, very much like a computer programmer who's just not,
19 you know, very much creative or something; and the other
20 place he was saying "hanky-panky" and things like that. And
21 I said, "I think it's a different person."

22 And he asked me, "Well, what if it's" -- you know,
23 "What if it's the same person?"

24 And I said, "Well, they were in a different mood."

25 That's what the conversation was.

1 Q. The hanky-panky comments in the patch code?

2 A. Right.

3 Q. That's not the code wherein there is a deliberate
4 failure to check for the communications buffer, correct?

5 A. Right. No, you're right.

6 Q. And you didn't look at the patch code at the time you
7 gave your deposition, right?

8 A. Right. I was drawing that statement from the testimony
9 yesterday where you heard "hanky-panky," and I was using
10 that as an example.

11 Q. I was focusing on your deposition testimony after you
12 had looked at the source code where you determined the
13 author failed to check for buffer overflow in the
14 communications buffer.

15 A. Fair enough. But my point is still that the commenting
16 was written in a different voice in the two sections.

17 Q. Which was the bad mood? The failure to check?

18 A. I don't remember.

19 Q. Now, one of the explanations given for the failure to
20 check the communications buffer or overflow was the claim
21 that there was no memory after the buffer, correct?

22 A. That's one of them, yes.

23 Q. And if it turned out that the author of the code
24 actually knew about memory aliasing, that false explanation
25 would be indicative of an insider attack, correct?

1 A. No, I think the insider attack theory is fatally
2 flawed, and I've heard -- I've heard that theory before from
3 Mr. Jones. But I have an explanation of why that can't be
4 possible or at least is incredibly unlikely.

5 Q. Did you ever speak to the author of the code, Mr. Osen?

6 A. No, I did not.

7 Q. Do you know whether Mr. Osen was aware of memory
8 aliasing when he wrote the code?

9 A. I believe he was not aware of it.

10 Q. Do you know for a fact that --

11 A. I do not.

12 Q. Going back to your report, Exhibit 799 --

13 A. Yes.

14 Q. -- and focusing you, please, on Page 36 of 37. And I
15 believe it has 799-36 at the bottom.

16 A. Right.

17 Q. And that was signed by you on August 29th, 2007,
18 correct?

19 A. That's right.

20 Q. And you wrote, "This report is based on the evidence
21 provided to me in this case," correct?

22 A. Yes.

23 Q. And that evidence was provided to you by counsel, I
24 take it?

25 A. That's right.

1 Q. And you reviewed those materials and tried to be as
2 accurate and thorough as you could be in preparing your
3 report, right?

4 A. Of course.

5 Q. And you stand by everything that's in your report,
6 correct?

7 A. We discovered in my deposition, and you were nice
8 enough to point out, a couple of minor mistakes.

9 Q. Those were the diagrams that were wrong?

10 A. Right. One of the lines in the Haifa report I had
11 drawn in the wrong place. And then the other was not really
12 a mistake, but you pointed out one of my conclusions didn't
13 have a lot of supporting information in the report, and I
14 agreed with you.

15 Q. Apart from those two instances, do you stand by
16 everything else in the report?

17 A. Yes.

18 Q. If you could go to Page 2 of 38 of your report, please.
19 You have a section that's entitled, "Data and Other
20 Information Considered," correct?

21 A. That's right.

22 Q. And in that paragraph you wrote, "Other significant
23 case files used in the preparation of this report include
24 the plaintiff's Fourth Amended Complaint, the Nipper
25 postings from December 23rd and December 24th, 2000,

1 correct?

2 A. Right.

3 Q. And that was based on information provided to you?

4 A. That's correct.

5 Q. If you could go to Page 18 of 38 of your report.

6 And you wrote, "There are two Internet postings of
7 particular interest. One of them was posted by xbr21 on
8 December 23rd, 2000, signed NipperClause00. I will refer to
9 this posting as Nipper1." Is that correct?

10 A. Yes.

11 Q. And that again was information provided to you by the
12 attorneys representing plaintiffs?

13 A. That's right.

14 Q. And based on all the information you received from the
15 lawyers, you understood one of the key issues in dispute was
16 whether NDS was responsible for the December 23rd posting of
17 the so-called recipe to dump the EEPROM, correct?

18 A. That's right.

19 Q. And in your report you say that "Prior to
20 December 23rd, 2000, those features were undocumented in the
21 DISH Network hacker community," correct?

22 A. Where? I lost you. Where are you reading from?

23 Q. If you look at Page 6 of your report?

24 A. Page 6?

25 Q. Correct, sir.

1 A. Okay. Where on Page 6?

2 Q. Number 1, I believe it is.

3 A. Can you please repeat the question.

4 Q. Sure. In your report there, you say, "Prior to
5 December 23rd, 2000, the buffer overflow vulnerability, the
6 RAM ghost effect, the index variable --

7 A. The location and purpose of the index variable --

8 Q. Right. It's --

9 THE COURT: Just a moment. You two are speaking
10 over the top of each other.

11 THE WITNESS: I would say.

12 THE COURT: Stop.

13 THE WITNESS: Got it.

14 THE COURT: Thank you.

15 Counsel.

16 BY MR. STONE:

17 Q. Looking at that section, it says, "Prior to
18 December 23rd, 2000" -- I won't read it all in, but the
19 components you discuss, correct --

20 A. Right. They're there.

21 Q. -- were undocumented in the DISH Network hacker
22 community, correct?

23 A. Correct.

24 Q. And to this day you have not changed that in your
25 report, correct?

1 A. Right. I haven't changed that opinion.

2 Q. Like you did your diagrams, which you did change,
3 correct?

4 A. I did change the diagram, and we submitted the new one.

5 Q. And nowhere in your report is there any reference to a
6 December 21st, 2000 posting; isn't that correct?

7 A. That's right.

8 Q. And no one ever showed you any December 21st, 2000,
9 posting; is that correct?

10 A. I believe I was told that the December 23rd posting was
11 a reposting of something that was posted on December 21st,
12 but --

13 Q. Did anyone ever show you a December 21st, 2000,
14 posting?

15 A. I don't think so.

16 MR. STONE: Thank you.

17 No further questions at this time.

18 THE COURT: Redirect.

19 MR. HAGAN: Thank you, Your Honor.

20 REDIRECT EXAMINATION

21 BY MR. HAGAN:

22 Q. Dr. Rubin, you reference something in your testimony
23 with Mr. Stone called a resource-constraint device, but he
24 didn't give you an opportunity to explain that to the jury.

25 Can you do that?

1 A. Sure. And yesterday Christophe Nicolas talked quite a
2 bit about this. A Smart Card is a little computer, and it
3 doesn't have as much memory as a computer, a regular
4 computer. It doesn't have as fast a processor, and it
5 doesn't have as big an address space. It's -- we call those
6 things resources, and we would say that this is a
7 resource-constrained device where everything really matters.

8 When the program is executing, the chip goes off of a
9 clock, and every time the clock ticks, the chip can do
10 something in its processing. And every time the clock
11 ticks, we call it a clock cycle.

12 When I was looking at the code, I noticed that there
13 were numbers in parentheses and comments along a lot of the
14 lines of code, in particular as relating to the I/O buffer.
15 I had no way of knowing what they were. When I met
16 Christophe Nicolas, he explained to me what those were.
17 Those were a count of how many clock cycles each instruction
18 required. And I thought to myself, "Well, that's really
19 unusual. When would you ever care -- I mean, clock cycles
20 are milliseconds. They're very, very fast. Why would it
21 matter that this instruction takes three clock cycles and
22 this one takes five?" It's not a thing that you usually
23 would care.

24 And he explained to me the importance in the Smart Card
25 of keeping track of the clock cycles, because when you, for

1 example, press a button on your remote, you want the channel
2 to change right away. And in the I/O buffer, you're in the
3 critical portion where bytes are being copied from the
4 set-top box into the Smart Card, and if you're still
5 processing one of the bytes, the next byte's gonna come in,
6 and you won't be ready for it. And so that entire card was
7 designed under very, very tight constraints. In fact, the
8 comment in the code said that there were only six clock
9 cycles left to spare.

10 Now, that is code that executes in a loop. Every
11 single time a character is read in, that code executes
12 again. So if you were to add some instructions to that, you
13 would be increasing the clock cycles, and you would be
14 increasing it times the number of times that you would loop
15 around the loop.

16 And so I have no doubt that once they were aware of the
17 problem, they wanted to patch it. But it wasn't a simple
18 matter of just adding a couple of lines of code, because
19 that could have really messed up the card, and they were
20 very sensitive to not doing that.

21 And now I understand from Christophe Nicolas that when
22 they finally did issue a patch, they completely restructured
23 the way that they did the input and output, and that's where
24 they ended up with this thing of checking it twice. It had
25 to do with the new structure that they had completely

1 rewritten, and that's why it took them a while before they
2 could release the patch.

3 Q. Thank you, Dr. Rubin.

4 A. No problem.

5 Q. So is it your testimony based on what Mr. Nicolas
6 testified to yesterday that the plaintiffs acted reasonably
7 to develop and launch that software patch?

8 A. I think they did.

9 Q. Now, Mr. Stone also referenced to you an opinion --
10 that I think their expert may have -- that somehow this was
11 an inside job, that EchoStar and NagraStar and NagraCard
12 hacked their own system, cost themselves their own millions
13 of dollars. Do you agree with that opinion?

14 A. No, I don't. I don't understand how you could think
15 that.

16 Q. And can you explain to the ladies and gentlemen of the
17 jury why?

18 A. Sure. If I'm setting out to build a card, and my goal
19 is to later be able to hack the card for some reason -- I
20 don't know if you're familiar with the term Rube Goldberg.
21 It's a term for a crazy -- some of you are nodding, but I'm
22 going to explain for those who aren't -- a crazy, wild
23 contraption.

24 Let's say your goal is to close a hot dog. You put the
25 bun on top, and you build something where marbles fall, and

1 it hits a pad and launches a spring, and water shoots down.
2 The whole convoluted thing is called a Rube Goldberg, and I
3 view this hack as a Rube Goldberg hack. It's complicated.
4 It's got a lot of moving parts.

5 If I was going to build a Smart Card that I was going
6 to hack later in this particular system, it would be
7 trivial, because there's something called the "entitlement
8 management message." This is something that's sent to the
9 Smart Card and can include code that can run on the Smart
10 Card when it gets there. And the only thing you need to do
11 to get that to run is know the secret key. If I'm the
12 developer of the card, I made the secret key, I know what it
13 is. All I would do is hang onto that key and let the card
14 go into the field, and I would hack it whenever I wanted to.
15 I wouldn't build something that required a buffer overflow
16 into a ghost alias memory effect into an address index
17 variable into a checksum. I mean, that to me is completely
18 ridiculous.

19 Q. And Mr. Stone, I think, tried to make the point that
20 the engineers for NagraCard intentionally did not check the
21 communications I/O buffer in the ROM 3 card. Is that
22 correct?

23 A. That's correct.

24 Q. And Dr. Rubin, it's true, based on Mr. Nicolas'
25 testimony and your review of the source code, that they did

1 not check that buffer, correct?

2 A. That's right.

3 Q. And can you explain to the ladies and gentlemen of jury
4 what you believe the explanation for that is?

5 A. So that's what I did a few minutes ago. Oh, I'm sorry.
6 It's not. It's a new question. I'm getting confused.

7 So the reason from -- based on the comments and on the
8 testimony that I heard yesterday that I believe that they
9 didn't check the I/O buffer overflow is that they didn't
10 know about the ghost aliasing effect that the memory would
11 wrap around. And they were very, very concerned about clock
12 cycles. And it was their belief that the code -- anything
13 that would be written past the buffer would just dissolve
14 into nowhere because there is no memory over there. And so
15 to save time and to avoid using up these precious clock
16 cycles, they decided not to check the buffer bounds there.

17 Q. In essence it would have been, at least in their mind,
18 superfluous?

19 A. It would have been, in their minds.

20 Q. And while I understand that the defendants and their
21 counsel now disagree with that opinion, David Mordinson does
22 agree with your opinion. In fact, he has that same opinion
23 in the Headend Report itself, doesn't he?

24 A. That's right.

25 Q. And you quote from that in your report where

1 Mr. Mordinson says, "The application designers did not check
2 the maximal possible length of an incoming message while it
3 is being collected as to determine if it exceeds the buffer.
4 Probably they believed such a verification was superfluous.
5 Indeed, there is no physical memory allocated from the
6 location Ox200 to Ox1FFF.

7 "However, due to the RAM ghost effect, an incoming
8 message of maximal length, i.e, 255 bytes, will affect RAM
9 locations from 0X19C to 0X1DD and from 0X20 to 0X9A."

10 Do you agree with Mr. Mordinson's analysis back when he
11 wrote the Haifa report?

12 A. That's right. I think that he conjectured as I'm
13 saying now that they just didn't think about any possible
14 damage that could happen from writing into areas of memory
15 that they didn't think existed.

16 Q. Now, Mr. Stone also showed you Exhibit 809. Do you
17 have that in front of you still?

18 A. Yes.

19 Q. And the date on 809 is May of 1999. That's when that
20 was released, that article, correct?

21 A. Right.

22 Q. And that was coauthored by a former pirate named
23 Oliver Kommerling, correct?

24 A. That's right.

25 Q. And you understood in the testimony of Mordinson and

1 Shkedy that Oliver Kommerling was employed in some respect
2 with the defendants at the time that they created the hack
3 for EchoStar's system, correct?

4 A. I honestly don't remember that. Sorry.

5 Q. Not a problem.

6 You would agree, Dr. Rubin, that you haven't seen any
7 evidence in this case, either from Mr. Stone at deposition
8 or from the documents that you reviewed, that suggest in any
9 way that EchoStar's security system was compromised or
10 hacked at any point in time prior to Mr. Mordinson and
11 Mr. Shkedy developing a hack for that system?

12 A. That's correct.

13 Q. Now, you were also present yesterday when Mr. Nicolas
14 testified that if Chris Tarnovsky e-mailed portions of the
15 ST system ROM to Jan Saggiori in March of 1999, the only way
16 he could have gotten that portion of EchoStar's code was if
17 he either participated in the defendant's Headend Report and
18 project or he got that information from them.

19 Do you have any reason to disagree with Mr. Nicolas'
20 testimony on that point?

21 A. No.

22 Q. Now, it's not disputed, and you heard Mr. Nicolas
23 testify yesterday, that NagraCard developed a software patch
24 which they hoped would fix this hack methodology in early
25 2001, correct?

1 A. That's right.

2 Q. Now, would that software patch have been effective if
3 the pirates were using a device called a blocker?

4 A. No.

5 Q. And can you explain to the jury why that is?

6 A. A blocker is something that is specifically designed by
7 pirates to prevent a patch that's going to fix a
8 vulnerability from actually fixing the vulnerability. It
9 blocks it. That's why it's called a blocker.

10 Q. And would that software patch be effective on pirated
11 ROM 3 cards that were reprogrammed using the Nipper or Haifa
12 recipe prior to early 2001?

13 A. It depends.

14 This -- if -- the pirates could have built a mechanism
15 into cards that were pirated already that would prevent
16 future hacks from being effective.

17 Q. And Mr. Nicolas testified yesterday that that was the
18 case.

19 Do you have any reason to disagree with that testimony?

20 A. No.

21 (Live reporter switch.)

22 (Further proceedings reported by Jane Rule in
23 Volume IV.)

24 -oOo-

25

1 -oOo-

2
3 CERTIFICATE

4
5 I hereby certify that pursuant to Section 753,
6 Title 28, United States Code, the foregoing is a true and
7 correct transcript of the stenographically reported
8 proceedings held in the above-entitled matter and that the
9 transcript page format is in conformance with the
10 regulations of the Judicial Conference of the United States.

11
12 Date: April 17, 2008

13
14
15 _____
16 DEBBIE GALE, U.S. COURT REPORTER

17 CSR NO. 9472, RPR
18
19
20
21
22
23
24
25

A				
ability 41:17 77:13 84:22 92:15 109:9 117:8	11:3 30:2 36:3 36:16 39:21,21 40:9,10 41:20 85:10 86:14,14 86:18 87:2,5,7 88:11,13,15 99:20,21 107:15 114:22,25 115:18 127:5 130:16	113:20 116:25 117:23 121:24 122:8 131:10	71:7,11,16,22 74:8 76:1,12,15 76:22 77:5,17 77:21 78:5,8 81:2,7,9 83:3 84:11 87:19 90:20,24 91:5 100:6	asking 65:16 95:23 104:7 113:16 114:24 aspects 103:16 105:1 106:4 assembly 42:22 51:25 52:3,6,8,9 52:16,17,18,25 53:2,4,5 93:9
able 12:9 14:17 14:18 15:22 16:24 22:8,10 23:19 25:14 29:9 34:19 63:12,14,18 64:21 66:14 67:1 71:17 78:2 85:20 89:23 91:5 95:24 108:13,23 129:19	addressing 103:10 104:20 admitted 28:21 admonished 91:22 adopted 77:24 advance 17:16 advantage 70:24 115:3 advice 20:18 21:12,14,16 advised 98:15,18 advising 96:3 98:6 affect 132:8 afternoon 5:11 6:20 44:20,21 agents 58:15 ago 131:5 agree 61:14 91:13 107:21 129:13 131:22 132:10 133:6 agreed 14:7 103:1 103:16 104:11 123:14 agreement 46:2 75:1,6,7,18 air 34:1 airplane 85:20 al 1:5,8 2:2,9 algebra 35:7,10 alias 7:3 110:7 112:2,7 130:16 aliasing 99:13,17 99:19 110:23 111:5,10,17,18 111:20,23 112:18,24 113:3 113:4,10,15,17	allocated 132:5 allocates 86:4 allow 28:2 65:4 90:17 allowed 10:25 allows 115:7 all-time 80:1 Amended 123:24 amount 7:15 25:17 Ana 1:16,23 5:1 analysis 29:6,9 30:21 41:2 42:9 44:7 62:3 72:1 73:8 132:10 analyze 55:21 119:19 analyzed 97:7 101:20 analyzing 9:15 Anderson 60:10 62:9 Angeles 2:18 answer 13:22 28:9 54:19 antenna 19:7 anybody 16:11 64:8 anymore 89:14 anytime 67:5 99:19 anyway 34:25 35:24 apart 80:19 123:15 APPEARANCES 2:1 appeared 49:3 appears 6:9 74:7 Apple 21:22,25 22:9,24 23:10 23:20,22,25 24:1,2 63:6 64:2 65:9,20 68:2,2 68:14 69:10,12 70:8,18,22 71:1	Apple's 68:5,8 84:9 application 40:24 40:25 64:22 65:25 66:1 132:1 applications 66:1 115:18 applied 98:21 appreciate 16:25 appropriately 117:3 approve 71:22 74:8 approved 84:13 approximately 8:7 71:1 April 1:17 5:1 135:12 area 34:2 60:20 71:18 areas 61:21 132:14 arms 30:20 102:10,10 array 14:20 arrays 14:22 arrive 10:15 arrived 10:14 article 60:2 61:12 62:16,25 63:2,4 132:20 asked 7:13 16:3 16:10 18:24 19:2 24:19 25:21 49:15 50:2 54:15 75:4 75:15,16 79:23 92:24 93:15,16 95:3 108:4 111:22 119:3 120:5,10,22	assignments 44:11 associated 59:3 ASSOCIATES 2:3 assume 119:23 assumed 33:4 assumes 27:24 113:4 assuming 81:5 101:9 assure 5:13 ATR 40:24 attached 47:6 48:10 attack 7:23,25 8:1 9:3,11,13 11:9 11:12,12 16:15 17:19,24 27:19 29:20 31:3,7 33:20 35:2 36:9 36:11,23,25 37:2,3,6 41:14 42:12,12 43:9 43:13 64:5,15 72:16,17,18,24 73:6,13,18,21 82:7,22 83:2,6 88:8,21 89:18 91:13,18 92:18 92:19 94:2,3,7 95:20,21 99:15 99:22,23 102:2 106:15 109:5,7 110:4 111:4 114:12,16,24 115:1,16 116:2 116:4,6,8,10,11 118:7,9 121:25 122:1 attacked 57:8

attacker 9:10 33:17,22 34:12 34:17,20 35:11 35:17 36:23 37:23 38:1,3,6 38:22,23,24,25 39:14 88:19,20 110:15	back 5:7,12,14,19 5:23 6:8 8:11 10:17 12:13,15 13:15 14:5 19:7 40:3 53:22 55:18 67:1 75:4 86:10,12,18 87:2,2 88:15 92:4 97:6 98:14 116:19 122:12 132:10	125:2 126:10 131:4,8 believed 132:4 belt 35:25 Bender 5:7,13 benefit 13:6 14:8 Best 20:2 beta 74:25 better 37:9 43:13 100:2,5 101:4,8 beyond 7:9 big 15:4 67:9 127:5 bill 19:17 binaries 64:21,25 65:5,8,19 66:18 66:20 biologist 40:12 bit 7:12 8:11 11:12,14 26:20 37:7 42:19 53:11 66:19 71:12 119:1 127:2 bits 8:15,18 33:25 bitty 14:23,23 black 58:5,8,10 58:12,14,14,22 59:3,9,12,16 69:22 73:1,19 74:4 79:20 82:19,20 83:4 84:6 89:10 92:17,21,25 93:2 95:21,23 96:15 97:25 98:9,10,16 101:20 102:17 102:20 blanket 10:12 blocker 134:3,6,9 blocks 134:9 blueprint 30:5 board 61:10,11,12 61:13 bono 13:23 bothered 17:2 97:9 bottom 41:5 48:6 61:4 70:16,17	72:5 76:19 84:3 102:24 107:9 122:15 bought 19:9 bounce 86:23 bounds 7:9 96:13 117:20,21 119:9 131:16 box 8:12,14,18 9:10,22 10:3 32:9 73:1,19 82:19,20,23 83:4 92:17,21 92:25 93:2 95:21,24 96:15 97:6,25 98:9,10 98:14,16 101:20 102:17,21 128:4 break 13:2 16:24 22:17,23 23:18 25:15 57:17 64:16 91:20 breaking 13:11 13:11 14:6 19:20 56:17 58:17 breakthrough 15:5,21 brief 13:25 14:5,5 68:11 briefly 6:24 8:2 9:17 63:5 broadened 58:14 broader 67:4 broke 6:20 17:8 19:5 broken 117:7 brought 16:12 18:23 browsers 15:11 buffer 6:25 7:20 8:10,19,23 9:6 29:21 32:9,10 32:25 33:1,2,5 36:3,10 37:21 37:24 38:1 40:2 41:6 42:18 64:4 64:9,17 66:15 66:22 67:2,6,12 67:15 73:25	77:24 80:3 88:23 91:15,17 92:13,16,18,22 93:2,5,10,12,17 93:21,22 94:2,3 94:4,7,14,18 95:11,18 96:5,9 96:12,15 98:7 98:15,25 99:12 99:18,23,25 100:6 101:11 105:5,9,13 109:5,7,19 110:3,8,10,14 110:20 111:4 112:1,5,17 114:12,16,18,20 115:10,12,15 116:18,21,25 117:15,18,23 118:9,10 119:7 119:10 120:1,15 121:4,13,14,20 121:21 125:5 127:14 128:2 130:15,21 131:1 131:9,13,16 132:3 buffers 86:22 118:2,6 119:25 bug 79:19 82:5 build 73:13,18 74:1 129:18,25 130:5,15 building 43:9,14 built 14:19 19:6 105:16 134:14 built-in 42:25 103:25 104:1,25 bun 129:25 bunch 10:4 14:20 31:17 42:16 59:6 86:4 burn 75:19,20 business 58:20 63:25 butt 49:11,22 button 128:1 buy 12:15,17 18:24 56:22
attacker's 30:4 33:19 37:25 38:20	backing 92:11 backwards 39:14 bad 10:22 11:8 36:13 76:20 120:4,14,15 121:17 bag 16:13,17,18 16:18 Barr's 56:8 based 11:9 29:9 44:6 97:1 122:20 124:3,14 129:5 130:24 131:7 basic 35:7,10 115:15 basically 7:7 12:23 13:2 14:13 24:5,16 29:17 37:22 38:6 68:16 69:2 72:17 77:23 84:6 115:15 Bates 46:18 47:3 Bay 54:21 55:5 beam 62:3 began 101:17 Beginning 100:19 behalf 6:15 26:10 44:16 92:7 behavior 9:16 belief 131:12 believe 17:15 28:10 39:24 51:24 60:3,15 65:1 89:7 90:25 99:15 101:10,19 108:21 117:19 118:8 122:9,15			
attacking 74:19				
attacks 31:13 40:5 41:16 62:3 62:18 64:9 77:24 101:13 119:7				
attempted 56:6				
attended 58:8				
attention 61:3				
attorneys 2:5,11 2:17 22:2 118:24 119:2 124:12				
attributed 46:14				
audience 80:5 81:7				
August 70:22 74:5 80:15,17 122:17				
author 119:24 121:13,23 122:5				
Authority 5:9				
authorized 12:17				
available 8:22,22 9:1 55:12,15 111:8				
Avenue 2:17				
Avi 4:4 6:16				
avoid 131:15				
Award 20:2				
aware 57:18 91:9 91:12 96:9 97:5 97:8 101:12 111:12 112:21 119:2 122:7,9 128:16				
B				
B 17:2				

buzz 22:20	Cambridge 60:8	35:14 41:24	Charlie 58:6 69:7	cite 80:4
byte 8:5,23,25,25	60:10	43:2 44:7 46:13	76:7	claim 100:10
9:2,9 31:19 32:2	capability 25:14	46:19 48:3	chart 8:6 31:9	121:20
32:7,13,15,21	capable 117:12	50:18 52:2 53:5	cheaply 53:23	classes 44:10
33:3 35:19 39:2	car 16:14,15	54:3 55:19	check 110:19	classic 80:1
39:6,16 41:13	carbon 104:15	56:22 57:16	117:21,23	cleaner 109:10
103:7,12 104:11	card 7:10 8:13,16	58:13 63:4	118:16,21 119:9	clear 68:7 92:11
104:17	9:23 11:2,7 12:7	73:10 74:1 93:8	119:21 120:1	clearly 37:19 43:5
bytes 32:19 33:9	12:10,16 31:7	97:10,12 99:2	121:4,13,17,20	clerk 5:16,17
33:13,15 37:15	31:10,14,24,25	103:23 122:21	130:20 131:1,9	clever 34:12 35:4
37:16,18 38:5	32:1,10 33:4,13	123:23 133:7	131:16 132:1	cleverly 39:2
81:18 83:1	33:14 34:17,18	134:18	checked 94:18	104:9
84:22 93:22	36:21,25 38:4	cases 13:23 21:13	117:18 118:4	click 75:5
94:8 105:8,14	38:16 40:4 41:1	cat-and-mouse	119:24	clicked 75:17
111:6 112:2,7	41:18 43:1	78:3	checking 86:24	client 97:9,15
114:18,19	48:25 49:13,14	cause 77:25 83:25	96:12 117:20	118:19,23,25
115:24 128:3,5	51:13,15 52:1	112:20	120:15 128:24	clients 26:10
132:8	53:25 55:12	caused 10:9 82:5	checksum 9:18	Clint 28:18
byte's 128:5	60:8,18 62:1	causes 9:2 10:2	10:12,12,22	clock 14:11 127:9
B-A-R-R 56:9	78:16 88:3	30:2	11:8 30:1 36:13	127:9,10,11,17
	91:15 93:12,17	causing 83:17	36:14 39:17,17	127:19,21,25
	94:13,17 95:16	caution 25:8	39:19,19 40:10	128:8,13 131:11
	98:22,25 99:13	caveat 28:1	107:7 113:22	131:15
	100:2 101:11,15	Center 2:12	117:2 130:17	close 129:24
	105:10,17,17	CENTRAL 1:2	Cheese 46:14 48:7	closely 77:9
	109:7,9,20	certain 45:24	50:10	coauthored
	110:20,24 111:2	77:19 90:12	chip 7:16,20	132:22
	111:5 112:10,12	103:25 115:3	12:21,21 13:14	code 11:1 24:4
	112:14,15 113:9	CERTIFICATE	14:14 19:4 21:1	29:17 30:4,9
	113:17 114:2,6	135:3	27:7 30:14	31:3,5 32:6
	114:8 127:2,24	certify 135:5	34:13 37:8 39:6	34:13,22,23
	128:4,6,19	cetera 38:12	51:1,4,7,10,25	36:4,16,17,19
	129:18,19 130:5	Chad 2:4 6:14	52:15 53:14	36:20,21,24
	130:9,10,12,13	chain 12:5	60:16 127:8,9	37:9,16 38:25
	130:21	challenged 16:19	chips 16:14,15,16	39:9,11,15,23
	cards 50:25,25	chance 71:13	16:17,18,18,21	40:8,11 41:7,17
	54:22 62:14,17	change 7:1 33:19	52:22 60:14,21	41:21 42:12,17
	91:10 95:14,19	35:7 79:18 82:1	61:1 111:9,13	42:22,23 43:1
	95:24 96:22	82:2,3,4,5,6	choir 84:6	43:20 49:7,12
	111:9 134:11,15	87:1 114:19,24	chose 85:10 90:24	50:14 51:9 52:4
	card's 38:5	126:2,4 128:2	120:1	52:4,6,8,10,17
	109:24	changed 10:1	Chris 133:14	53:8,9,10,14,16
	care 38:7 127:19	97:19 114:7	Christine 2:4	53:18,21,22,24
	127:23	125:24 126:1	28:16	54:9,22 55:17
	careful 25:5	changes 33:16	Christophe 94:20	55:25 60:21,25
	carefully 29:8	changing 9:4	127:1,16 128:21	63:14 64:22
	carried 18:1	channel 10:1,2	Christopher 54:3	66:6,7,15 69:6,7
	CARTER 1:3	128:1	54:5	73:19,21,24,24
	case 5:15 21:12	character 128:11	circumstances	74:1 75:8,11,13
	22:3 29:5 34:18	charge 21:14	57:6	78:12 82:23,24

82:25 85:17,22 87:2,5,15 88:2 88:11,17,17,18 88:19,19,21,23 88:23 89:6,8,11 90:3 93:16,18 93:20,25 94:5,8 94:12,17,24 95:5,6 97:12 98:24 103:23,24 104:9 105:4,24 106:5,14,19,21 107:6,18 108:5 108:15 109:2,3 109:15 114:5,23 115:7,12 117:4 117:9,17,19 118:1,2,13 119:4,8,11,20 119:24 120:3,4 121:1,3,6,12,23 122:5,8 127:12 127:14 128:8,10 128:11,18 130:9 130:25 131:12 133:16 135:6 codes 52:12 69:3 91:9 coding 43:20 collaboration 44:12 collected 132:3 combination 34:24 come 8:25 17:16 20:3 21:5 26:11 35:9 45:6 46:6 91:24 116:10 128:5 comes 8:17 32:8 90:5 comfortable 26:17 coming 54:13 comment 53:20 128:8 commented 55:17 commenting 121:15 comments 120:6	120:6,17 121:1 127:13 131:7 commercially 115:16 common 64:5 86:20 110:15 115:19 commonly 77:24 communication 8:9,10 10:9 14:17 15:3 communications 93:10,12 105:5 105:9,13 110:20 118:10 121:4,14 121:20 130:21 community 79:21 124:21 125:22 compact 37:10 43:1 104:9 companies 24:8,8 57:18 company 11:19 24:4 25:19 26:18 56:24 63:6,10,17,21 64:2,14,19 66:10,14 67:18 68:18 69:18 71:5 77:6,11 81:6 82:12 89:8 89:25 90:4,25 96:3 98:6 100:4 100:10 112:1,4 115:3 compare 30:15 102:17 compared 17:11 comparing 44:3 comparison 30:21 55:9 102:1 competitor 26:19 competitors 26:12 Complaint 123:24 complete 29:25 107:9 completed 50:4 completely 11:9	94:13 128:22,25 130:17 complex 35:7 complicated 94:20 130:3 complies 48:14,19 component 8:2 9:14 41:15 components 6:22 7:2 29:10,19 43:16 115:15 116:24 125:19 composed 85:15 compromise 62:13 compromised 133:9 compute 85:21 computed 9:23 computer 7:11 16:18 31:24,24 31:25 36:13 39:6 59:10 64:6 64:8 66:3 67:1 68:23 69:1 85:14 86:1,7 91:7 120:18 127:2,3,4 computers 14:23 14:23 15:10 17:4 57:17 concept 67:3,24 conceptually 52:17 concern 46:2 concerned 6:5 131:11 conclude 116:7,14 concluded 119:11 conclusion 113:20 116:10 119:6 conclusions 123:12 conditional 50:22 62:13 conduct 29:5 55:24 conducted 69:5 conference 15:12 15:13 17:22	18:10 20:2 58:5 58:14,19,22 59:3,8,20 69:23 74:5 89:10 135:10 conferences 58:8 59:12 confirm 113:18 conformance 135:9 confused 131:6 conjectured 132:12 conjunction 59:17 115:12 connected 19:8 consecutively 9:2 34:7 consider 57:13,14 63:13 considered 17:4 51:1 96:13 123:20 consistent 41:18 constantly 15:9 15:10 constraints 97:21 98:3,21 101:15 128:7 consult 32:3 consumers 14:8 contact 68:14 contacted 20:12 20:13,25 21:3 22:24 contacting 22:2 contain 29:4 contained 107:22 107:25 108:2 content 26:1,1 contents 95:16 continue 6:12 107:16 Continued 6:18 contraption 129:23 contrast 24:16 controlled 9:9 Controlling 84:19 controversial	13:5 25:3 convenience 12:14 conversation 5:15 81:9 120:25 convoluted 130:2 cool 25:10,14 cooperation 56:21 copied 8:19,23 33:5,9,10 34:7 35:19 36:4,6 39:16 128:3 copies 31:9 74:25 copy 4:12 28:17 32:14,18 34:3 36:7 52:24,25 74:3 104:15 copying 36:6 37:15 44:12 52:20 copyright 25:25 75:18 copyrighted 26:2 26:3,5,7 CORPORATI... 1:5 2:2 correct 11:17,21 20:22 21:7,11 26:24 27:4,9 35:1 40:19 41:3 41:25 43:17,24 44:24 45:1 46:19 47:4,7,12 48:11,22 49:4 49:24,25 50:5 50:12,16,17 51:9,13,18 52:1 52:5,6,13 53:10 53:14 54:2,9 56:7,11 57:1,11 59:14 60:14,18 62:6 63:6,11 64:6,9,12 65:20 66:8 67:2,22 68:20,24 70:8 71:23 72:4,12 72:13 75:2,22 76:13 77:19,20 78:6,10,13,16
---	---	---	--	--

78:23 79:9,12 79:15,16 80:23 81:19,20,22 82:14 83:8,19 83:25 84:23 90:11,15,18 91:11 92:23 93:6,10,13,14 94:2,9 95:16,22 96:2,6,7,10,23 96:24 97:7 98:1 98:17,22 102:3 102:5,15 103:3 103:5,7,10,17 104:3 105:2,17 105:19,22 106:13,23 107:14,16,19,23 108:3,15,19,24 109:5,14,20 110:3,12,16,21 110:25 111:13 112:18,21 114:2 115:7,10,11,13 115:22,23,25 117:4,16,25 118:10,17 119:7 119:13,16,17 120:4 121:4,21 121:25 122:18 122:21 123:6,20 124:1,4,9,17,21 124:25 125:19 125:22,23,25 126:3,6,9 130:22,23 131:1 132:20,23 133:3 133:12,25 135:7 correctly 20:7 43:21 74:21 109:1 111:25 corresponds 31:17 cost 129:12 counsel 2:1 5:6 6:3,12 13:9 14:4 20:12 22:3 45:24 68:1,8 71:13 100:21 102:13,13	122:23 125:15 131:21 count 93:21 127:17 counted 39:14 94:8 counter 85:4 countermeasure 78:3,5 countermeasures 59:14 62:17,25 63:1 77:23 counting 107:7 country 15:10 County 5:8,10 couple 44:22 123:8 128:18 course 6:6 10:7 52:7 123:4 court 1:1,21,22 5:5,13 6:2,8 19:12 28:1 44:15 45:24 46:7 47:16,18 49:20,22,24 54:12,13 70:2,4 74:12,14 91:19 91:22 92:4 100:19,21 102:13 106:11 125:9,12,14 126:18 135:15 courtroom 26:22 cover 78:20 crack 15:23 16:21 20:9 22:10 64:15 cracked 16:7 cracker 14:19 15:5 16:24 19:5 crackers 15:6 craft 9:7 crafted 35:18 104:10 114:16 114:17 crash 38:16 83:15 83:18,23,25 87:20 crashes 83:22 crazy 67:8 129:21	129:22 create 89:17 90:17 106:2,7 114:1 created 20:16 40:18,22 133:2 creating 113:22 creative 120:19 creators 49:13,14 credit 12:7,10,16 24:11 63:21 credited 24:5 critical 128:3 CROSS 4:3 cross-examinati... 44:15,18 92:6,9 cryptographic 12:8 14:25 CSR 1:21 135:16 curious 76:8 current 62:14 customers 24:19 cutting 62:2 cyber 72:21,21 cycle 127:11 cycles 127:17,19 127:21,25 128:9 128:13 131:12 131:16 C3 38:23 39:1	deals 56:9 Debbie 1:21 135:15 December 28:25 29:1 44:5 45:5 55:13 56:1 91:14 92:12 98:1,8,10,13 99:3 123:25,25 124:8,16,20 125:5,18 126:6 126:8,10,11,13 decide 13:8 decided 38:25,25 131:16 deciding 19:22 decimal 17:4 decision 14:8 119:13,17 decisions 119:12 decrypting 10:1 deduce 117:9 deduced 117:2 defeat 25:20 DEFENDANT 2:9 defendants 1:9 6:23 7:22 27:7 27:15,21 29:13 40:25 131:20 133:2 defendant's 9:15 26:23 30:17 41:9,24 44:3,8 56:3 133:17 definitely 60:24 degree 7:6 104:16 deja 29:18 delay 70:22 deliberate 119:13 119:17,20 121:3 deliberately 110:19 118:16 118:20 119:25 delivered 91:10 demonstrate 31:12 37:5 demonstrates 29:25 demonstration	6:24 41:18 43:6 demonstrative 30:23,25 denied 100:21 depends 65:7 95:17 134:13 deployed 100:1 deposition 21:21 54:2,24 60:3 62:20 63:5 75:3 79:23 80:8,10 80:12,16 93:15 96:25 100:18 101:9 111:15 113:25 119:2 121:7,11 123:7 133:7 depositions 54:16 55:9 derive 108:9,14 108:23 derived 44:8 108:6,11 109:13 deriving 97:12 describe 17:18 24:15 30:19,20 61:25 104:6 described 67:4 83:24 94:1,9 116:17 describing 72:3 72:23 115:14 description 27:18 83:5 design 51:12 119:12 designed 26:6 115:18 128:7 134:6 designers 10:4 11:2 97:22 132:1 designing 101:17 detail 27:11 43:6 87:18 detailed 70:19 details 15:20 17:23 70:21 78:22,25 83:7 83:10 84:13,21
D				
D 2:4,16 4:1 17:3 Dalla 54:3,5,7 damage 132:14 DARIN 2:10 data 42:16,23 62:1 83:18 87:20 105:17 111:12,16,18,21 123:19 date 5:10 24:2 132:19 135:12 David 1:3 2:11,22 26:22 34:17 117:6 131:21 day 1:8 5:2 12:24 15:15 64:13 125:24 days 25:16				

90:6,16,21 determination 12:25 22:14 determine 20:8 22:11 29:9 53:23 55:25 67:2,11 113:2 132:3 determined 20:24 110:10 121:12 determining 8:8 113:8 develop 23:19 89:24 129:7 developed 6:23 7:22 19:21 21:10,10,13 29:12 40:25 133:23 developer 75:24 76:2,3 130:12 developers 33:4 developer's 75:1 75:13 developing 43:4 73:4 101:18 103:24 133:11 development 73:20 96:4 device 12:4 65:23 66:4 85:15 86:7 92:17,21 93:2 96:15 126:23 127:7 134:3 devices 14:18 55:21 devise 113:3 devoted 22:25 DF 35:3,4,9,18 38:22 diagram 126:4 diagrams 123:9 126:2 difference 104:17 105:12 differences 37:5 42:4,8 43:15 109:2 different 7:10 10:5,15,21 28:5	30:9,13 37:4 42:13,19 43:3 43:20,20,21,24 43:25 44:2 46:1 46:1 52:12,19 61:21 63:13 66:20 72:23 103:3,5,7,9,10 103:12,17,20 104:2,5,8,12,20 105:2 120:10,11 120:21,24 121:16 differentiate 73:7 difficult 101:16 108:18 digit 7:17 Digital 25:25 direct 4:3 6:13,18 61:3 71:9 directed 67:12 directors 61:11 disagree 94:12 110:22 119:14 131:21 133:19 134:19 disagreed 117:14 117:17 disassembled 110:18 disassembling 66:7 disassembly 55:17 disc 75:19,20 disclose 84:9 118:24 disclosed 89:8 100:13 101:7 discloses 81:21 111:19 disclosing 84:13 88:2 disclosure 17:15 19:21 23:10 24:24 27:22 28:11 101:3,4,8 discover 66:15 67:15 73:22 113:1,10	discovered 12:19 66:22,24 72:10 123:7 discovery 96:5 discuss 23:13 58:17 59:13 91:22 125:19 discussed 6:23 27:14 46:5 62:10 100:7 discussion 45:9 DISH 54:11,18,20 54:25 124:21 125:21 disk 53:17 displayed 28:19 31:1 48:16 59:25 83:12 84:17 88:6 89:2 107:3 dispute 94:16 111:16,19,22,24 124:15 disputed 133:22 dissected 29:7 dissolve 131:13 distinction 57:15 distracting 5:19 6:8 District 1:1,2,22 DMCA 26:1,6 DNA 30:7 31:12 40:13 41:2 101:24 102:15 102:17,21 116:12 DOC 1:7 doctor's 46:1 document 28:19 28:23 31:1 46:19,23 47:2,9 47:11,19 48:16 59:25 83:12 84:17 88:6 89:2 107:3 documentation 70:19 111:11 documents 48:3 55:3,16 108:10 111:8 133:8	dog 129:24 doing 9:5 13:3,6 14:2 17:21 22:4 24:21 26:17 50:19 63:21 72:7 73:8 83:5 86:10 87:3,16 128:20 dollars 129:13 donations 20:17 doubt 44:6,9 62:22 128:16 Dov 2:23 download 75:17 75:19,20 76:10 91:7 downloaded 75:6 75:7,8,10 Dr 4:10 6:20 11:11 20:6 24:12 28:14,23 40:16,17 44:3 44:13,20 46:12 58:6 65:15 67:13 71:5 75:5 75:10 79:24 81:7 92:11 106:13 126:22 129:3 130:24 133:6 draft 50:6,7 draw 55:8 drawing 121:8 drawn 37:21 123:11 drew 35:12 119:6 drive 18:24 75:20 drove 19:6 due 117:15 132:7 dull 120:18 dump 46:13 48:7 48:24 49:3 50:11 87:20 95:16 108:15 109:9 124:17 Dumping 41:6 dumps 36:19 55:11 duty 101:11 D5V3 1:25	E E 4:1 17:3 earlier 29:3,19 31:4 58:13 86:25 101:22 102:5 116:25 118:22 early 55:18 76:4 133:24 134:12 easier 82:25 easily 82:7 easy 77:21 93:5,6 113:3 EBERHART 2:11 EchoStar 1:5,25 2:2 6:15 7:20 26:24 27:2 41:1 48:6 49:7,14,15 49:17 54:22 60:18 91:9 92:17 111:9 129:11 EchoStar's 6:22 27:4,9,16,22 41:11 44:4 133:3,9,16 educating 100:8 EEPROM 36:20 41:6 42:24 46:13 49:3 50:11 98:24 109:9 124:17 EFF 20:13,15 22:2,6 effect 7:3,19,23 29:23 33:8,10 36:10 40:2 41:7 73:10 125:6 130:16 131:10 132:7 effective 134:2,10 134:16 effectively 22:8 94:1,13 efficient 43:1 efficiently 53:24 effort 5:14 13:16 efforts 20:19
--	--	---	--	--

21:21 26:11 27:4,8 eighth 12:12 either 37:5 38:13 38:17 43:12 55:25 117:15 133:7,17 elaborate 71:12 electronic 13:19 14:2 18:6 20:13 20:14 95:17 electronically 95:14 elements 113:21 else's 12:10 56:17 Embarcadero 2:12 embedded 51:1,4 51:6,10 60:21 60:25 embedding 85:22 employed 133:1 employee 77:6 employees 26:23 49:17 63:18 enable 17:25 enabled 72:4 encouraging 14:6 encrypted 68:23 69:1 encryptions 12:16 ended 128:24 engage 20:20 23:16 24:25 26:11 engaged 21:21 engaging 27:3,8 engineer 26:23 engineering 11:14 11:16 24:13,17 24:23 27:21 28:11 56:14,24 57:21 59:13 engineers 24:4 59:17 130:20 ensures 10:12 enter 17:2 entered 17:1,3 entire 44:24 55:17 128:6	entitled 61:23 70:7 73:1,19 79:5 81:12 82:19 83:14 84:19,25 89:4 123:19 entitlement 130:7 entry 82:1 envision 114:11 equally 117:12 equipment 13:1 15:4 19:2 EROM 54:17 err 25:8 error 9:18 10:9 34:21 113:13 escalating 113:12 essence 131:17 et 1:5,8 2:2,9 38:12 ethical 13:25 24:12,23 27:21 28:11 evaluate 56:18 67:18 evaluated 50:21 evaluation 70:8 73:16 82:24 evaluations 56:17 Evaluators 11:20 26:18 56:11 eventually 14:17 23:25 everybody 71:18 evidence 4:9 27:25 28:21 47:21 59:24 70:5 74:15 96:8 96:12 97:24,24 107:2,24 108:1 108:10,21 109:4 119:5 122:20,23 133:7 evolved 59:8 exact 22:1 29:19 29:21 83:7 89:22 exactly 12:21 23:4 40:7 66:17 69:10 105:7	115:18 examination 6:13 6:18 126:20 examine 119:19 example 25:9,9 52:22,23 58:15 72:19 73:9 74:2 78:1 82:13,22 85:18 103:21,23 114:15 121:10 128:1 exceeds 132:3 exception 9:16 10:24 88:14 113:22 114:1 117:1 Excerpt 4:10 exchange 12:8 24:10 excited 13:8 14:12 exclamation 79:19 exclusive 18:7 excuse 52:5 executable 78:12 88:2 116:18 execute 41:17 69:6 114:6 executed 78:13 executes 128:10 128:11 executing 36:16 39:20,23 40:11 41:7,22 87:24 127:8 execution 115:7 exercise 56:4 Exhibit 4:9 28:15 45:18 47:1,15 47:21 59:23 69:15,16,25 70:5 71:23 74:3 74:10,15 76:11 78:19 82:17 89:1 102:23 107:1 122:12 132:16 EXHIBITS 4:8 exist 103:25 111:2 existed 132:15	exists 113:6 expect 43:9 experience 11:16 44:7 expert 41:24 46:6 50:18 56:13 129:10 expertise 64:8 experts 56:3 explain 7:3 8:4 42:7 52:15 65:22 120:12 126:24 129:16 129:22 131:3 134:5 explained 98:19 127:16,24 explaining 97:24 explains 41:13 43:7 81:14 explanation 98:20 120:2 121:24 122:3 131:4 explanations 121:19 explicitly 119:24 exploit 73:5,5,6 73:23 74:2 80:22 81:15 89:6 99:17 110:2,7 exploitation 73:1 77:3,16 82:19 82:20 exploited 64:3 72:11 99:14,16 116:6 exploits 29:21 73:14 explored 25:11 express 91:23 extract 53:24 54:21 64:21,25 66:14,18 extracted 66:20 extracting 60:21 60:25 61:25 Exxon 12:1,3,5,21 13:10,12 15:19 16:6,7,14 18:12	18:22 20:9 21:3 21:14 22:6 23:8 e-mail 47:6,23 65:25 e-mailed 133:14 <hr/> F F 17:3 44:11 fact 14:7 32:8 33:9 36:1 44:9 52:3 64:11,18 71:16 115:6,9 117:9,10 122:10 128:7 131:22 factorial 85:21,22 85:23,24 86:3,3 86:9,11,13,17 86:23 factors 110:1,2 facts 27:24 failed 110:19 118:21 121:13 failure 119:21 121:4,17,19 Fair 121:15 fall 33:6 129:25 false 121:24 familiar 12:3 14:3 24:12 25:13 49:6 51:25 58:1 60:4 62:19 97:2 116:4 129:20 familiarity 53:13 family 60:13 111:8,13 fancy 14:22 84:7 far 7:8 28:5 30:5 37:23 44:24 103:13 fast 15:2 127:4,20 faster 15:22 16:24 fatally 122:1 FBI 58:15 feature 12:14 99:13 112:12 features 124:20 Federal 1:21 feel 26:19,21 feeling 36:8 felt 15:25
--	--	---	---	--

FF 36:14	93:3 95:12	founder 56:10	G	26:8 31:13
FID 19:4	97:14 99:23	Fountainview 2:6	Gale 1:21 135:15	33:25 34:15
field 115:16	101:17 107:9	four 6:22 11:10	game 23:16 78:4	39:20 40:10
130:14	109:19 110:3,15	29:19 30:11	garbage 37:22	46:3,10 47:2
fielded 62:13	111:14	41:5 61:11	gas 12:6,6,15,18	48:13 49:2
field-program...	five 15:9 127:22	84:22 116:24	18:24,25 19:6,8	58:25 61:22
14:21	fix 18:15,17,17	fourth 9:14	19:9,14,18	71:15,21 72:4
figuratively 34:22	19:24 21:11	103:14 123:24	gate 14:21	72:20,25 74:3
figure 13:5 34:23	23:5,19 24:10	FPGA 14:21	gather 59:13	76:11,24 78:2
82:5,7 83:1	93:6 98:15	FPGA's 15:1	gears 37:11	78:18,19 79:3
108:18	100:12,15 101:1	fragments 55:14	general 13:9 14:4	81:1,24 82:16
figured 22:17	101:6 133:24	frame 86:4,13,25	20:12 22:3 67:5	82:21 83:1
35:6 53:22	134:7	Francisco 2:13	68:8 93:7	84:16,25 86:10
101:21 107:24	fixing 98:7 101:3	frequency 14:14	generally 79:8	86:11,18 88:5
108:22 109:4	134:8	friendly 81:8	generates 63:25	88:16,25 95:6
file 48:10,21	flavor 52:15	front 132:17	generic 36:23	107:5 123:18
53:17	flawed 122:2	Frontier 13:19	gentleman 47:24	124:5 130:14
files 53:20 123:23	flexibility 11:1	Frontiers 20:14	gentlemen 6:21	goal 25:1 33:19
fill 37:21	focus 7:1 8:2	full 16:13	7:4 22:15 28:24	100:11,25
filler 115:24	28:15 46:9	fun 77:3,17 80:2	42:7 129:16	129:18,24
filmed 18:21	focused 62:2	function 42:25	131:3	goes 7:8 87:10
final 50:6	focusing 45:20	85:21,23,24	getting 19:14	103:23 105:25
finally 9:14	121:11 122:14	86:2,6,16,21,22	25:16 34:10,11	109:7 127:8
128:22	folks 58:19 74:25	87:8 104:1,24	36:8 57:2 66:6	going 5:21 9:5,6,6
find 5:19 24:9	follow 14:15	104:25 107:16	85:1,3 131:6	9:8 15:20 17:13
25:6 27:14 38:9	21:24 23:9	functional 106:16	ghost 7:3,23 33:8	17:14,17,21,23
41:9 55:11	80:22	109:5,6	33:10 36:10	17:24,25 18:2,7
71:15 73:9 79:5	followed 5:13	functionality	41:7 73:10	18:14 23:13,14
79:14,17,18	22:1 27:21	34:16	125:6 130:16	23:17 24:3 26:9
118:1	28:10 45:13	functioning 73:4	131:10 132:7	28:2,6 31:13,23
findings 70:18,20	following 30:6	73:5,6	ghosting 7:19	31:25 32:2,3,5
finds 100:11	37:1	functions 8:14	29:23 40:2	32:12,13,14,15
finite 7:7	forced 100:6	85:16 103:25,25	give 7:8 18:7 23:2	32:18,18 33:12
firm 56:13 58:4	Ford 12:19 16:16	fundamental 25:1	25:9,9 28:5,16	35:9,15 37:11
68:1,2	foregoing 135:6	29:10 43:16	40:9 42:10	38:12 39:12
firmware 65:19	form 41:14 68:23	funded 13:20	44:11 70:22	42:20 45:25
65:22,23 66:4	91:23 97:11	further 5:22 21:9	76:25 81:3,4	46:6,8 49:10
66:15	formal 23:15	126:17 134:22	83:22 84:22	55:18 64:16
first 8:17 17:17	format 135:9	future 90:22 91:1	114:22 116:9	66:19 71:5,8,18
25:5 26:17	former 132:22	134:16	126:24	75:24 76:2 78:2
28:18 32:2	forms 64:5	fuzz 67:13	given 53:16 117:9	82:9 86:6,11
33:13 36:5 38:4	formula 32:5 35:7	fuzzed 116:20	119:20 121:19	87:2 122:12
46:13 52:3,9,20	forum 54:17	fuzzing 66:18,20	gives 81:18 84:21	129:22 130:5,5
52:24,25 60:4	found 15:7 17:5	66:24,25 67:3	giving 82:12	134:7
61:3 63:10,14	24:6 50:18	67:10,15,18,21	116:15	Goldberg 129:20
64:14,17,18	55:14 72:3,6,8	67:23 72:11	glitch 10:9 62:3	130:2,3
66:7 69:13,18	72:11 81:21	81:22 87:11	go 9:10,10 11:3	gonna 16:19
73:8 74:6,17	87:10 115:21	110:11 112:1,4	12:6 17:25	31:18 32:1 33:1
80:2 81:4 89:25	Foundation 13:20	113:1	19:22 20:21	33:3 34:3 35:25
91:16 92:13,22	20:14			37:2,6,12,25

38:1,23 39:9,13 39:20,23 74:22 85:12,18,24,24 91:2 128:5 good 6:20 22:22 22:23 44:20,21 63:17 67:9 79:19 83:14,14 91:19 117:20 120:3,13 Google 50:19 gosh 36:13 gotten 38:8 133:16 government 57:2 grab 36:16 grad 15:16 16:23 17:21 greater 100:11,25 101:2 group 1:8 2:9 54:11,14 guarantee 80:21 80:24 89:15,19 guess 32:19 36:12 85:12 116:3 guidance 117:7 guys 16:25 17:12 18:15 25:10,18 26:8 73:15	89:25 90:5,6,9 90:17 95:24 109:9 129:19 130:3,3,6,14 133:2,11,24 hacked 27:22 63:6 64:2 68:7 68:11,18,22 69:12 76:21 100:4,24 129:12 133:10 hacker 24:22 57:13,14,15 80:21 89:16 124:21 125:21 hackers 58:10,24 58:25 59:1,7,7 65:11 68:19 70:23 78:4 82:4 87:9 hacking 27:16 54:17 58:18 59:13 63:13,20 67:13 69:19 71:6 74:18,22 86:20 87:22 115:2 hacks 89:24 90:21 91:1 100:10 134:16 Hagan 2:4 4:4 6:14,14,19 20:5 28:13,16,20,22 30:24 40:15 44:13 46:4 47:17 70:3 74:13 126:19,21 Haifa 29:20,22 31:3 35:1 36:19 37:3 38:15 40:22 42:12,17 42:22 43:11,13 55:8 73:24 88:22,22 101:23 103:21 106:20 108:5,5 109:8 109:11 116:11 123:10 132:11 134:11 half 53:9	handed 5:16 handler 9:16 handling 10:24 88:14 hang 106:22 130:13 hanging 31:20 hanky-panky 120:20 121:1,9 happen 26:13 33:3 72:19 89:4 99:19 132:14 happened 12:23 15:15 18:9,9 90:10,12 99:21 113:18 happens 7:8 8:17 10:19,20 12:11 23:2 30:3 31:19 31:23 33:15 39:18 67:9 happy 13:18 18:12 69:12 hard 13:2 52:16 53:17 104:6 harder 98:19 hardware 10:11 57:19,21,22 59:17 65:24 66:2 hard-program 10:23 HARTSON 2:15 Hasak 47:24 Hat 57:14 58:5,8 58:10,12,14,14 58:22 59:3,9,12 59:16 69:22 74:4 79:21 84:6 89:10 Headend 27:12 36:19 41:9 43:5 44:8 45:6,13,14 50:4 101:22 103:2 105:4 106:1,6,14,20 106:21 107:22 108:2 131:23 133:17 header 36:6 37:14	heap 78:12,13,14 81:15 82:11 84:3,8 115:6 heard 27:1 33:7 45:3 49:15 54:11,14,17 57:24 96:20 97:2 121:9 122:2,2 131:8 133:22 hedging 60:15 held 19:7 59:2 92:1 135:8 help 30:22 36:22 45:20 66:19 87:22 hex 17:2,4 31:4,15 33:13 38:5 hexadecimal 103:7 104:11,17 higher 19:14 highlight 70:20 high-level 83:5 hired 24:20 26:19 56:16,19 history 59:5 hit 36:12 hits 130:1 hoc 18:4 HOGAN 2:15 hold 12:6,14 honest 76:22 honestly 133:4 Honor 6:14 28:20 30:24 44:14,17 45:22 46:4 47:14,20 69:25 74:11,13 92:8 100:17,22 106:9 126:19 HONORABLE 1:3 hope 36:8 hoped 133:24 Hopkins 11:20 12:23 14:4 16:8 16:12 20:8 24:25 hosted 54:18 hot 25:7 129:24	hour 53:9 Houston 2:7 hundred 12:20 105:14,15 111:6 hypothesis 108:13 H-E-A-P 78:13
I				
IBM 52:22				
idea 23:2 63:15				
identical 31:9 43:17				
IDENTIFICAT... 4:9				
identified 6:21 7:2 9:15 40:22 40:23 41:6 62:25 63:1 72:14 73:4 107:12 113:21 116:24				
identifier 14:14				
identifies 41:12 42:2 72:5				
identify 28:23 29:15 42:3 114:1				
identifying 72:7,8				
ignoring 7:17				
III 1:8 5:2				
imagine 62:7,21 70:14 77:5				
immediate 96:4				
impact 33:22				
implemented 62:24				
implications 20:12				
implies 58:23				
importance 127:24				
important 8:8 11:13 12:9 25:2 35:15 38:21 41:2 110:5				
impossible 17:5				
impression 119:9				
improve 21:16,18 27:15 43:11				
improvement				

109:11 improving 23:25 24:1 include 58:15 105:4 123:23 130:9 included 63:23 includes 62:2 including 70:19 76:4 incoming 132:2,7 incompetence 117:16 inconsistent 43:25 incorrect 30:1 39:17,18,19 40:9 45:11 59:19 97:18 107:7 113:22 117:1 incorrectly 10:10 increase 89:23 increases 8:24 increasing 128:13 128:14 increasing-sized 113:7 incredibly 122:4 increment 32:13 incremented 8:22 9:1 incrementing 34:4 independent 11:19 26:18,18 26:21 56:10 index 8:3,21,22 9:1,4,7,12 29:24 29:24 30:1 31:22 32:3,4,6 32:13 33:18,18 33:19 34:4 35:5 36:10 37:19 38:9,19 39:5 40:5,6 41:8,10 41:13 73:10 78:1 85:6 114:11,13,19,21 117:1 125:6,7	130:16 indicated 111:17 indication 78:9 indicative 121:25 individual 117:3 117:5 individuals 20:17 industry-wide 28:3 infinite 105:25 109:7 informants 54:25 information 15:21 17:14 19:4 23:7,13,23 53:17 55:4,11 55:14 68:25 69:2,5 71:6,19 79:20 83:22 97:2,11 98:4 109:12,13 114:8 116:9 123:13,20 124:3,11,14 133:18 informing 19:24 22:25 infrequency 23:3 inherent 7:19 input 82:13 128:23 inputs 66:25 84:19 110:11 112:11 113:12 Input/output 32:9 inquire 5:22 inside 31:24,25 83:4 129:11 insider 121:25 122:1 Insights 57:24 inspect 119:19 instance 91:16 92:13,22 93:3 95:12 110:3 instances 123:15 instruction 127:17,21 instructions 11:4 28:5 87:18 128:12	Instruments 13:12,13 16:12 intact 10:14,15,17 integral 41:14 intended 20:20 58:25 intent 24:22 76:6 intention 75:22 intentional 117:15 intentionally 9:4 10:8 33:1 117:18 130:20 interest 124:7 interested 13:10 14:1 76:4 interesting 18:3 36:2 37:15 Interestingly 12:19 Internet 37:13 48:4,22 49:3 50:10 55:12,18 56:1 99:12 108:24 124:6 interview 18:21 intricate 36:9 invalid 9:18 66:25 110:11 invented 17:19 inverted 43:23 investigation 55:24 invite 18:7 invited 16:10 inviting 19:15 involved 15:9 16:4 58:19 ion 62:3 iPhone 21:22,25 22:9,18,19,21 23:10,20 24:1 24:18 63:6,10 63:15,18,20 64:2,15,19,22 65:4,20,25 66:5 66:6,11,15 67:14 68:3,5,7 68:12,18,19,22 69:12 70:8,13	71:6 72:4,19,23 74:23 76:21 78:22 79:1 81:10,16 82:14 82:20 83:3,17 84:23 87:19,22 89:9,12,13,17 89:25 90:5,9,13 90:18,21 91:1,5 91:7 100:5,25 115:2,4,6,9,13 115:20,22,25 116:6,14,20 IRD 8:15,23 31:14 irresponsible 91:3 ISE 21:21 22:8 24:25 26:11 65:16 81:5 issue 90:24 91:2,3 91:5 94:22 97:25 128:22 issued 71:7 79:12 84:12 96:22 97:6 99:2,9,11 issues 58:17 115:19 124:15 items 116:24 iTune 25:15 iTunes 25:13,13 IV 134:23 i.e 132:8 I/O 6:24 7:20 32:9 36:3 37:21 42:18 73:25 88:23 119:10 127:14 128:2 130:21 131:9	43:19 94:12,17 103:1 107:13 111:12,16,19,22 117:14 119:11 122:3 JUDGE 1:3 judgment 56:23 Judicial 135:10 July 70:19 94:23 jump 30:2 73:25 107:12,15,18,21 107:25 108:2,6 108:14,19,22,23 jumping 39:8 105:21 jumps 41:21 jurors 5:17 6:5 jury 1:15 5:4 6:1 6:4,21 7:4 22:15 28:24 42:8 92:3 92:5 126:24 129:17 131:3 134:5 jury's 6:2 justifications 13:3
K				
Kathy 5:8,12 keep 34:4 39:8 68:25 69:1 76:22 keeping 127:25 keeps 9:1 KENNETH 2:16 key 6:22 7:2 8:2 9:14 12:5,21 15:8,18,23 16:14,15 18:18 19:5 116:24 124:15 130:11 130:12,13 keys 12:11 14:25 17:1,6,10 48:7 kind 13:22 23:16 24:11,17 25:15 26:20 29:18 37:15 59:1 64:15 66:3 79:21 80:2 83:6 84:4 104:6				
J				
Jailbreak 65:2,3 65:15 66:21 Jan 133:15 Jane 134:22 job 129:11 Johns 11:20 12:22 14:4 16:8 20:8 24:25 Jones 41:25 42:2				

kinds 9:24,25 16:4 58:16 67:8	Kuhn 60:5,6,12 62:9	left 19:10 128:9	48:17 62:14	127:12
KLEIN 2:16	Kuhn's 60:20	legal 14:1 20:17	66:19 67:4	looks 5:19 12:15
knew 18:2,25	<hr/> L <hr/>	26:3 34:18	71:12 109:10	48:5 49:6 86:17
33:22 34:13	L 2:16	38:17 99:20	127:2	loop 105:22,25
37:20 50:2 90:2	lab 12:22 13:1	legitimate 9:23	Live 134:21	109:8 128:10,14
96:12 106:18	14:3 15:18	38:14 59:8	load 76:8	128:15
117:19 121:24	16:11,23 18:8	61:14 87:3	loaded 25:4	Los 2:18
know 5:21 9:25	18:21,22 21:6	legs 102:12	loader 52:23	loss 13:7
11:8,11 15:25	54:21 55:1,4	length 42:20	loading 76:4	lost 124:22
16:6 17:8 18:14	108:5,6	93:22 132:2,8	located 8:7	lot 15:20,21,22
20:3 22:20 25:3	laboratory 60:7,8	lengths 42:13	location 7:13,14	16:24 17:5 18:6
30:10 33:7,8	ladies 6:21 7:4	43:20 103:5	7:14 39:4 125:7	19:14,15 22:20
35:9,15 45:10	22:15 28:24	Leopard 74:18,22	132:6	35:22 42:3 53:4
45:25 49:14,17	42:7 129:16	75:1,8,8,11,17	locations 132:9	55:9 64:11 76:3
50:2,9 53:18	131:3	76:4 80:22	log 79:18 82:1,4	82:25 98:19
54:5,7 59:9	lady 5:18,22 6:5	letters 17:2	87:21	103:22 123:13
62:11,24 63:1	laid-out 23:16	let's 8:2 11:3	logs 82:2,6	127:13 130:4
64:9 65:12	language 42:23	22:21 37:14	long 32:25 45:25	lots 68:19
73:12 76:17	51:25 52:4,16	76:11 81:1,11	53:11 69:1	lower 113:8
81:4,9,25 83:4	52:25 53:2,5,6	82:2 83:11	86:24	lunch 6:20 12:23
85:19 86:11	82:12 93:9	85:18 86:21	longer 18:18 34:3	<hr/> M <hr/>
96:21 98:4,13	laptop 15:12,13	109:17 110:23	39:7 67:23,24	M 2:4
99:1 100:8	19:6,8	129:24	look 34:22,22	Mac 74:19 79:5
101:17 103:22	large 24:8	level 116:4	36:15,18 37:16	79:15,18
106:15,17,17	larger 7:15	library 104:24	47:9 55:7 57:17	machine 53:16
108:8,22 109:3	laser 62:2	licker 49:11,22	61:23 62:19	66:6,14
109:13,25 111:3	late 45:4	light 15:8,14,17	70:16 74:17	machines 14:3
111:7,11 112:19	latest 82:9	lights 15:6	81:11 83:9,11	Mac's 80:23
113:4,11 120:19	laugh 80:6	likes 57:16 76:9	95:3,6 106:25	main 105:22
120:22 122:7,10	launch 129:7	limited 28:6	116:10,11 121:6	maintain 68:22
129:20 130:11	launches 130:1	line 36:5 100:20	124:23	maintain 68:22
130:12 131:10	law 2:5,11,17	103:14 107:6,7	looked 29:7,17	maintained 65:19
knowing 10:14	25:24 28:4	107:9	31:6 35:22 37:6	making 19:24
127:15	laws 14:6 26:5	lined 14:24	42:13 47:13	malicious 24:22
knowledge 8:3	lawsuit 50:21	lines 32:23 94:8	48:11 55:19	24:22 57:15
49:18 54:10	55:22	100:18 106:10	63:5 78:15 95:5	58:10,24,25
89:23	lawyers 13:19,24	123:10 127:14	111:12,14,21	59:1,7,18 65:6
known 36:23 40:4	16:1,12 20:13	128:18	114:17 116:11	68:19 72:22
46:13 62:6,7	22:6 25:5,12,21	link 109:24 110:1	117:17,19	80:21 88:17
79:9 106:4	50:18 124:15	listed 29:19	121:12	89:16 118:6
116:13,14,16,20	layer 65:23	listening 94:19	looking 15:10,11	maliciously 9:3
116:21,23	lead 113:19	97:20	15:13 22:20	87:4
knows 38:1,3	learn 48:3 66:12	lit 19:9	43:13 46:12	management
Kommerling	learned 30:13	literally 12:12	47:23 53:6,7,12	130:8
132:23 133:1	109:12	little 7:10,11,12	53:15,19,21	manipulation
Kudelski 27:3	learning 37:7	8:11 11:12,14	60:4 62:23	62:3
62:24 96:3,8,18	43:12	12:4 14:22,23	71:23,25 74:6	manual 53:3 62:2
97:25 98:7,15	led 101:4,8	15:6 16:13	80:6 89:23	manually 17:1
101:10		26:20 37:7	112:25 113:5,11	manufacturer
		40:18,19 42:19	119:5,8 125:17	21:1 27:7

manufacturers 13:14	35:16 38:12 77:19,25,25	89:16,20 Michael 56:8 58:1	moment 5:14 125:9	nature 15:1 NDS 1:8 2:9 5:18
marbles 129:25	82:13 84:23	micro 60:13	Monday 5:10 97:16	6:6,7,7 27:2 44:16 46:18
March 52:5 94:23 97:4 114:9 133:15	85:14 86:4,5,14 86:14 99:13,17 99:19,21 107:15 110:7,23 111:5 111:9,17,18,20 112:2,7,18,24 113:2,4,8,10,14 113:17,20 115:22 116:25 117:22 121:21 121:24 122:7 127:3 130:16 131:10,14 132:5 132:14	microchip 51:15 51:20 110:24 microchips 60:17 microcontrollers 60:14 microprobing 62:2 microscope 34:21 Microsoft 115:16 middle 31:20 39:3 88:15 Millennium 25:25 Miller 58:6 64:18 65:15 66:23 67:13 69:23 74:4,21 75:5,10 77:6,17 78:12 79:17,24 80:9 80:13 81:7,14 81:25 82:12 83:17,24 84:2 87:4 115:2,21 116:7,15 million 13:21 millions 129:12 milliseconds 127:20 mind 23:15 30:7 30:12 44:6 131:17 minds 131:19 minor 123:8 minute 71:21 minutes 15:24 44:25 91:25 131:5 missed 44:25 misspoke 36:5 mistake 123:12 mistakes 123:8 Mobil 12:1,3,5,22 13:10,12 15:19 16:6,8,14 18:12 18:23 20:9 21:3 21:14 22:6 23:8 Mobile 72:11 modular 85:17	monitoring 48:4 months 14:11 18:11 80:19 117:6 mood 120:3,4,13 120:14,15,24 121:17 Mordinson 26:22 27:1 34:17 40:14,18,21,23 41:5,8,16 82:22 104:23 105:19 106:5 117:6,12 131:21 132:1,25 133:10 Mordinson's 27:12 31:3 105:24 132:10 morning 9:17 15:15,17 Moskowitz 2:22 Motorola 51:15 52:10,25 move 47:15 69:25 74:10 movie 25:13,16 26:1 moving 130:4 MYERS 2:10	92:7 106:15,17 107:24 108:5,22 109:3 124:16 necessarily 59:18 81:9 113:19 necessary 41:14 106:15 109:4,6 110:4 need 16:7 40:7,8 53:2,3 85:20,21 85:23,25 86:2 86:18 107:15 130:10 needed 15:4 19:4 97:11 needs 34:23 42:17 net 13:7 network 72:21,22 124:21 125:21 networking 72:20 never 51:12 53:23 58:9 61:21 69:7 75:6,8,10 99:16 108:6 new 17:18 18:5,9 18:19 53:6 74:22 84:7 89:17 114:7 126:4 128:25 131:6 newest 74:19 news 79:20 nice 123:7 Nicolas 45:1 94:20 101:19 127:1,16 128:21 129:5 130:24 133:13,19,22 134:17 Nigel 41:25 Nipper 11:12 28:25 29:11 30:16 31:5 37:2 40:23,24 42:12 42:15,25 44:5,7 45:4,5,13 49:10
matrix 40:19,21 matter 35:24 38:3 38:13 39:1 52:19 53:7 85:15 91:23 127:21 128:18 135:8 matters 127:7 maximal 132:2,8 mean 6:7 8:4 65:16 67:23 69:2 78:7 79:11 85:3,11 87:11 87:13 98:13 104:14 108:9 127:19 130:17 meaning 14:20 means 24:15 56:17 59:18 meant 81:25 83:17 84:2 measure 104:16 mechanism 91:6 134:14 media 18:3,20 22:21 meeting 16:19 17:7 21:5 23:12 meetings 59:1 member 61:8 memory 7:7,9,11 7:12,15 8:6,19 8:23 9:1 29:23 31:9 33:6,14,16 34:2,7,14,15	Menard's 29:12 mention 20:1 55:7 mentioned 9:17 116:22 118:22 message 8:15,17 8:18 9:7,22,23 10:9,13,16 12:12 31:14 32:1,25 36:14 38:9 39:18 73:14 113:13 130:8 132:2,8 messages 8:8,14 9:24 10:2,5 14:13 32:11 44:10 113:7 messed 128:19 met 44:22 101:20 127:15 method 72:14 methodologies 42:4 43:16 103:3,10,19 104:20 methodology 6:22 9:16 19:21 29:2 29:11,12 30:11 30:16,16,17 42:22,25 44:4,5 44:8 64:23 103:20,22 133:24 methods 62:10,12	millennium 25:25 Miller 58:6 64:18 65:15 66:23 67:13 69:23 74:4,21 75:5,10 77:6,17 78:12 79:17,24 80:9 80:13 81:7,14 81:25 82:12 83:17,24 84:2 87:4 115:2,21 116:7,15 million 13:21 millions 129:12 milliseconds 127:20 mind 23:15 30:7 30:12 44:6 131:17 minds 131:19 minor 123:8 minute 71:21 minutes 15:24 44:25 91:25 131:5 missed 44:25 misspoke 36:5 mistake 123:12 mistakes 123:8 Mobil 12:1,3,5,22 13:10,12 15:19 16:6,8,14 18:12 18:23 20:9 21:3 21:14 22:6 23:8 Mobile 72:11 modular 85:17	N N 4:1 Nagra 49:7,16 62:24 96:8 97:25 101:10 110:19 NagraCard 129:11 130:20 133:23 NagraStar 6:15 27:2 53:24 92:18 129:11 NagraVision 27:3 name 26:17 45:4 58:23 59:3,9 named 132:22	

49:21,22 55:8 88:8,21,22,23 91:14 92:12,15 92:19 95:10,15 101:23 102:18 102:20 103:2 104:9,24 105:5 105:16,21 106:4 106:19 107:1 116:12 123:24 134:11 NipperClause00 124:8 Nipper1 124:9 nodding 129:21 nonethical 24:21 nonimportant 42:16 nonprofit 13:20 20:16 normal 10:5 87:8 Normally 46:2 note 5:16 6:4 notice 34:9 37:16 42:14 62:16 noticed 53:20 127:12 notified 20:19 70:18 notify 23:22 27:7 notion 7:7 nouns 43:23 November 50:4 number 12:10,16 25:24 46:18,19 47:3 81:18 99:22 102:24 105:8 125:2 128:14 numbers 17:3 32:23 127:13	objections 74:13 objective 17:13 37:18 objects 71:11 obtain 83:18 obtained 15:18 41:20 54:8 74:25 98:1 obviously 42:18 occasionally 56:16 occurred 25:18 91:8 occurring 113:15 October 46:14 47:25 49:4 50:7 50:9,13 80:16 96:10,18 99:6,9 offer 57:19 62:14 official 1:21 65:9 oh 65:1 72:17 76:14,16 131:5 okay 16:17 20:21 31:2,24 34:15 35:20 36:8,20 37:1,11,18,19 39:7,14 40:12 45:15,19 47:1 60:2 62:24 68:9 69:15 73:12 74:17 75:12 77:3 82:18 85:2 85:12 86:2 103:1 125:1 old 84:3 old-fashioned 81:21 Oliver 132:23 133:1 omit 17:23 once 15:3,7,20 22:17,22 34:2 36:23 38:11 39:7,15 52:18 53:4,19 57:11 73:3 90:18,20 115:17 128:16 ones 24:5 31:17 40:23 41:5 60:17	Ontario 54:21 55:5 oOo 134:24 135:1 open 79:17 opening 68:1 operate 34:19 operating 66:4 74:23 76:5 79:15 80:23 97:22 98:3 115:17 operation 9:23 10:5 29:25 33:17 operations 52:12 opinion 27:20 28:2,6 42:9 43:4 60:25 91:23 94:12 98:5 102:20 109:11 111:16,19 114:7 117:14 118:20 126:1 129:9,13 131:21,22,22 opinions 47:12 60:20 97:12 118:24 opportunity 27:11 41:23 126:24 opposed 81:6 opposite 42:3 58:23 options 10:16 Orange 5:8 orchestrate 40:8 order 17:19 34:16 67:11 70:22 71:17 organization 13:20 20:4,16 61:8,15,16 original 119:4 originally 36:21 74:21 118:12,15 originates 59:9 OS 74:19 79:5 80:23 Osen 122:5,7 outlined 17:16	output 128:23 outside 5:4 27:25 97:13 overflow 6:25 7:20 9:5 29:21 33:2 36:10 37:24 38:1 40:1 64:4,9,17 66:16 66:22 67:2,6,12 67:16 77:24 80:3 81:15 82:11 84:3,8 92:16,18 93:10 93:17 94:2,3,14 94:18 95:18 96:5,9,16 98:7 98:15,25 99:18 100:1,6 101:12 109:5,7,19 110:3,8,10,14 110:20 111:4 112:1,4 114:12 114:16,18 115:10,13,16,24 116:18,21,25 117:15,24 118:4 118:17 119:7,21 121:13,20 125:5 130:15 131:9 overflowed 81:19 91:15 92:22 overflowing 33:1 41:6 94:4 99:12 99:22 112:17 overflown 91:17 92:13 93:3 95:11 114:20 overflows 82:13 93:5 overriding 86:25 113:8 Overruled 28:8 106:11 overshot 35:23 overwrite 38:14 39:7 85:10 87:7 88:8 overwriting 34:15 owner 72:22 Ox1FFF 132:6	Ox200 132:6 Ox73 107:10 Ox81 107:10 o'clock 92:1 O'MELVENY 2:10 <hr/> P <hr/> pad 130:1 page 22:25 28:18 42:11 45:20 46:12 47:2,7,23 48:13 49:2 60:4 61:3 67:7 70:16 70:17 71:22,25 72:25 74:17 75:18 76:11,14 76:16,24,25 77:1 78:18,20 78:22 81:1,11 81:24 82:16,16 83:11 84:16,25 88:5,16,25 100:18 102:24 103:13 106:10 107:5 122:14 123:18 124:5,23 124:24 125:1 135:9 pages 74:6 paid 19:13,18 paper 17:9,11,18 17:18 18:10 20:1,2 61:6,18 62:7,20,21 70:20 80:1,2,6 80:11 papers 62:9 paragraph 61:23 72:5,25 73:19 123:22 parallel 14:20 parameter 52:20 parameters 107:19 108:7,14 108:18 pardon 47:2 parentheses 127:13 part 9:20 11:19
<hr/> O <hr/> O 1:3 oath 97:10 object 71:18 objected 71:16 objection 27:24 47:16,17 70:2,3 74:12				

29:5 30:15 48:21 52:4,16 63:4 64:22 71:9 73:16 88:8 89:7 107:18 108:23 120:13,14,15 partake 25:4 participated 133:17 particular 7:16 7:23 8:5 11:3,12 19:23 35:9 41:13 52:18 53:14 60:16 83:23 91:18 94:3 99:22 103:21 110:24 116:17 120:7 124:7 127:14 130:6 parties 6:10 19:24 92:4 parts 130:4 party 56:19,21,25 pass 12:2,3,22 13:11 15:4,19 16:8,14 18:23 20:10,25 22:7 44:14 49:23 passes 107:18 passing 108:6,14 patch 21:10,11 23:5,19,24 24:2 24:5 71:1,3,7,8 79:12 84:12 90:25 91:2,4,5,9 94:12,17 95:6 95:17,19,24 96:4,22 97:5,25 98:24 99:2,8,11 99:25 100:6 101:11,18 121:1 121:6 128:17,22 129:2,7 133:23 134:2,7,10 patched 89:13 90:18,20 95:14 patches 21:13 70:23 91:7 115:17	path 20:22 pay 18:25 19:12 PC 85:1,3,4 pedagogically 43:8 Peel 5:8,12 penmanship 43:24 people 13:6 22:25 30:13 33:13 43:22 44:1 50:2 58:16 59:6,12 63:12 71:9 80:5 80:25 81:7 82:8 90:2 100:8 117:11 120:10 120:11 perform 12:8 17:19 33:17 83:1 111:4 performs 12:16 permission 68:5 68:14,16 84:9 person 12:17 19:3 37:7 117:8 120:2,12,21,23 personally 65:17 phase 73:8 phases 73:7 phone 65:25 83:25 91:8 phonetic 5:8 phrase 49:6,9,18 49:23,23 50:11 50:14 physical 66:2 102:7 132:5 picked 18:20 piece 17:9,10 73:25 97:12 pieces 85:17 Pilon 54:24 piracy 54:11,18 54:21 55:1,4,21 119:25 120:2 pirate 57:16 92:17 96:15 101:13 118:6,9 132:22 pirated 134:10,15	pirates 58:11,16 134:3,7,14 place 30:3 39:15 58:16 85:23 86:5 105:8 114:5 120:7,8 120:20 123:11 places 32:10 105:5 PLAINTIFF 2:2 plaintiffs 1:6 6:15 50:18 55:3 124:12 129:6 plaintiff's 6:16 54:25 123:24 plan 13:11 16:3 17:16 23:17 planning 22:4,19 plans 23:6 play 57:17 100:17 plays 9:11 PLC 1:8 2:9 please 10:17 48:13 49:2 69:15 70:16 74:3 78:18 79:3 81:1 82:16 83:11 84:16 88:5 89:1 101:5 122:14 123:18 125:3 plenty 94:22 Plex 54:11,18,20 54:25 plural 81:6 pocket 17:11 18:23 point 15:23 19:19 22:18 24:7 30:22 33:24 34:8 38:20 39:21,24 47:15 57:2 79:19 82:21 84:11 98:2 111:23 121:15 123:8 130:19 133:10 133:20 pointed 58:13 88:15 123:12	pointer 85:8 87:11 pointers 87:1 points 11:4 42:11 42:21 policy 25:17 portion 41:17 42:5 45:23 46:8 49:20 128:3 133:16 portions 46:1 77:19 133:14 position 25:6 possible 7:21 72:24 91:14 92:12 95:11 99:24 122:4 132:2,13 post 28:25 106:14 posted 29:11 44:5 45:4 48:21 50:15 91:13 124:7 126:11 posting 31:5,6 37:12 55:8 91:14 92:12,15 95:10,15,15 98:1,8,10,11,13 99:3 101:23 102:18 104:24 105:16,21 106:2 106:7,19 107:1 107:25 108:7,23 109:3 124:9,16 126:6,9,10,14 postings 99:13 100:1 123:25 124:6 posts 103:2 power 62:3 practice 24:23 25:2 preaching 84:5 precarious 25:6 preceded 45:6,14 precious 131:15 precise 39:4 40:7 precluded 94:13 94:18 premier 20:4	preparation 123:23 prepare 46:24 prepared 37:23 68:11 preparing 123:2 presence 5:4 6:1 92:3 present 2:21 5:6 5:23 6:3,3 40:17 61:18 92:5,5 133:13 presentation 4:12 5:10 69:22 74:4 74:8 80:9,17 81:2,8 84:5 89:9 89:17 90:9,17 116:15 presentations 77:13 presented 58:4,7 61:6,18 89:11 89:13 preserve 34:16 president 56:10 77:11 PRESIDING 1:3 press 14:15 128:1 pretend 75:24 76:2 pretty 14:12 15:25 16:9 23:7 23:15 39:25 44:25 80:24 82:7 84:21 115:16 116:14 prevent 93:6,16 94:1 98:24 99:25 134:7,15 prevented 95:18 99:12 previous 21:20 50:7 82:7 89:12 previously 6:16 86:15 97:2 102:1 prices 19:14 prior 50:9,13,21 55:12 56:1 84:10 96:9
---	---	--	--	---

97:10 100:1 124:19 125:4,17 133:10 134:12 private 57:4,20 pro 13:23 probably 112:25 132:4 problem 23:4 32:20,21 70:14 101:21 113:16 128:17 129:4 133:5 problems 10:11 procedures 21:24 22:1 proceedings 1:14 61:19 92:2 134:22 135:8 process 8:16 20:7 23:15 66:10 81:14 83:7 110:11 processes 11:24 processing 105:22 127:10 128:5 processor 51:16 51:22 52:10 127:4 processors 52:13 62:1 produce 70:23 produced 46:19 product 12:4 14:9 21:17,18 23:25 27:16 56:22,25 57:8 77:21 79:18 82:6,8 100:2,5 101:4,8 Productions 46:14 48:7 50:10 products 11:17 23:1 56:14 67:19 78:8,11 100:10 professor 16:8 60:10 profit 79:19,24,24 80:2,4,7 program 10:4	30:12 36:24 39:25 42:19 43:4 44:1 51:3 51:20,22 65:2,6 65:9,13 75:13 85:4,5,14,15,19 86:12 87:3,8 105:19 106:22 107:16 118:2 127:8 programmed 36:15 programmer 37:20 43:12 120:18 programming 44:2 53:5,6 93:9 103:3,19,20 104:2,5,7 106:13 programs 30:6,9 40:1,2 43:3 88:14 103:5,9 103:17 105:1 project 15:9 19:23,23 21:25 23:11 25:10 40:25 133:18 projects 13:4,7 14:12 26:14 project's 26:8 proof 45:5,8 proper 114:22 properties 11:10 116:12 property 7:16,25 11:6 29:23 31:12 40:3 110:23 112:14 112:15 protect 26:5 62:17 93:9 protected 33:15 34:3 38:5 62:1 94:4 protection 62:14 protocol 27:21 28:3,3 protocols 24:24 25:2	prototype 74:25 proud 15:25 provide 20:17 23:22 55:16 62:17 87:18 provided 55:3 70:22 122:21,23 124:3,11 providing 23:5 public 48:7 59:2 69:10 publication 23:6 71:22 83:24 90:6 publications 70:13 publicity 63:23,25 100:8 publicized 60:13 64:11,13 publicly 111:8 publish 24:3 28:18 30:24 45:22 47:14,18 62:9 71:6 90:1 published 45:25 46:6 61:12 62:8 68:15,17 69:18 78:6 publishing 19:25 pulled 17:10 pump 12:7 19:8 purpose 8:20 13:21 14:24 31:11 125:7 purposes 35:14 43:6 48:11 63:20 pursuant 135:5 push 10:19,20 put 8:12 10:12 12:4 15:7 16:23 32:7,22 33:23 34:18 35:21,25 36:24 40:6 49:12,16 71:8 80:4,9 86:6,13 86:21 88:22,23 105:13 106:18 108:8 114:18,19	129:24 puts 42:15,17 putting 67:7 p.m 5:3 92:1,2 <hr/> Q <hr/> question 28:6 45:17 50:1,25 52:22 54:15 99:8 100:19 119:1 125:3 131:6 questions 13:22 117:8 126:17 quickly 9:19 12:11 16:9 17:6 101:16 quite 127:1 quote 131:25 quotes 76:12 83:15 <hr/> R <hr/> R 2:11 race 33:18 40:6 radio 14:14 raise 67:9 RAM 7:3,19,23 31:10,19 33:8 33:10 41:7,7,17 41:21 73:10 78:14,15 88:3 125:6 132:7,8 ran 18:20 19:11 19:16 65:4 random 67:8 randomization 115:22 randomize 77:17 77:18,25,25 78:9 reach 46:2 reached 39:19,19 reaching 47:11 read 7:14 8:8 9:9 27:11 32:7,12 42:6 54:2,20,24 55:9 63:3 79:18 80:5 93:21 106:9 125:18	128:11 reading 93:23 124:22 reads 32:11 ready 128:6 real 43:10 59:6 realize 98:19 realized 34:13 53:10 97:20 really 9:19 12:11 14:15 15:4 16:22 18:13,20 19:20 22:23 23:18 25:10 27:18 29:18 30:10 38:7,13 38:21 86:24 101:16 123:11 127:7,18 128:19 reason 8:24 13:13 16:25 42:15 53:15 90:24 94:11,16 96:21 96:21 97:5 100:4 101:7 111:22,24 117:20 119:20 129:19 131:7 133:19 134:19 reasonably 129:6 reasons 9:25 25:22 76:3,21 100:7,9 101:2 120:9 rebut 56:7 recall 23:12 45:7 69:22 74:21 80:12 93:16,18 105:14 113:22 receive 10:13 received 6:4 10:10 14:5 32:1 47:18,21 70:4,5 74:14,15 95:19 124:14 receiver 10:3 receiving 32:2 recess 91:24 92:1 recipe 29:3,4,6 124:17 134:12
---	--	--	--	--

recognize 69:16	rely 40:2	135:15	restructured	27:10 30:3
recommended	remaining 70:21	reporters 19:15	128:22	31:15,15,16
96:4 98:7	remember 15:11	REPORTER'S	result 68:17 86:17	32:8,21 33:1,6
record 5:5	15:12 28:2 34:4	1:14	results 16:1 67:1	34:10 35:8,10
recording 5:20,24	41:20 45:8,9	reports 46:1,3,6	68:15,23 90:1	35:11,20 36:17
6:9,11	48:5 67:5 69:9	77:9	resumed 6:17	36:17 37:25
RECROSS 4:3	69:21 93:19	reposted 29:1	92:2,9	38:15 39:11,22
Redirect 4:3	105:15 121:18	reposting 126:11	retain 68:2	41:4,20 42:1,18
126:18,20	133:4	representations	retained 13:24	43:18,25 44:22
refer 58:10,12	remembered	76:22	55:22 68:2	44:23 45:11
73:20 124:8	79:25 80:11	representing	94:23	46:20,22 47:5
reference 40:24	remind 6:10 39:5	124:12	retransmit 10:17	47:13 48:10,23
41:9 46:9,13	remote 10:19	reproduce 15:22	return 85:10	49:1,5,10 50:8
53:3 79:25	128:1	reprogrammed	86:14,18 87:1,5	50:11,15 51:2
126:5,22	rent 25:16	134:11	87:7,15 88:11	51:14,18 54:4
referenced 129:9	rental 25:14	require 40:1	88:13 114:22	56:8,18 57:13
references 63:3	rentals 26:2	62:21	returned 16:9	58:20,21 59:19
111:9	repeat 9:20 125:3	required 92:15	Reuven 47:24	60:5,11,15,19
referencing 47:4	report 4:10,11	99:22 127:18	reveal 87:10	60:21 61:7,24
referred 29:2	20:21 23:3	130:15	revealed 77:17	63:7,15,18,21
80:11	27:12,12,14	reschedule 5:9	reverse 11:14,16	63:23,24 64:7
referring 7:4 77:5	29:22 31:3 35:1	research 22:5	24:13,17,23	64:13,19,20,23
81:5,5	36:19,19 40:14	23:16	27:21 28:11	65:21 66:9,12
regions 113:9	41:10,12,15,23	researcher 60:6	56:13,24 57:21	67:16,19 68:9
register 52:23	42:2,5,6 43:5,7	researchers 13:22	59:13	68:10,21 69:24
87:14	43:11,14,19	24:9 59:4	reverse-engineer	70:7,9,24,25
regular 127:3	44:8 45:6,10,13	reserved 8:20	20:9 21:22 22:9	71:8 72:2,6,15
regulations	45:14,16,18	resource 98:2,21	24:19 26:12	73:2,17 74:19
135:10	46:12,24 48:11	resources 127:6	27:8	74:20,24 75:14
related 9:18 55:10	50:4 55:8 56:9	resource-constr...	reverse-enginee...	76:1,20 77:2,23
55:19	69:3,8,18,20	127:7	11:24 26:2,4	78:17,24 79:1,2
relating 69:5	70:7 73:24	resource-constr...	57:1,19	79:10,13,20
127:14	80:13 101:22,23	126:23	review 41:23 42:5	80:10,13,14,18
relationship 18:5	102:23 103:1,2	respect 21:24	51:9 52:4 53:8	80:20 81:3,13
24:7	103:21 105:4	23:10,17 59:10	55:3 97:24	81:16,17,23
release 17:14	106:1,6,14,18	60:20,23,24	130:25	82:15 83:20
18:10,13 23:13	106:20,22	70:12 119:16	reviewed 52:3,9	84:1,15,19,20
23:17 63:8	107:22 108:3,8	133:1	123:1 133:8	84:24 86:15,23
70:21 71:9,19	109:8 122:12,20	respond 14:18	reviewing 48:3	87:23 88:1,3,4
82:9 129:2	123:3,5,10,13	response 10:21	53:9,10 118:1	89:4,5,10,17,18
released 18:16	123:16,18,23	responsibility	REV3.13 48:24	90:1,7,14,19
24:2 69:7,9	124:5,19,23	101:11	rewritten 129:1	91:16 92:14,15
132:20	125:4,25 126:5	responsible 17:15	RFID 14:14	92:19,20 93:1,4
releases 115:17	131:23,25	19:21,24 23:10	RICHARD 2:16	93:11,24 94:6
releasing 23:7	132:11 133:17	24:24 27:22	ridiculous 130:18	94:10,15,19,25
relevant 104:18	reported 134:22	28:11 48:4 90:5	right 5:5 6:2 7:7	95:3,12,13
relied 7:23 46:23	135:7	90:8,25 124:16	8:12,25 10:19	96:17,19,20
47:11 56:8	reporter 1:21	responsibly 17:15	11:13,18,22	98:2,11,12 99:4
relies 9:3 29:22	18:5,22,25 19:7	71:20	20:23 21:2,4,8	101:25 102:6,16
29:24 36:9	19:13,16 134:21	rest 78:25 79:22	21:12,23 27:5	103:4,6,8,12,15

103:18 104:4,11 104:13,22,25 105:3,18,20,23 106:24 107:11 107:17,20 108:25 109:21 110:13,17 111:1 111:14 112:6,19 113:24 114:3,10 115:4,5,8,14,14 116:1 117:13 118:11,14,18 119:8,18 121:2 121:5,5,7,8 122:16,19,25 123:3,10,21 124:2,13,18 125:8,20 126:1 126:7 128:2 131:2,24 132:12 132:21,24 134:1	row 5:18,22 6:6,8 14:24 RPR 1:21 135:16 Rube 129:20 130:2,3 Rubin 2:23 4:4 6:16,20 11:11 20:6 24:12 28:14,23 40:16 40:17 44:3,13 44:20 46:12 71:5 92:11 100:24 106:13 126:22 129:3 130:24 133:6 Rubin's 4:10 100:18 rule 24:11 134:22 run 12:7 36:24 63:14 73:22,23 76:5 90:3 130:9 130:11 running 48:24 86:2 runs 86:17	99:7 111:25 120:20 132:13 says 11:5 48:24 61:25 62:12 70:18 71:25 73:3 77:3,16 78:8 79:19 82:11 85:10 87:13 88:16 107:10 125:17 132:1 scanned 19:3 scanning 19:2 scenarios 72:16 72:17,23 school 84:3 science 17:18 scope 27:25 55:7 97:13 scratch 43:14 script 82:12 search 14:25 50:19 88:16 seat 19:7 second 5:18,22 6:6,8 7:2 12:12 30:14 52:21,24 53:1 70:16 76:25 92:11 107:5 secret 130:11,12 section 8:19 33:21 35:16 120:3,3 120:17 123:19 125:17 135:5 sections 121:16 sector 57:4,20 secure 12:25 14:9 56:14,23 67:18 68:23 81:10 100:2,5 security 10:11 11:19 20:2 22:17,22 26:18 27:4,9,16,23 41:11 44:4 50:22 56:11 58:20 59:4,10 59:17 60:6 61:17,22 64:8	70:7 73:8,16 78:4 80:1 100:8 109:24 133:9 see 5:22 11:4 15:8 15:14 16:10,22 22:21 35:12 37:2,8 38:6 45:16 46:16 48:1,8 49:6 52:23 57:8 62:4 67:8 79:6 82:2 83:4,9,15 86:18 87:24 97:23 104:18 108:10 110:5,19 111:18 113:13 seeing 5:9 119:11 seen 31:21 50:14 60:2 118:13 119:4,12 133:6 Semiconductor 57:24 send 8:15 10:16 66:25 111:6 112:10 113:6 sending 14:13 32:25 110:11 sends 9:22 31:14 31:15 sense 88:13 sensitive 128:20 sent 9:24 23:24 130:8 sentence 43:22 sequences 103:7 103:12 104:12 104:17 sequentially 78:2 serious 26:3 serve 100:11,25 serves 70:20 services 57:19 session 92:4 set 12:22 13:1,4 14:10 15:2,6 20:8 21:5 53:23 setting 79:22 87:13 129:18 setup 19:1 set-top 8:12,14,18	9:9,22 10:3 128:4 set-up 79:22 share 102:15 shared 102:21 sheet 111:13,17 111:18,21 shell 36:21 37:9 39:15 40:11 42:17,21 63:14 73:19,24,24 74:1 87:2,5,15 88:11,17,17,19 88:21,23,23 90:3 105:4 114:5,23 115:12 shift 37:11 Shiloh 47:25 Shkedy 117:7 133:1,11 shoots 130:1 short 26:1 shortcuts 37:8 shorter 37:9 shorthand 31:16 show 7:24 8:6 9:11 18:1,8,20 21:6 24:9 31:15 34:6 39:13 45:18 46:8 47:1 59:23 69:15 78:20 80:3 82:3 96:25 102:23 126:13 showed 16:13,17 19:11,17 55:16 60:3 67:6 102:2 126:8 132:16 showing 31:11 shows 48:6 79:14 85:4 side 5:18 6:6,7,7 25:8 sides 46:1 side's 46:5 signal 91:10 signature 36:11 signed 75:12 122:17 124:8 significant 7:17
robustness 21:17 21:18 24:1 27:15 ROM 41:1 48:25 50:25 51:15 52:1 53:9,12,16 53:22 54:8,8 55:11,12,17,20 55:25,25,25 78:16 82:23 88:3 91:15 93:12,17 94:13 94:17 95:14,16 96:22 98:22,25 99:13 100:1 101:11 105:9 108:5,15 109:15 109:20 110:18 110:20,24 111:2 111:5 112:1 114:1,6 117:3,8 117:9 118:2 130:21 133:15 134:11 room 1:22 17:9 98:24 Rose 47:24 48:4 Ross 60:10 routine 105:16	S SACV 1:7 Safari 72:11 82:1 safe 87:7 Saggiore 133:15 San 2:13 Santa 1:16,23 5:1 sat 44:24 satellite 1:5 2:2 50:22 91:10 Saturday 97:16 save 131:15 saved 85:10 saw 22:22 73:15 75:18 94:24 96:8,11,11 107:24 108:1,21 109:3 113:12,19 116:12 saying 6:5 10:17 14:22 20:21 45:8 72:10 77:22 79:17 83:23 84:6 99:5			

11:16 42:8 123:22 similar 21:24 29:10 37:3 39:25 62:10 78:15 85:6,8 88:3 92:19 102:2 similarities 40:22 109:17 similarity 31:12 simple 18:17 86:7 128:17 simply 17:25 21:12 66:21 93:21 106:22 109:10 112:23 119:6 simulator 19:5 single 110:4 128:11 sir 26:24 40:19 57:5 65:17 70:17 77:1 84:14 89:21 96:1 100:20 107:5 109:23 124:25 sit 34:21 site 50:13 75:17 97:22 sitting 15:12 situations 16:4 six 18:10 117:6 128:8 size 105:14 skilled 117:3,5 slide 77:16 78:20 79:3,5,14,22 82:21 84:25 87:10 slides 83:9 slowly 31:13 small 61:4 Smart 7:10 8:13 8:16 9:22 31:10 31:14,24,25 32:1,10 50:25 51:12 53:25 60:8 62:1,14,17	127:2,24 128:4 130:5,9,9 Smashing 80:1 snippets 69:8 SNYDER 2:10 society 13:6 software 10:11 21:10 23:19 24:8 36:20 40:24 41:24 51:3,6 56:13,15 57:22 59:17 62:1 65:24 72:7 75:1,25 79:18 82:2 129:7 133:23 134:2,10 somebody 10:8 12:24 16:7 17:25 24:19 37:5 56:22 59:16 90:17 95:8 103:23 106:21 108:18 118:16 somebody's 104:6 someone's 26:19 soon 98:16 100:12 100:16 101:7,12 101:14 sophisticated 8:3 sorry 51:18 75:9 78:19 85:13 131:5 133:4 sounds 82:3 source 44:1 45:14 79:17 94:24 95:5 116:8 118:13 121:12 130:25 so-called 107:12 124:17 SP 87:11 space 127:5 spare 128:9 speak 122:5 speaking 80:5 125:9 specialty 19:19 specific 17:23 59:20 84:21	89:8 103:16 115:20 specifically 26:6 55:24 134:6 speculating 120:9 speed 12:2,3,22 13:10 15:4,19 16:8,14 18:23 20:9,25 22:6 spending 117:6 spent 53:8 61:11 spoke 97:14 spot 35:9 spreadsheet 40:19 spring 130:1 ST 51:15,20 60:13 60:17 101:20 110:24 133:15 stack 11:4 30:3 35:12,13,15,18 36:16 38:25 39:1,3,13,16,20 40:9 41:21 78:15 80:1 85:8 85:11,13 86:3,5 86:5,13,21 87:1 87:5,11,14,16 87:24 88:2,9,12 88:15,24 105:6 105:9,14 115:7 115:24 116:18 staff 23:12 stamp 46:18 47:3 stand 6:17 123:5 123:15 standard 79:21 80:25 Stars 2:17 start 10:1 13:8 23:18 36:6,16 37:14,15 38:2 38:11 39:20,23 40:10 67:7 86:25 started 22:3 53:15 53:19 97:17 113:13,16 starts 41:22 state 6:4	statement 68:1 100:14 111:22 121:8 statements 93:9 States 1:1,22 135:6,10 station 12:6 18:24 19:6 stenographically 135:7 step 18:2 21:9 66:7 73:4,23 steps 18:1 21:17 22:15 23:9,14 80:22 89:16,20 89:22 stipulation 56:20 stocks 66:1 Stone 2:16 4:5 27:24 44:16,17 44:19 45:22 46:8,11 47:14 47:20,22 48:15 48:17,20 49:21 49:23,25 50:3 59:24 60:1 69:25 70:6 74:10,16 79:3,4 83:11,13 84:16 84:18 88:5,7,25 89:3 91:21 92:6 92:8,10 100:17 100:20,22,23 102:14 106:9,12 106:25 107:4 125:16 126:16 126:23 129:9 130:19 132:16 133:7 stop 73:11 77:13 77:14 93:23 125:12 store 7:13 stored 8:9 stories 18:6 story 18:8,19 19:11,16 strain 30:8 Street 1:22 strings 89:8	Strizich 58:1 strong 100:14 structure 30:10 39:10 53:18 102:2,4,7 128:25 student 17:9 students 14:11 15:16 16:23 17:21 19:2 44:9 52:7 studies 25:5 study 13:24 94:22 studying 13:10 style 104:7,8 120:7,8 styles 43:20 44:2 103:3 104:2,5 subject 115:9 submit 17:22 submitted 23:3 41:24 126:4 subroutines 85:16 subsequent 116:6 successful 13:11 13:18 sue 13:18 sufficient 70:23 106:2,7 suggest 133:8 suggested 59:16 suggestion 45:3 45:11,12 Suite 2:6,12,18 superfluous 131:18 132:4 supervise 77:8,9 support 13:16 20:18 supporting 123:13 supposed 9:8,24 17:21 36:14 57:3,6 65:1 sure 7:6 8:5 9:20 11:15,25 19:9 19:24 29:16 30:22 34:24 35:22,23 42:10 48:5 60:9,16
--	--	---	---	--

65:23 71:14 76:8 87:16 101:6 106:17 112:8 116:19 118:4 125:4 127:1 129:18 suspenders 35:25 Swiss 46:14 48:7 50:10 switch 134:21 sworn 6:16 97:4 synchronize 91:6 syntax 52:19 53:4 system 6:23 8:14 10:4 12:15 17:8 23:4 25:15 26:3 26:4 27:4,9,16 27:23 41:11 44:4 50:22 53:17 56:17,25 58:17 64:16 66:4 73:21,22 74:23 76:5 79:15 80:23 82:24 129:12 130:6 133:3,9 133:11,15 systems 19:20 20:4 51:1,4 61:16 62:14 64:6 115:17 S-T-R-I-Z-I-C-H 58:2	33:13 37:12 41:8 57:3,7 62:21,22 69:8 72:18 74:17,22 78:25 81:11 97:9 110:23 talked 22:5,6 41:10 72:19 73:11 75:4 109:2,17 118:22 127:1 talking 24:18,18 59:20 77:18 92:25 talks 42:11 target 40:5 Tarnovsky 133:14 tasked 95:2 taught 52:7 53:5 67:6 teach 52:17 80:4 team 20:8,20 21:10,21 22:8 24:25 technical 14:15 14:21 15:20 17:17,18,22 20:1 31:13 42:3 42:8 43:15 70:19,21 Technician 48:19 technique 84:7 86:20 techniques 59:13 60:12 61:25 74:18 technological 96:21 97:5 technologies 14:16 technologists 20:18 technology 22:12 26:12 27:17 Ted 47:24 television 50:23 tell 5:7 11:23 16:20 22:14 27:1,2 30:5	33:12 35:16 76:25 91:19 95:8 97:18 113:14 118:19 telling 23:4,6 85:13 tells 39:5 term 14:21 20:3 24:12,15,16 58:10,12,13 59:16 67:23 70:12 129:20,21 terminates 105:21 terminating 99:20 terms 30:19 test 14:13 112:23 113:2,4,6 tested 16:15 testified 26:24 27:6 40:18 41:16 45:1 51:24 54:8 89:15,20 93:5,8 94:5,11,16 95:10 98:21 101:22 102:1,5 108:21 113:25 117:2,11 118:12 118:25 129:6 133:14 134:17 testify 27:1 58:22 100:24 101:2,6 106:1,6 111:7 112:23 114:4 117:22 119:23 133:23 testifying 97:10 testimony 21:20 29:3 33:7 43:21 49:15 54:20,24 75:12 90:4,8,16 90:20 93:25 94:17 97:1,4,18 97:23 99:8 108:11,17 118:15 119:5 121:8,11 126:22 129:5 130:25	131:8 132:25 133:20 134:19 Texas 2:7 13:12 13:13 16:12 text 48:21 thank 5:24 6:2,11 6:14 11:11 20:6 24:4 28:14 32:4 40:16 44:13,17 47:20 92:8 100:22 125:14 126:16,19 129:3 theory 35:24 122:1,2 They'd 89:18 thin 34:1 thing 8:12,17 10:18,24,25 12:14 13:2 14:13 15:7 30:8 32:12 35:3 37:7 37:10 42:21 64:17,18 70:10 70:11 73:12 85:17 104:6 127:22 128:24 130:2,10 things 10:16,21 15:2 17:20 30:11 32:10 34:14 35:16 38:2 40:13 42:10 43:7,7 52:20 53:7 57:17 67:8 82:2 82:3 85:20 86:6 86:13 87:3 110:15 115:14 115:19 116:13 116:13,17,22 120:20 127:6 think 9:17 11:13 12:23 13:15,21 16:25 25:3 43:12 45:4 53:11,19 54:15 59:1,5,5,8 61:13 61:13 64:16 65:2,24 69:9,20 76:24 80:16	85:16 87:20 91:2 94:20 97:16 100:15 101:14 106:8 109:25 110:16 114:5 116:3,9 119:1 120:21 122:1 126:15 129:8,10,14 130:19 132:12 132:13,15 thinking 25:19 119:6,10 third 8:2 56:19,21 107:6 third-party 56:16 Thomson 101:20 thorough 123:2 thought 13:17 20:11 22:21 25:10 30:6 46:5 50:1 51:17 63:17 64:14 75:3,15 108:4 127:18 thousand 12:20 three 7:1 14:10 63:8 69:9,10 114:17 127:21 Thunder 54:21 55:4 ticks 127:9,11 tiger 19:9 tight 128:7 till 15:17 time 5:23 8:22 15:13 19:14 25:18 28:7 45:21 52:3,9 53:12 58:14 59:8 68:18 70:1 70:23 74:10 75:15 78:5 80:13 86:2,6 89:12 91:19 94:22 96:23 97:6,14 111:21 118:12 121:6 126:17 127:9,10 128:11 131:15
T				
T 2:3 table 40:21 take 5:16 13:14 13:23 19:22 21:16,17 32:2 60:2 65:5,8 67:7 70:24 73:13 77:24 91:24 93:8 109:17 116:19 122:24 taken 76:9 109:25 takes 12:12 86:3,4 127:21,22 talk 7:1 11:13 12:1 16:7 25:5				

133:2,10 times 18:5,9,19 31:6 44:22 128:14,14 timing 69:9 tinkerer 57:16 title 81:25 135:6 titled 74:18 today 94:20 102:5 told 9:21 13:14 16:11 21:19 22:4 25:12 26:8 31:3 68:16 69:13 75:5,10 75:16 81:7 86:8 126:10 tongue-in-cheek 84:5 tool 65:4,7 67:20 tools 64:24,25 74:18 top 8:12 11:3 35:18 36:15 39:16,20 40:3,9 41:21 48:18 61:16 73:3 76:15 82:1 125:10 129:25 total 81:18 totally 119:14 touched 114:20 track 127:25 trajectory 85:19 transcript 1:14 135:7,9 transmit 105:17 transmitted 12:13 12:13 transpired 18:19 Transportation 5:9 treat 83:3 trial 1:15 5:20,24 6:9,11 20:3 34:20 44:24 97:17 trial-and-error 67:3 trick 35:4 tried 13:2 64:19	123:1 130:19 tries 42:3 triggered 53:21 trivial 130:7 trouble 37:1 true 5:21 71:5,11 84:14 95:21,23 99:11,15,25 104:23 118:20 130:24 135:6 truly 26:21 try 7:12,13 21:16 21:17 38:9 43:2 64:16,17 80:6 trying 9:20 35:8 68:19 78:1 114:25 115:1 tune 13:21 turn 15:8 28:15 44:10 turned 119:16 121:23 turns 28:4 34:25 TV 8:13 tweezers 34:22 twice 30:12 35:21 35:25 94:18 97:10 128:24 two 30:9 31:9,12 40:13 42:4 43:10,22,25 44:9,10 52:23 71:1 73:7 76:12 93:8 94:8 102:10,10,12 103:17 105:1 118:2,6 119:24 121:16 123:15 124:6 125:9 type 61:4 types 43:24 62:18	underneath 72:16 understand 9:12 11:7 14:9,17 21:20 30:19 36:22 43:19,21 45:17 49:12 53:3 56:3 84:2 98:2 109:1 111:25 128:21 129:14 131:20 understanding 29:25 46:4 50:17,20 52:16 52:19 66:7 understood 15:3 20:7 48:25 97:21 98:20 124:15 132:25 undertake 25:11 undertaken 26:14 undertaking 19:23 25:11 undertook 56:3 undocumented 124:20 125:21 United 1:1,22 135:6,10 university 11:21 13:9 20:20 60:7 unknown 29:22 35:8 56:25 unlimited 25:17 unlock 63:12 90:2 unpatched 109:22 unusual 7:16 31:21 119:1 127:19 unwritten 24:11 upgraded 82:9 upset 81:10 up-and-coming 14:16 usage 8:3 use 10:6 14:9 18:17 24:3 30:22 32:5 38:15 41:17 49:7 64:25 65:13,15 66:18 67:20 70:12	75:20 81:15 88:13 89:16,19 89:22 103:24 112:4 114:11,12 114:12 116:25 useful 83:23 USENIX 20:2,3 61:6,10,14,19 61:20,21 USENIX's 62:21 uses 42:25 43:2 67:18 104:24 105:16 115:6 usually 127:22 utilized 106:5 116:2 utilizing 112:17 112:19 U.S 26:3 135:15	verbs 43:23 verification 132:4 verify 113:5 119:20 version 48:25 76:4 89:12 versions 82:8 versus 24:20 57:16 view 130:3 violation 25:24 28:4 visiting 50:13 voice 28:6 121:16 Volume 1:8 5:2 134:23 voting 14:3 18:6 vs 1:7 vu 29:18 vulnerabilities 21:7 22:12 23:1 24:9,10 66:16 70:24 72:3,8,9 73:11 100:11,12 100:13,16 101:1 101:1,3,7 110:14 115:3 116:7 117:9 vulnerability 6:25 7:21 20:25 21:11 23:3 64:3 64:4 66:22 67:16 71:25 72:6,10,18 73:3 73:9,23 79:8,11 79:14 80:22 81:12 84:10 90:13 94:14 96:5,9,11,14,16 98:8,16 100:6 101:12 109:19 110:10 116:19 116:22 117:15 125:5 134:8,8 vulnerable 73:12 82:9 95:20
	<hr/> U <hr/> Um 76:20 unbeknownst 75:25 unbelievable 118:16 unchecked 82:13 uncommon 10:10		<hr/> V <hr/> valuable 34:14 value 8:7,7,24 31:22 32:4 34:4 35:23 40:7 52:24 86:24 87:14,17 values 17:5 31:4 34:7,18,25 35:1 38:3,13,14,16 38:17,18 52:24 113:14 variable 8:4 9:4,7 9:13 29:24,24 30:1 31:22 32:3 32:4,6,13 33:18 33:19,20 34:5 35:5 36:10 37:19 38:10,19 39:5 40:5,6 41:8 41:10,13 78:2 85:6 114:11,13 114:19,21 117:1 125:6,7 130:17 variables 86:22 various 61:21 vehicles 12:20 vendor 101:10 vendors 100:12 100:25 101:6	<hr/> W <hr/> W 2:10 WADE 2:3,5

waited 15:17	wealthy 20:17	wished 80:8	write 17:17,24	90:23 108:4
walk 7:24 9:18	web 15:11 22:25	witness 6:16	18:8 33:14,25	114:7
18:22 19:3 31:8	67:7	19:13 28:10	38:5,12 39:6,9	year 61:20
walking 17:7	webcam 15:7,8,14	31:2 44:14 50:1	39:12 73:21	years 12:1 19:20
want 7:1 10:1	website 29:12	125:11,13	76:6 81:15	43:10 44:6 52:7
12:9 16:20,21	37:12 54:18	WITNESSES 4:3	84:22 85:18,24	61:11
16:22 18:4,15	67:5 71:9	woke 15:16	86:21,24 93:16	yesterday 27:13
28:15 31:8	Wednesday 1:17	won 20:1	93:17,20	33:7 49:15
36:24 38:24	5:1	wonder 12:24	writes 42:22	97:20,23 121:9
45:17 56:22	week 22:10 54:13	wondered 35:21	writing 38:2,11	127:1 129:6
77:14 79:24	54:13 115:17	word 79:23	39:8 42:23,24	131:8 133:13,23
81:3 86:9 87:14	weeks 22:19 63:8	words 43:23	43:13 44:1	134:17
128:1	69:9,10 71:1	work 11:9,20	103:23 104:8	yesterday's 97:1
wanted 5:12	WELCH 2:3,5	14:10 15:2	132:14	Yoni 47:24
35:11,17,23	well-known 60:6	16:24 24:24	written 9:2,13	York 18:5,9,19
38:23 71:15	110:14	29:5 30:15	17:10 20:21	
76:8 115:13	went 15:14,17,18	33:20 34:17	30:12 33:6,16	Z
128:17 130:14	19:4 21:9 30:20	35:2 36:1 43:10	35:4 36:21 37:9	zero 32:17,17,22
wanting 76:5	53:22 62:16	46:10 53:2 55:7	38:4,18,19 39:3	32:22 33:15,25
wants 9:10 38:24	69:10 75:4,17	56:20 57:3	39:11,15 43:11	33:25 34:6,6
56:22 88:20	weren't 97:5	60:13 63:4	51:3,6,20,22	38:6
wasn't 18:12	106:4 117:20	65:13,14 71:18	65:11 88:19	zeros 31:17 33:21
22:23 27:18	120:10	77:8 80:3 92:21	120:7,8,9,11,11	33:23 34:9
35:22 44:12	West 1:22 2:12	93:2 101:17	121:16 131:13	zoom 48:17
52:2 80:6 95:2	we'll 11:4,5,11	110:5	wrong 53:10,21	Zvi 117:7
100:4 117:18	13:16 24:10	worked 8:1 14:2,4	123:9,11	
128:17	71:21 91:24	18:6,16 26:23	wrote 13:25 16:1	S
wasted 37:17	we're 5:5 13:11	34:13 41:11	30:14 43:22	\$10 13:21
water 25:8 130:1	13:18 15:9	52:18 89:14	69:3,4,6 71:3	o
way 7:21 8:13	16:19 17:7,7,13	91:18	74:1 78:12	0X 31:4
10:13 11:2	17:14,25 18:14	working 5:9 9:12	104:23 105:19	0X1DD 132:9
14:22 18:4	25:19 26:9,17	14:11 52:6,7	119:24 120:3,13	0X19C 132:9
21:11 24:16	31:18 34:3 35:8	works 58:6 60:7	120:14 122:8,20	0X20 132:9
29:21 36:7 38:8	38:12 39:8,20	73:7 76:7 86:1	123:22 124:6	0X9A 132:9
42:19 44:12	71:23 72:23	workshop 61:6	132:11	0-day 79:6,8,11
52:21 58:12	84:7 92:4	workshops 61:20		82:1
66:20 67:12	we've 11:25 16:7	61:22	X	00 36:7
73:13 79:14,21	21:13 31:6 38:8	world 43:10	X 4:1 80:23 86:23	0001 31:16 38:12
80:25 81:22	38:9 44:22	worried 10:7	86:23	0010 31:16
86:1 88:8 104:5	56:21 114:17	worry 32:15	xbr21 101:3,8	0060 39:12
108:16 109:10	whack 76:9	wouldn't 14:6	106:2,7,14	019C 35:17 36:2
109:15 110:2	whatsoever 33:23	25:14 33:22	107:18 108:7,13	36:12
113:7,11 114:1	white 57:14 82:23	89:19,22 99:23	108:22 109:2,13	03-950 1:7
114:4,16 116:11	widely 62:13	112:25 113:15	124:7	04 37:22
127:15 128:23	wife 22:4	113:19 119:9		1
133:9,15	WiFi 72:20,21	130:15	Y	1 37:16 38:6
ways 37:4 99:22	wild 129:22	wrap 9:6 29:23	Y 86:23	39:10 125:2
103:22 113:9	WILLETTS 2:4	38:2 131:11	Yeah 22:16 52:14	1st 50:4
weak 110:1	wireless 72:20	wrapping 7:17,24	56:15 61:11	
weakest 109:24	wires 66:2	wraps 40:3	67:23 68:13	

1-053 1:22	94:23 122:17	110:20,24 111:5	74 4:12
1:01 5:3	2008 1:17 5:1	112:1 114:1,6	753 135:5
10 74:19 79:5	52:5 94:23 97:4	118:2 130:21	77057 2:7
115 114:19	114:9 135:12	134:11	785-4600 2:19
12 61:20 100:18	2008-04-16 1:25	3:00 92:1	799 45:18 102:23
100:20 106:10	21 31:15 32:8	3:18 92:2	122:12
126 4:4	106:10	30 25:16	799-19 45:20
1400 2:18	21st 28:25 126:6,8	310 2:19	799-32 102:24
16 1:17 5:1 14:24	126:11,13	32 33:13 102:24	799-36 122:15
17 135:12	22 78:20	33 84:25	
17th 70:19	225 4:10 47:1,15	34 88:5,16	8
172 100:18	47:21	36 122:14	8 76:11,14
1760 46:21	225-003 49:2	37 122:14	800 4:11 69:15,16
18 100:18,20	225-01 47:7,23	38 88:25 102:24	69:25 70:4,5,17
124:5	225-02 47:2 48:13	123:18 124:5	71:23
19 46:12	48:15		800-5 71:22
1989 67:21	225-03 47:3	4	800-6 72:25
1990's 64:11	23 78:18	4 37:17 38:6	802 4:12 74:3,3,10
1998 45:4 46:15	23rd 29:1 123:25	39:10	74:15 76:11
47:25 50:5,7,9	124:8,16,20	4th 1:22	78:19 82:17
50:14	125:5,18 126:10	411 1:22	89:1
1999 2:17 61:19	24th 46:14 123:25	415 2:13	802-20 77:1
62:6 132:19	2401 2:6	44 4:5	802-22 78:22
133:15	25th 47:25 49:4	47 4:10	802-25 81:1,11
	50:9,13		802-26 81:24
2	255 132:8	5	802-30 83:11
2 37:16 38:6	26 94:23	5 1:8 5:2 34:10,11	802-31 84:16
39:10 53:9,12	26th 52:5	37:17 38:6,15	802-8 76:18
53:16 54:8	2600 2:12	39:10 71:25	809 59:23 132:16
55:17,25 70:17	27 82:16	511A 107:1,5	132:19
98:22 111:2	27th 50:7 97:4	55 39:18	81 34:10,11,23
123:18	114:9	558-8141 1:23	38:15 107:12,21
2nd 70:22 74:5	275 2:12		107:25 108:2,6
2:00 15:15	28 52:5 135:6	6	108:14,19 109:6
20 15:24 33:13	29th 122:17	6 4:4 37:17	
38:5 76:24,25		124:23,24 125:1	9
91:25	3	6505 51:16	9B 39:9
20/20 18:21 19:11	3 37:16 38:6	6805 51:17,18,22	9C 39:9
19:16	39:10 41:1 48:6	52:10	9D 36:7
2000 44:5 55:13	48:25 50:25		90067 2:18
55:18 56:1	51:15 52:1	7	92701 1:23
61:13 91:14	53:22 54:8	7 37:17 76:16,18	94111 2:13
92:12 96:10,18	55:12,20,25	106:10	9472 1:21 135:16
98:1 99:3,6,9	78:16 88:3	70 4:11	952-4334 2:7
123:25 124:8,20	91:15 93:12,17	700 2:6	98 45:5
125:5,18 126:6	94:13,17 95:14	713 2:7	984-8700 2:13
126:8,13	95:16 96:22	714 1:23	998 28:15
2001 133:25	98:25 99:13	73 107:12,21,25	
134:12	100:1 101:11	108:2,6,14,19	
2007 74:5 80:15	105:9 109:20	109:6	