An Interview with

TERESA F. LUNT

OH 426

Conducted by Jeffrey R. Yost

on

4 June 2013

Computer Security History Project

Palo Alto, California

Charles Babbage Institute
Center for the History of Information Technology
University of Minnesota, Minneapolis

Teresa F. Lunt Interview

4 June 2013

Oral History 426

Abstract

This interview with computer security pioneer Teresa Lunt discusses her work in the computer security field at MITRE Corporation, SRI International, DARPA, and PARC. At SRI she was a principal research scientist on IDES, the first meaningful intrusion detection expert system. Lunt also discusses her work at SRI pioneering a highly secure database system (research initially launched by Dorothy Denning), as well as her work at DARPA's Information Technology Office where she started programs to fund research in computer and network security. The interview concludes with her discussion of security work at PARC, where she is the Director of the Computer Science Laboratory.

This material is based upon work supported by the National Science Foundation under Grant No. 1116862, "Building an Infrastructure for Computer Security History."

Yost: My name is Jeffrey Yost from the Charles Babbage Institute at the University of Minnesota and I'm here today in Palo Alto, California, at PARC with Teresa Lunt. I'll begin with just some basic biographical questions. Can you tell me when and where you were born?

Lunt: In Pittsburgh, Pennsylvania, 1954.

Yost: And did you grow up in Pittsburgh as well?

Lunt: I grew up there until I was about 12 or something. Then we moved to New Jersey.

Yost: Who were your greatest influences in your childhood and adolescence?

Lunt: My parents. [Laughs.] My dad was an engineer. He was always; he was working in some research lab for Westinghouse and a couple other companies over time, but he was always bringing stuff home and telling us about stuff. I remember when computers were brand new, he somehow met somebody who gave him this commercial brochure from Burroughs, or one of those companies, and he brought it home for me. It was about punch cards and the little holes and how they encoded stuff. I just thought this was really fascinating. He took me into the laboratory. He was a metallurgist and he showed me how they had these machines for testing metals and they had this one where you could put a metal rod that was maybe two inches in diameter and a foot long. The machine would grab the two ends and then just pull it apart like taffy until it broke. Just like a

piece of taffy. It was pretty incredible. He showed me electron microscopes and all this stuff, so I got pretty hooked on science at an early age. My mom's influence was more in the arts direction, though my dad was an amateur musician as well. So I would say my early influence was mostly from them.

Yost: Did you take a lot of math and science courses in high school?

Lunt: Yes, I loved learning about that stuff and I even got into; I would even just like beg them to send me to these summer courses they were having in math and science. Math in particular I really enjoyed as a kid.

Yost: And you attended Princeton University?

Lunt: Yes.

Yost: Did you know what you wanted to major in, early on in college?

Lunt: No, because I had a hard time because I was really interested in music. I was studying violin and piano at the time; and also really interested in the arts, and also literature and language. I had been studying French and German in high school. I had a really hard time figuring out; 'course I loved science but you know, just the choice of going to Princeton over a place like, say Julliard or something, was a choice not to be a professional musician; not to try to pursue that. When I got to Princeton, I had gotten a

scholarship from the state of New Jersey to pay for my college. It was a competition they had, and so I had a four-year scholarship [that] paid for almost everything. And it was from the Department of Environmental Protection so I sort of thought that I should do something that would be relevant to that but there was not very much at the school. [Laughs.] So I went into the geology department, which was the closest thing, although I don't really have a head for rocks, soils and all. But I did find there's a whole section of geology, which is geophysics, which is all basically physics and math and I just loved this stuff, so that was great. Computing was just getting off the ground at Princeton. There were some big time sharing machines at their computer center, which you know, you'd have to travel there, go ride by bike, and they'd have a little light; I remember they had a little light on the outside so you didn't waste your time going in if the computer was down. [Laughs.] They had a red, green, yellow light. You'd show up with your heavy decks of cards and everything. Then there was this huge breakthrough for me when the department—because I had to do a thesis my junior year and my senior year thesis—when they got a minicomputer so I could start digitizing the traces of earthquakes, which up to then had been stored on microfiche. They were like photographic images on microfiche. And so I started tracing and digitizing those images for this particular historical event that I was studying, and so I could then manipulate the data. That time, I'm not sure how much storage was on one of these gigantic disk packs that were like three feet across, but it was very clumsy. And then time sharing started happening. I mean, not just time sharing but terminals where you could start; like the precursors of word processing started happening and so there were these homegrown programs that people were using on these terminals that started showing up. You could start writing

your thesis on these things and that was huge. Because, you know, typing and retyping is just this horrible pain. So it was all starting to happen then. Because I had this scholarship from the state of New Jersey, I had to work there for the summers for this program. I found out, well they had this same computer at the DOT that we had at school, right? So I could program it. So I was able to get a time sharing terminal installed and was programming these water quality models in there and taking data that we collected in the field, and then predicting as it goes through these processes through the water, whether it would become suitable for drinking water further downstream. So we started doing all kinds of stuff remotely there. That's where I really got the computer skills I needed to get into my first real job. And that's where I got interested in security, when I went to MITRE.

Yost: In using that time sharing system at the computing center at Princeton, did they have any security on that system?

Lunt: I don't think so. I mean, that was just not even a consideration.

Yost: I know that Dartmouth, which had a pioneering time-sharing that really focused on student use more than research, unlike the Multics system, the Dartmouth Time-Sharing System had virtually no security.

Lunt: Yes. I just don't think it was a consideration because there was no way; you didn't get direct access to these things. You had to give them a deck of cards. At least on the

IBM 360, you had to give them a deck of cards. There was an operator and the operator was the one who was giving you access. We had accounts and you could get we called it funny money. You had like 50 dollars of funny money in your account and when you used it up, you'd have to go ask for more. You just couldn't, you know; you didn't have your hands on the machine. At least I didn't know anyone who was trying to figure out if I insert these little instructions in there it will somehow break the model. I don't think there was a security model to break so none of this was in my mind. It was probably possible to do all kinds of things but it was just not something that occurred to me to do, or anybody I knew.

Yost: Did you teach yourself to program or did you have instruction?

Lunt: Yes, I did. I had to teach myself. I mean, I had to do it for my thesis. In my junior thesis, first I had to program this machine. Then I got married early. My first husband was doing this little business on the side where he was buying and selling computer time. He would buy time from these time sharing systems — they had all the same computers, they were all IBM 360s and 370s — so the 370s were the time sharing computer. And he would buy time and he would add software to the mix and then he would resell it. So he was reselling it to some company that was doing all this water quality stuff and he wanted me to write all these programs for him. And so I had to learn the assembly language for that, which turns out also was really useful for this job I had with DEP because then you know you can get so much access to the machine with programming assembly language. So I learned assembly language and FORTRAN. And then he got this silent 700; you've

probably seen those suitcases; that was really cool because you could take it home, you didn't have to go all the way into school to use the computer. So that was pretty cool, I would use that and then he was trying also to build up all these other aspects of the business like lists of sales prospects and things like that. So I started learning how do you use; there were no databases, really, but how do you do some recordkeeping on the computer? It turns out then that the state of New Jersey, where I had this summer job, also they had this — I forget what it was called; some primitive database system; it was not a relational database because that hadn't been invented yet — but it was this convoluted thing. And there was this guy there I found who actually knew how to use it. So I was learning from him how to do that, and we would store this data, quality, water sample kinds of information in there. So I was just kind of learning as I went. It was fun. It gets tedious after a while, programming in assembler, but when you're learning how to do it and experimenting, it's kind of fun.

Yost: What was your actual major?

Lunt: My undergraduate, I got a degree in geology and geophysics from Princeton; and then I got a master's later in math, applied math.

Yost: Did you go straight to grad school?

Lunt: No. I had agreed to work for the state for three years, so I did that and then I left to go to grad school in the middle of that. Then they made me come back; basically

threatened to make me pay all the money back. [Laughs.] There was a small group of us who had these scholarships they gave out. For a few years, they were giving out four or five a year, so there was like 10 or 12 of us who knew each other. They all tried to get away from working for DEP for three years and, you know, many of them became doctors and stuff like that; they went to medical school so some of them actually paid back the money but I went back. I spent two years getting a master's degree and then I went back and worked for two more years for the state.

Yost: And did you do a master's thesis?

Lunt: No, I didn't. No. It was math. I don't know what you'd do in math for a thesis. [Laughs.]

Yost: You led an effort to create a database, to computerize the major regulatory [pause]?

Lunt: There were a lot of new things happening at the time, so I was working for a while in the Bureau of Water Pollution Control, so this was after I'd finished school. They were experimenting at the time with this then-new idea of issuing permits to companies that were discharging into their major rivers and streams in the state, and giving them limits on what they could put in on these different substances. And then, of course, they would fine them if they went over the limits. That much was pretty standard and there were people going out to test what they were putting into the rivers. But then we started using

these models that, could we charge for the permit based on the amount of pollution that's going to end up interfering with downstream uses of the water, right? So it was pretty typical that you'd have these major rivers and big industries pouring stuff into the water. Not only big industries, but sewage treatment plants and stuff like that. And then downstream many miles there'd be an intake for water treatment for drinking water or other industrial water. So you really need those natural processes to be able to clean the water, and if it can't then there's costs to those downstream water plants to clean it. So you want the costs to be borne by the people putting the stuff in so we started trying to figure that all out. I started implementing this on the computer, like how much money would you be able to get under these different schemes for permitting charges and would they pay for the actual costs of cleaning the water that they were putting stuff into. So it got a little bit more complex and I started learning about the regulatory side of the whole thing. It was pretty interesting.

Yost: If companies were putting in similar pollutants, how could that be correctly identified and apportioned?

Lunt: Those models would chunk the stream or river into segments that may be a quarter mile long or something like that. So you figure who's got these discharges in there and if they're all in the same segment, well you just combine them and you figure relative to one another how much you measured that they're putting in, right? So it's not that difficult.

Yost: So it's in 1981 that you took a position at MITRE on the technical staff?

Lunt: Yes. I think they were interested in me because of that experience I had in programming these databases. I wouldn't have hired me on that basis, but they did. [Laughs.] I was never trained in any of that stuff, so I just had my own personal experience using these systems and that was my background. But that was enough for them to hire me at MITRE. So then I was at MITRE, I went up to MITRE Bedford.

Yost: Did you know anything about MITRE? Were you responding to an ad, or . . . ?

Lunt: No, I was wanting to move to Boston so I was looking around; what's there to do in Boston. I still didn't know what I really wanted to do; I mean, I was just looking for something interesting, right? It's different for kids today; they have their whole career planned out. Back then, I was just like, I'll do what I love doing and there'll be something I could figure out. So I was looking for something in Boston and I'm not sure how I found MITRE Bedford, but they must have advertised somewhere. And so I sent them my resume and I got an offer of an interview. I went in and was with this group. I didn't know anything about MITRE, so I ended up working with this group, systems engineering. I don't know how much you know about MITRE, but a large part of their business is overseeing these big systems engineering projects. So this group was part of that. They were overseeing this huge thing that was being built out in Colorado that companies like Martin Marietta were building. And we had the really boring job of reading all their documents and seeing if they're meeting the requirements, and flying out

there for meetings, you know, program review meetings, and everything. This was my first real exposure to big computing systems, where there's not just my little piece that I know something about, but all these other pieces; communications and stuff like that, which I really hadn't been exposed to before. So it was a big learning opportunity but on the other hand, we didn't get to see any of the interesting development or science part of it. We just got to see the administrative side, figuring out is this contractor doing what they're supposed to do. I wasn't thrilled with what I was doing there, but MITRE's a big place and like many companies — even here — I think people meet people by going to the gym. So I met a whole bunch of people by going to the gym. [Laughs.] I met another woman. I had been going to the gym and they had a sauna there, which was kind of nice; so I had plenty of time in the sauna, you know, after you work out to talk to these other women. So she was telling me about this thing that she was doing and it was very mathematical. I really wanted to do something that was mathematical, not just some computer thing but it had to be technical. They were proving properties of programs so they could prove that they were secure. So I listened to this for a while, then she offered to introduce me to her boss and so I ended up moving over to this group at MITRE that was doing computer security.

Yost: Do you recall who led that group?

Lunt: I knew you were going to ask that. I will not remember the guy's name. I'm trying to remember who all; so I had an office mate named Marcia Olson, and she's actually; I'm still in touch with her after all this time because I lost touch with her and then it turns

out that she worked for Xerox; she was in New Hampshire. She was working for Xerox and had a meeting at PARC. And at the time I was having a; I'm an amateur photographer so I was having an art show. You notice our walls are; there's no art on them now because we're in between art shows. But every two months we put up another collection, usually from some local artist. So I had my photographs there and with my name on them. She was walking down the hall and saw my name, figured out I was at PARC, and arranged to meet me. And so, you know, that was almost 10 years ago now and we've been in touch again, so that was kind of nice. Who else? That was so long ago. I can probably dredge up the names and I can send them to you if you're interested.

Yost: Did you have any interaction with Steve Lipner?

Lunt: I did, but that was through; that was probably after I came to California and I started working for Dorothy. She introduced me to a whole bunch of people, including all those folks who would go to the Oakland conference. That's really; the Oakland conference was great because it was a small community back then, everybody was there, and you could meet everybody. And because I knew Dorothy, and Dorothy knew everybody, I got introduced and that was great. So back in MITRE, it was; I would say even after I started working in that computer security stuff, there wasn't anything there that was really grabbing me. What happened was a new office was recruiting in Germany and some of the people who had been working in the security group had gone there and so I went there, too. There were two people I knew who had been in this group at MITRE and they were working on intelligence stuff that was really highly classified; they

couldn't tell me about. But I was joining this other group that was co-located with them and I worked there for about a year and a half. The job was horrible. I was mostly there just to vacation in Europe. [Laughs.] But a person who I worked for at MITRE Bedford moved to California. His name was Bill Wilson and he hired me for a computer security job out here, so that's when I started really doing it in earnest. I was at a small company called Sytek. Tom Berson was a VP there. I don't know if you've talked to Tom.

Yost: No I haven't.

Lunt: He's a local here in Palo Alto. It was a local area networking company. Local area networks was a big new thing back then. In fact, at MITRE I had been exposed to it; they had the MITRENet and so when I was working with Marcia, she wasn't doing security. And then we got this other woman in our — because they were hiring like crazy — we got a third woman in our office who was working on MITRENet and other networks. I remember we had this time sharing thing for email. I forget what program; I think it was called MH mail. You could completely fake an email and there was no security. I discovered that I could fake a "From:" line in there and people would just believe it was from that person. [Laughs.] We played a lot of pranks. That was interesting but then I came out here; when I started working for this company, Sytek, whose main business was local area networks, they had this sideline that Tom Berson was working on for secure local area network product, a little box you could connect to your terminal. It was not even a PC or anything back then it was a dumb terminal to some minicomputer like a VAX or something. But they were doing some government contracting as well, in

security. So that's when I started working on contracts for Rome Labs. It was then called RADC and I started meeting some of those folks and I did my first intrusion detection project there. So I knew about the Jim Anderson work; and I knew there was a little bit of work starting at SRI. We had this little experiment where we built this little program to collect information on other people in the office and some people really objected to it, they didn't want people spying on them. But we started to see are there were patterns, and what kind of patterns are there in people's behavior. Then I met Dorothy because she was also getting money from Rome and they were having meetings two or three times a year with all their contractors. She was also working on intrusion detection and so she hired me and I started working with her on the intrusion detection project there and also on a database security project she had going. We also had work from the FBI on intrusion detection. I think it was the first actual place to experiment with it in practice, so we built a little system for them and it ran on this gigantic machine — which is probably no more powerful than this phone now — that we installed in FBI headquarters in Washington. And they were analyzing data that came into their system; they had a database system in the basement that was some brand name that no longer exists, like ADABAS [Adaptable DAta BAse System] or something like that. So we were taking records as they came into the system, the audit records, and processing them through the intrusion detection system looking for patterns. And they were looking for basically were people selling information? Were there things in the data to suggest people were selling information? This is a big problem for them because they have informants on the street and if their identities become known they can be killed, and they would never really tell us about what they found other than that they found it useful and they found patterns, and they

found evidence of bad behavior there. We were able to profile different jobs, different kinds of jobs people had within the FBI, and also the kind of scope; they specialized in different kinds of crime, right? Also, the offices are regional, they're geographically located in different parts of the country. So you can start looking at what kind of different patterns you'd expect in Oklahoma for people in this certain specialty, and are they starting to access data that sort of deviates from that. And those are the kind of things that we were looking at. Also, some of them had very particular personal habits like when they show up in the morning, when they go home at night, and when they take their breaks. You kind of see patterns and that sort of led to the work we were doing for the DoD, which was about are there patterns that indicate whether it's you or someone who has broken in because by then, the Morris worm was happening and all that stuff. So there was suddenly more of an emphasis on people attacking you remotely rather than bad apples in your organization. So we got some money from the Navy; a big contract; a series of contracts from the Navy; SPAWAR, which was their Washington office.

Yost: Before we get into IDES at the SPAWAR, just so I understand, you came out here leaving MITRE for Sytek?

Lunt: Yes, out here. My former boss at MITRE Bedford left MITRE and came here, for Sytek; he was helping to build up this little security group.

Yost: And Tom Berson, was he was one of the founders of Sytek?

Lunt: He was one of the founders of Sytek. There were, I think, four or five of them. They had been working for Steve Walker on contract, I believe.

Yost: Interesting. I interviewed Steve.

Lunt: And they decided that they were going to start this company. I'm not sure how it all got started and Tom would be a great person to ask. That company was later bought by somebody like Hughes and I'm not sure the technology has survived in any form but it was an interesting time because all this stuff was so new and they were some of the first people to offer security products.

Yost: Moving back to MITRE for a moment, were you aware of the work that Bell and LaPadula were doing?

Lunt: Oh yes. Everybody was. It was all there was really to know back then and so everybody of course knew all that stuff. Don't read up; don't write down; all that information flow stuff. We were talking about how does information flow through a computer system. That's what all the formal proofs were about, to show that, if all this data is labeled with its security level, was there any way that some user connected through this time sharing system could find out stuff that was higher than their security clearance. And there were direct ways, of course, which you could block. Then there were the indirect ways and yes, we knew all about covert channels and timing channels. Yes, everybody knew; it was like Security 101. The whole world knew that.

Yost: At what point did you read the 1980 Jim Anderson article on intrusion detection? What did you learn from it?

Lunt: I think that must have been when I was at Sytek; or it could be when I joined SRI. But it was right around that time; like early 1980s. He had some really preliminary results and one of the things that he did was categorize motives for the intruder, which is probably still valid today, you know, the motives he came up with. I can't remember if he had political in there, which is maybe a new one in addition to organized crime today, because, you know, networking wasn't well established in those days. But still, he recognized people were doing it just for fun, like when I was forging emails in those early days at MITRE and pretending they were from God and stuff like that. Some people are going to do it for fun, some people are going to do it for profit, some people are going to do it for revenge, these various motives, right? And that was a really useful thing, and he also had some very preliminary results indicating that patterns were there to be exploited. So if people behave in these patterns then you can use that to decide whether they were doing something good or something bad by categorizing and classifying them in some way.

Yost: By the 1980s, given the prominence of the Anderson committee and the Anderson report, and what followed on that research by the Air Force and MITRE, Jim Anderson was obviously one of the huge figures in computer security. Did his writing that article lend immediate credibility to the area of intrusion detection?

Lunt: In that circle, there was. I mean, security sort of got launched, and security funding, I guess, got launched as a result of that. So there was NSA now starting to focus on computer security, you know, the origins of efforts like The Orange Book and all that. So I think it focused funding on that but beyond a small circle of people, security was still not known. Even in computing there was really no computer science, it was still an embryonic area back then. It really wasn't even a discipline. Like today, you can get a Ph.D. in it, you can study it at school and everything. That wasn't possible back then. It was an interdisciplinary thing, which is not a great thing in a university and so there were no journals for it. It was just this little backwater that some people were working in.

Yost: A handful of people, like Dorothy Denning, got a Ph.D. in computer science in the early to mid 1970s, but [pause]

Lunt: Yes. It was hard, though. And even when it was becoming established, it was still a small community of people who knew about it. It was hard to do anything commercially with it. The threat wasn't there. The threat was there in the; I can't say when it was there but apparently the fear of the threat was there in NSA, the government. Outside that, nobody was experiencing that because the internet was not widespread. Back then you could still buy a magazine like a trade magazine and see a story about somebody who was revolutionizing their company because they bought this minicomputer. Those were the kinds of days it was, right? So nobody was thinking about security commercially and as I tried to find commercial interest in intrusion detection, it

was almost impossible. I remember going to Japan and giving a talk there at Fujitsu, in Fujitsu Labs. And afterwards, when they took us out to dinner and everything was great, but one of the main guys there took me aside as said that over dinner, you know, this is not really a problem in Japan; only Americans behave badly. [Laughs.] That's not a problem in Japan. Of course, now it's a problem everywhere. They sell; they have a whole; now, in recent years, we've been working with Fujitsu on security stuff, here in Silicon Valley. Times changed.

Yost: Interviewing Roger Schell about the work for the Air Force, he emphasized that what they were seeking to build was systems that were mathematically proven to be secure and it wouldn't matter if it was built by the KGB, as he put it. So did the goals with the origin of this high assurance work in some ways kind of counteract the idea that there was a need for intrusion detection, because that's obviously detecting a breach in security, a compromised system or situation?

Lunt: I don't really think so. I don't think I've even encountered that point of view. You know, the practicality of proving anything correct was so far from being mainstream. You were going to have to do something else in the short term anyway, right?

Yost: Yes, there weren't A1 systems.

Lunt: Yes. And I think actually that may be like the beginning of the divide between the provable community, you know, everything's either provable or it's not secure; and then

the more practical minded folks like intrusion detection, you can't prove anything, it's all statistical. And there's still that divide between the formal folks and everybody else and there's not a lot of learning from one community to the other, which I started to find frustrating after a while, particularly when I was at DARPA.

Yost: Now you collaborated with van Horne and Halme?

Lunt: Yes.

Yost: Was that at Sytek or where?

Lunt: That was at Sytek. We did a massive amount of work. I may still have some of those reports; we produced these gigantic reports. I don't think they were published as technical reports, but [interrupted]

Yost: These were audit trail research reports, basically?

Lunt: Right.

Yost: And obviously, with IDES later on, expert system is part of the name. Did you understand that early work to be a form of expert system?

Lunt: It was around the time when I was at Sytek that I discovered — I might have been at SRI by then — I was only at Sytek a short time, only like a year or two, or something like that. But I discovered expert systems and it was partly through Tom Berson, because he knew this guy who was living near him in Palo Alto who was a founder of some small company, a start-up that was building an expert system for different applications. And so Tom thought it would be interesting to study expert systems from a security point of view so I knew about expert systems.

Yost: It wasn't Ed Feigenbaum was it?

Lunt: No, I think his name was Lee Hecht, and I think the company was; you know, Ed might have had some connection to the company, but; what was it called? My memory [laughs]. I think that one of their applications was something about wine, you know, they had these rules for how long you should keep what kind of wine, and stuff like that. Not a huge market, I'm sure [laughs], and I think they went out of business eventually. But I started learning about rule-based systems and I must have been at SRI because then, of course, they had the AI center and I had access to all these people. They were just around the corner from where I was working and so I started learning more about production rules and then we started thinking about how you would use those for intrusion detection. Such an obvious thing, in retrospect.

Yost: So while you were at Sytek was there any formal collaboration or partnership with SRI or is it just that you moved from Sytek to SRI?

Lunt: I don't think [there was] any collaboration. Of course, people knew about folks there; or knew about Dorothy and Peter; but I really didn't discover them, I think, until I started going to these meetings at Rome. And then it just felt like oh well, we're working on the same things and she invited me there.

Yost: You mentioned the Oakland conference. Did you also go to National Computer Security Conference while you were there? Did you start back in the MITRE days or later?

Lunt: I think I probably started when I was at Sytek, going to that. And the same with the Oakland conference; started going to both those because Oakland was right here. And then I just kept it up. I mean, that was sort of; those were the places to go in the field.

Yost: Can you compare and contrast those two meetings and the culture of those meetings?

Lunt: The Oakland conference was much smaller. There were usually about 300 people and it was more; it was meant to be a research conference. There was a larger audience than research, but everybody put research papers on the program and it was just like any of those other IEEE conferences. Very similar, whereas the one in Baltimore was more about practitioners and getting adoption, getting stuff into practice and learning from experience; and exposing the people who needed this stuff, largely a government

audience, to what's happening, what's available to them. But also having people learn from their experiences, I think. I used this product, here's what happened, here's what's good, here; here's what's bad. It was more practical minded and less academic. And much larger, much larger; so unlike Oakland, which had plenary sessions, there was you know, in Baltimore, three or four different things in parallel going on all the time.

Yost: When you were starting this intrusion detection research at Sytek what was the breadth of the group of potential users that you saw? Were you thinking that this would primarily be for government in the defense and intelligence communities, or that it would have broader applications?

Lunt: Funny, in those days, it was just a fun project. [Laughs.] Could we discover anything? I'm not sure I really thought about where the threat was. That came later when I was working at SRI and I had to get the money. [Laughs.] When I was at Sytek, someone else brought these projects in. I started developing these relationships with SPAWAR and with Rome but I think it was really when I went to SRI, and Dorothy hired me there. Maybe a year or two after she hired me, she moved and they offered me that position. And so I was now on the hook for feeding a few mouths and so what I thought was the people paying for this work, they want it for some reason. So I started learning about their view of the threat. Like I said, there was very little commercial networking of any kind yet, it was all very new.

Yost: So at MITRE and Sytek, it was becoming part of contracts that a corporation brought in, but at SRI, as a researcher, you were to go out and find a sponsor for the work?

Lunt: Yes. And it was still ARPANET in those days, we were on that network but there weren't a lot of people using that network. And then the Morris worm happened and it wasn't like if that happened today, it would have much bigger impact, right? But back then, it was limited to whoever is on that network and that was sort of the first personal experience that many of us had of the impact so that you could start imagining someone, for just the sake of doing it, could have a big impact on anybody who's connected to these networks.

Yost: I saw from your CV that you presented at the 1986 IEEE Symposium on Security and Privacy, on this area of research. Can you talk about that paper and the reception to that paper at that event?

Lunt: What paper was it?

Yost: I've got the title; it was "*The Trusted Domain Machine: A Secure Communication Device For Security Guard Applications*." 1986 Oakland conference.

Lunt: Oh yes, Security Guard. I'd forgotten all about those things! There was a collection of projects on these so-called security guards, which is a machine employed at

a boundary between two domains which are at different security levels. The security guard tries to make sure information doesn't pass from the "high" domain to the "low" domain unless it contains no "high" information. I guess it was a largely a manual task at one time and then there started to be machines to assist that person. And then some people had the idea you could have a totally automated guard. At that time, there were really very few — none — multi-level secure machines. So you'd have machines at different security levels and, of course, everything that went in would have to be labeled at that single level regardless of how sensitive it actually was. So you'd have things that were actually overprotected and then you'd want to get them out and not have to protect them as "high," but then it was risky to; so someone had to look at it. It was interesting because when I was at Sytek, it was all an academic problem to me because I had never used one of these things. When I was at MITRE, I remembered; well, they didn't have any security guards but we had a top secret computing system, right? Everything that touched it was sucked in and treated as top secret. Most of the stuff was not even classified, it was just a bunch of stuff they wanted to store on the computer, the only computer they really had. What I saw was when people needed to run a report from this top secret machine, they would just sign for it; like this one is unclassified, or this one's secret. Mostly, this is unclassified. So I kind of knew there was a problem there because how does anybody know what's in there? And as I started doing more you start becoming aware of how easy it is to deliberately encode classified information into these things. People back then envisioned pretty simplistic ways of steganography, and stuff like that. You could encode something in the first letter of every sentence, you know, and stuff could be embedded even without your knowledge. So you didn't have to be the malicious

party to unknowingly sign for stuff that was coming out of that machine, that it was lower than top secret. So at Sytek we started working on this trusted domain machine. I think it was pretty simple-minded. There was the high side, there was the low side, there was the middle part; they were provably separate; it was sort of inherited from this red/black separation in crypto. Crypto boxes are much simpler computers, of course, but you could chart the information flow there and show that there was nothing going from the red side to the black side unless you could go through this emergency bypass part; that had been part of the whole Blacker program. Then they proved all that red/black separation stuff. So we were trying to do something like that for this trusted domain machine. All the security processing would be done in the middle part, and that processing code would have to be proved correct, but anything running in the high or the low portions of the Trusted Domain Machine did not have to be proved correct, so it kind of simplified the problem. And yes, it was received well. The problem with an audience like Oakland is none of them are practitioners. They don't know what the real problems are. We were being told by our customers what are the problems. We could sell them an idea; and even then, you're selling them to a research funding organization and it's hard. I know, from being at DARPA. You're not in the mainstream of the agency, you don't have first-hand experience of these things; you're not an operator, you don't see what the actual problems are. So I think, you know, lots of stuff gets well received that ends up not being useful. I'll leave it at that. [Laughs.] But there were security guards, not that one in particular, but there were others being put into use and they would; I remember there was one that was used to sanitize messages that were being reported by intelligence about ships at sea. And so they would have things like — and this was being shared with other

governments, right? — so they would have rules about what was allowed to be shared with the Koreans, what was allowed to be shared with these other places, and they would take out; they had rules about what would have to be taken out of the message, or what would have to be substituted, or what would have to be reduced in accuracy and precision in numbers, things like that. Or what could just not be reported at all. And so those kinds of things were replacing people, started replacing people; and you can't prove they're secure. There's no way you can prove a thing like that secure. But this is where you get away from the provable; if you're going to have, if everything's got to be provably correct, you can't do anything. At some point it comes down to like a business decision, are we going to be able to do this kind of work or not? You just try to understand the risks. That's tough to do in security, it is tough to understand the risks.

Yost: At the National Computer Security Conference and the Oakland conference in the mid-1980s, is there starting to be a divide between these groups of researchers?

Lunt: Yes, I think there was always the folks who were doing these very formal proof kind of things, and then others who were trying to really do practical things that would be helpful to some agency. I think the real divide was the people that went to Oakland — there was some overlap, of course — but people would go to Oakland and many of those wouldn't go to the national conference because it wasn't; the proceedings weren't published, you know, archival proceedings; it wasn't going to help your resume; it was basically a practitioners' conference. So you started seeing a divide start to happen. I remember one project I did at Sytek. It was with Lockheed or some company. They

managed to sell this project that was proving something; use of formal methods to prove some security thing to them. Even there I could see it was going to be of very little use to use formal methods to try and do something practical to someone with a practical need. It was so far from ever being reality. You know, not to say it's not useful; it's nice to know, at least it's good to know you can design something as correctly as you can. It's a great starting point but you will never be able to prove that the actual machine that you end up with is actually secure doing the things that it has to do. It's just not possible. We'll get into this later, maybe, but I was starting to be interested when I went to DARPA, about how; because people were working on proving; you know, these papers that appeared at Oakland, where I have all these secure computers, now I want to hook them up together. What are the rules for composing them? So that the resulting combination is provably secure? Well, I started thinking it doesn't matter, because the individual machines weren't secure to start with. So what you really want to know when you compose them is how much worse off am I? And it becomes a quantitative kind of, or probabilistic, or some kind of quantitative measure that you want, not a "yes" it's secure or "no" it's not, because it will always be no, right? So "no" doesn't help, because I have to use these kinds of computing machines, so "no" doesn't help. So what would help? And we've made, I think, no progress on questions like that. I can't buy a machine today and ask how secure is it? How long will it be 'til this machine is no longer secure because the whole community out there has figured out how to break into it? There's still nothing that we have to offer to answer people's questions on the risk side, unfortunately.

Yost: So the different criteria that were set up in The Orange Book, even something that's designated as B1 or even B2, you really can't [interrupted]

Lunt: It's impossible. It's impossible to have provable security and get actual work done. You still see it now because covert channels is the whole name of the game in these things, right? You see, still, papers about "I can shine a light onto your office window and see from the vibrations everything you're saying" or "I can measure some emanations from a radio or your cell phone, whatever, and figure out all the data." So there are all of these side effects that can be measured; and inside the computer, of course, if you actually have direct access, yes, you can see timing, and you're sharing these resources, you can see disk access, and all that stuff. No, you can't really secure these things. And even The Orange Book, that was written before networking, right? So there wasn't even authentication as a requirement.

Yost: Before the Web, of course, but not before the ARPANET.

Lunt: Right. So there was, you know, passwords. You could still have only a password protected machine, and provable everything you could possibly prove, but there was all these other huge, unexplored places people could attack you.

Yost: I remember in interviewing Steve Lipner, he and Paul Karger, in fact, most of the decade of the 1980s worked on developing an A1 system for DEC. For commercial and business reasons the project was wound down in the late 1980s when Lipner felt they were close but he stressed the problem that they just had real trouble getting around was

covert channels. Do you think that's perpetually going to be a problem? Has there been better ways to address the problem of covert channels?

Lunt: I think it's just a fundamental issue in there's going to be side effects which are observable, not all of which are anticipated. So there's always that when you have multiple levels of security, so if what you're trying to protect is some secret and some adversary who's got the time and is able to get direct access to that system and observe stuff going on, even gets access as an unclassified user but can see the side effects of secret stuff going on, they'll be able to figure stuff out. It's like cryptography is all based on how long is it going to take? You know that you can't be provably unbreakable, right? So it's just how long is it going to take somebody to figure it out. Same thing with a covert channel. You give someone access for long enough, they'll figure out how to get all your secrets out of this thing. There doesn't even have to be malicious code in there. But today, the problems all shifted. The bigger threat; I'm sure there's big national security threats, right? But now like the whole economy is depending on this stuff so the bigger threat is the huge impact that attackers can have. They can hold companies hostage. Malware is ubiquitous. Companies have no secrets anymore.

Yost: Determined and well-funded adversaries can inevitably break through?

Lunt: Yes. Threats have changed so covert channels are there but they're probably not going to be exploited because there's all these other ways of exploiting systems now.

Yost: In industry, in the late 1970s and growing out of the SHARE meeting in the mid-1970s, IBM's RACF, and ACF2, and Top Secret come out as commercial access control products. Back in 1970 in the Defense Science Board Report that Willis Ware put out, he emphasized that industry needed to be part of the solution to computer security, yet when these products come out, by most accounts, they were not robust products and they're actually taking care to reduce the computing resources they're using at the expense of added security. Was there a sense that there had to be a partnership between researchers and industry in the 1980s or had that come about at all?

Lunt: Yes, I think the late 1980s, early 1990s; well, people were talking about The Orange Book and how this would be something that companies, if they built it, the government would buy it. The idea was that companies weren't just going to build secure systems and not know that there's a market. But back then the government was a big percentage of the market so they could say well, if I put these standards in, I mandate that people buy these things. Then you can lift confidence; put a good business plan together.

Yost: So it would be, too, can the company get a product certified quickly enough to reap adequate rewards with rapidly evolving systems.

Lunt: Right. But the government took too long not only to build these things, but to certify them that they meet A1 or whatever. That took years, right? So by the time all that happened, the whole world had changed. The government market shrank, products didn't appear, they would be out of date by the time they appeared compared to the non-secure

things on the market, and then the government never mandated anybody to buy them. So the whole thing just didn't work.

Yost: There was a SRI study by Hal Javitz on statistical analysis of data and IBM systems running MDS and DM to try to detect intrusions. Do you know when that project started and was that before you arrived at SRI or after?

Lunt: In my recollection is I met Hal at SRI because we were working on IDES and NIDES and somewhere in there I pulled him into the project. I was pulling in everybody I could. Working at SRI is a great place for that; there's all these different disciplines there. Somehow I got hooked up with Hal so he must have done some prior work for someone to introduce me to him. But so he had done some relevant work in profiling some kind of use of something on the computer, right? So then I started getting him involved in my projects and that was really the big jump in our statistical work; and he brought in Al Valdes and so between them, those two guys, they really did all the statistical stuff on the algorithms that we did. And it was all empirically based, so we would collect data; you know, I was guiding that work by hypothesizing what I thought would be interesting types of the features we would collect, and stuff like that, and then the kind of threats that we would be trying to look for. But they would; they basically would put all algorithms together. We would have almost weekly meetings, going through; here's the results of the statistics this week, and then we'd talk about how we could tweak the algorithms, and they'd go off and do that. It was a very iterative thing. And I think we were really the first

to really seriously pour any amount of effort into getting really good statistical algorithms for that problem.

Yost: How did the IDES project come about and was that an effort you led from the start?

Lunt: Dorothy had got the first project; between Dorothy and Peter at SRI. And I joined shortly after it was underway.

Yost: Then when she left, you took over.

Lunt: Yes. She, I think, was still there when Jagan [Jagannathan] and I were supposed to build this thing. [Laughs.] We got somebody to work on the expert system part, but we were still using just really simple rules; it was just kind of proof of concept to get something going. And I had to learn C in order to program; I can't even remember what part of the system I was doing. So we just had some very, very simple statistical things and some very simple rules and we demonstrated IDES. Based on that, we were able to get some more funding. Somewhere in there Dorothy left and the Navy kept being interested and we kept expanding the scope of the thing over time.

Yost: What was the size of the overall contracts to deliver IDES?

Lunt: Dollar amount, I'm not sure. But the first IDES, I think it was probably a two-person level of effort kind of thing. And then, over time, you know the last NIDES project I had was probably; you know, I think we got a few million dollars for it. In fact back then, that was the early 1990s, back then, it was quite a bit of money. I mean, way more than you would get like for a startup from VCs or anything. So, we had a team of maybe 10 or 12 people at SRI.

Yost: Was that at the IDES stage or not until the NIDES stage?

Lunt: That was the NIDES stage; that was at the end. That was pretty big, by SRI standards. I mean really big.

Yost: Did Becky Bace's research program contribute at all to that?

Lunt: Yes, they were starting to do; I guess they were doing much smaller projects. At least, the stuff I was aware of at NSA. They probably do things that they can't tell anybody about. But yes, there were some small things there so there started to be a whole community of people working in it. And I think before Becky — I'm trying to remember — there was somebody else at NSA. But while I was at SRI, one of the things that I did was; I'd heard about maybe five other projects anywhere in the world so I wrote these people an invitation; come to SRI; let's spend a few days, just talk to each other about these things. Like 10 or 12 people showed up.

Yost: Do you recall when that was?

Lunt: Must have been early 1990s. Like 1992 or something; I don't know; around then.

Yost: So it was the year of the final IDES report?

Lunt: It must have been before then. I could probably dig it up somewhere. But it was the first workshop that we had, and then I started holding them twice a year. It was just an information workshop. It wasn't a research conference, you know, academic reviewed papers or anything. We didn't publish proceedings. Anybody who wanted to be on the program could be on the program. So I sent out an invitation. Liz Luntzel, who is now Peter Neumann's wife, was the secretary for my group. So she was helping me organize these things. It was a lot of fun, actually. Every year there were more and more people until it got to be like 150 people. I ended those when I left to go to DARPA but it started growing into a pretty sizeable community. So at some point early in that set of workshops, I met Becky because she started coming to the workshops. I became aware of the work at NSA. And then also, international people were coming from Europe. People were trying out all different methods, all different techniques; you know, anything that would work for this problem.

Yost: Was it primarily people from the academic and research community versus startup companies? I know Clyde Digital was a startup.

Lunt: I don't know that those guys ever came. I think that they might've died out before we started that work. We were aware of them, of the fact of them, but not so much any details of what they were doing. A lot of academics but also there were government people; at some point, FBI people coming who got interested in profiling. They came from an arson background where they profile individuals. And so they started getting interested in approach like that for intrusion detection, as opposed to classes of individuals. Law enforcement people started coming. So it was getting to be a pretty mixed bunch of people.

Yost: Can you talk about the individuals that you brought into the IDES team and what they contributed to the IDES team?

Lunt: Yes. I found some people who were already at SRI in other groups who'd been doing programming. One of them wanted to do user interface stuff that was starting to be a new field. And one of them was doing just basic programming. I hired a couple of programmers and then I brought in statisticians, they had been in another group, a business focus group at SRI. So we sort of built up our team. There was a guy who I recruited to do the expert system who was a computer systems admin, and happened to have studied expert systems in school and he got interested. That was Fred Gilliam. And then I hired Alan Whitehurst, who I think was working on the expert system part as well. So there was a whole variety of people that could contribute so we had the expert system part, we had the anomaly detection part, and there was a big audit collection part, then, of course, we were looking at not just collecting from a single machine — which is how it

all started, like a single time sharing machine — but now it was people were using these Sun workstations on their desks and they were local area networked; we were looking at sort of a network picture, and then as well, the bigger network because now we were connected to other places on the internet. So all of those made it a more complex problem because now you needed networking skills, you needed all kinds of skills. Even just collecting the data was complicated because it was all distributed.

Yost: And during that period from the mid-1980s to early 1990s of NIDES, was this the most extensive and largest intrusion detection project out there?

Lunt: Yes it was, but the Navy security guy was giving us pretty much their whole budget for this. NSA was involved; they were involved in every aspect of the research funding in the government on computer security.

Yost: So all the DoD, NSA was involved in it.

Lunt: Yes. The agencies got their money from the NSA, or how; I can't remember anymore how that worked but there was some involvement from them. Anyway, there was a pot of money that SPAWAR had that they were going to use on this project and they ended up giving us all of it. They weren't giving us a lot of requirements or anything, so we were figuring it out, like, what did we need this thing to do? NSA was not directly involved, but sort of indirectly involved and so Becky was kind of indirectly involved in monitoring that project; as well as work going on at Davis and other places

where NSA might have, I think, been funding those things directly out of their own budget.

Yost: So was there any involvement as the project was going on, by individuals at SPAWAR about these are the kind of specifications this is what we needs to be done?

Lunt: No.

Yost: So it was pretty open ended research that you had the freedom to go with what you thought would be useful.

Lunt: Right.

Yost: And were you focused on building something useful for the defense and intelligence communities or were you thinking more broadly?

Lunt: We were thinking more broadly. By the time of NIDES, in the later days of NIDES, there were companies starting to get involved, like Trusted Information Systems, and Haystack, a little company called Haystack.

Yost: I remember Becky Bace mentioned it.

Lunt: Yes, they built a little intrusion detection system. It was bought by Trusted Information Systems, and then they were eventually bought by somebody I can't remember. So that thing, you know, was commercialized. There wasn't much of a market back then, but that was certainly what we were thinking about because it wasn't just a government problem anymore. By then, the internet was happening, the government was a smaller piece of the pie, there was a bigger world. You know people still weren't quite yet talking about national infrastructure, and terrorism wasn't even in the vocabulary back then. But it was starting; the research community recognized it's not just a government problem.

Yost: When you were holding these workshops for a variety of intrusion detection researchers, were any of those individuals hackers who had gained their expertise in intrusion into systems?

Lunt: Yes, there were like Tsutomu, who came to some of our workshops and I knew him because I was married at the time to a guy he used to work with at Los Alamos and I met Tsutomu when I was invited to go out there and give a talk on our intrusion detection work. Then he asked for a private meeting with me to talk about problems they were having there. He stayed in touch with me and from time to time; it became pretty clear; he was a clever guy and he was finding all these ways of breaking into systems. Then he wanted to talk to me about them and what he should do about them, and I always just told him who to report these things to. He was more on the hacker side, is how I viewed it; when you're looking for exploitations. Not that he was a bad guy but he was very

interested in finding these exploitations. Then we had other people like Dan Farmer. I can't remember his personal history but he had also been either on the hacker side or basically just finding exploitations. There was a growing community of folks like that who wanted to start helping on the security side and they had lot of practical knowledge about how to break into a system. So yes, they started coming to these things.

Yost: Can you tell me about the history and context with conceptualization of the Dissect tool?

Lunt: Yes. We were working in database security, and it was still about multi-level stuff at that time. And we had I think rather naively; the whole community somehow, they very naively thought that people would actually use the system that was labeling individual elements in rows of tables with a security label. But this is an example of what I mean by being remote from the actual problem. Today, you just wouldn't take the risk of mixing data at that granular a level. It would just be impossible to protect, even if you couldn't infer stuff, even indirectly, it would just be way too risky. No computer could separate that stuff. But we had come up with this notion that you could have these differentA1 systems, if they were ever built, to separate whole chunks of classified information at different levels, and then in the database layer you could have different copies of the database running at different levels. The top secret one could have access to everything and the secret one, access to just secret and below, and all that stuff. It could assemble the secret process view of all your data and give it to you, if you had a secret clearance. But there were still problems because when you look at how relationships

among data are constructed in these database systems, there are explicit relationships among data elements and you, a human being, can start thinking about well, this thing is pointing to this other thing, and then I can sort of figure out from what I can see, that there's something else there that either I can compute from the values because there's some direct relationship, or I can infer from the values because it's just common sense or whatever. And once you start going down that path you've got not only what could you prove from an information flow point of view, but then what can an intelligent being surmise from looking at all this stuff? And so we decided to take that problem on and by then I had met my present husband, Tom Garvey, he was working in the AI center there at SRI. He was interested in this problem and he had also worked with me on some intrusion detection stuff and introduced me to a bunch of people in the AI center that were very useful in solving these problems. I knew there was going to be some AI reasoning necessary here so I reached out to him, but also, he's a practical guy, too, so he's always interested in transitioning these things into actual military use. And he had worked on some project where he knew there was data to be had. He worked on some command and control project and he knew Rome Labs had all this data, and I needed data to actually illustrate this problem. It's very limited if you're going to make up a bunch of data and show you've got a problem, right? So I wanted to get access somehow to some real data and demonstrate that you can infer a bunch of stuff and then demonstrate that if you use Dissect, you can fix all those problems. So he introduced me to some folks at Rome and they didn't give me the actual data but what they gave me was the schema, actually a huge entity-relationship model of the data. It was printed on these rolls of paper, you know, like three, four feet long and all rolled up so you could unroll them and

they'd cover the room. [Laughs.] A couple of these rolls of paper and they had drawn out these entity-relationship diagrams and it was tons of types of data that were going to be stored when they built this database; things about airplanes, and pilots, and air fields, and payloads, and targets, and people, and missions, and all kinds of stuff. And it was all in some language that I didn't understand because it was these abbreviations of these military things. So I used Tom to help me decode what are these things? What does this mean? Some of them were very detailed stuff like the angle of the wing on the airplane, stuff like that. Some of them were more understandable like how much gas was going into the airplane, and where and how far was it flying. You can infer things like how far was it going to fly if you were putting in the gas, maybe. So we made a database system; I think we must have been using Oracle. I created all these tables to recreate those relationships. And then, with Tom's help, I put in what is the standard classification of these things, you know, so targets are going to be classified higher than who was the pilot, something like that. We labeled all of those fields. We could label entities and attributes. Then I can analyze this thing with the help of algorithms that I had worked on with people like Mark Stickel, who just passed away; really sad. But anyway, I worked with Mark and a few other people. A young woman that I hired, as well, in my group, named Xiaolei Qian. We built these algorithms so we could go through and by understanding how things were related to each other, and other sort of common sense reasoning that a person might bring to it, figure out what you might infer that's higher than your security level. So the tool would come out with a list of these things, what we called "inference problems"; as well as what was the inference path that led to these things; and then you could decide, along this path, which of the components you could

upgrade, so as to break the inference path by raising the classification of something in the path. After a while we put a lot more complexity into it. So the tool would suggest what was the lowest cost set of information components to upgrade because upgrading the level of some data is one thing, but it's got to match the real world level of the thing, right? It does no good to make something secret in your database if it's observable out in the world. It costs something to classify it, you actually have to hide it or mask it in the world. So if you're actually going to need to classify something in order to break an inference, you have to take all these real world measures. So we would try to figure this out; we would encode a bunch of cost models in there so the system would know what would be the cheapest things that you could upgrade to break those inference paths. We demonstrated the tool and all that. The tool is only going to be useful if people have multi-level database systems, which they don't. But it was interesting because I think prior to that work, people thought that you could just label data, end of story. But when it's labeled at this very granular level, it's not like you just have a bag of labeled data, you reach in, and if it says it's unclassified, fine. You're getting so many pieces of things that can be pieced together there to infer a bigger picture. And the bigger picture itself may be classified. I got really interested in this when people start talking about an "aggregation problem". You know, from a research or an academic point of view, it's just like a mathematical problem and it made no sense at all. I tried to figure out what is the sense or the intent of having a rule that says you could get up to 10 phone numbers from the NSA. As if anybody could call up and ask for a bunch of phone numbers. That is itself a fallacy. You couldn't just call up and say give me nine phone numbers. But you could call up with somebody's name and ask for their phone number. So if you should do

that nine times, if they figured it was the same person calling the tenth time, that was it, no more numbers. That was the way the problem was posed. I'm thinking this makes no sense. How would you have a multi-level database system that would protect, enforce that kind of a rule? And none of the pieces was classified, but if you got enough of them, it was classified, right? So people started to think of these complex rules you could encode, you know, somehow, and here's where you have a problem if you're in the provably secure community dealing with a thing like this. So I got real interested in what is it that they're ***really*** trying to protect? What's really behind here that could be encoded as a rule or policy; because sure, you can replicate these senseless rules, but they're not really in any way secure because my nine numbers could be added to your nine numbers, pretty soon we could get the whole phone book. Maybe there were some other kinds of measures you could take instead; that's kind of what I was interested in when I got interested in doing Dissect.

Yost: You first began doing research at SRI in databases with Dorothy, is that correct?

Lunt: Yes.

Yost: But she was gone by the time you . . .

Lunt: By the time we did Dissect, yes, she was gone.

Yost: You talked a bit about it but can you expand upon that earlier database research she did?

Lunt: Yes, when I got there they were basically doing a model for a multi-level secure relational database system and they were basically writing down a bunch of math defining what these things were. Relational databases themselves were a fairly recent invention at that point. I got involved in that project; Peter was also involved with that, as he was also involved in all the IDES and NIDES stuff all along, too. The multi-level database idea turned out to have these odd kind of properties, so if I had a table, say it's a list of departments and people in departments, and some of the people are classified and some of them aren't. Or something about the people is classified; say maybe their hair color is classified; some of them have a classified hair color. So now I see; I retrieve information for this person, I see there's no information about their hair color because I'm not cleared to see it. So how do I interpret that? I might interpret that the hair color is missing and put in a hair color, right? The system can't tell me that sorry, there's already a hair color for this person; and the database under normal circumstances, without this kind of security would say sorry, you're not allowed to enter a duplicate value for this field. But we couldn't say that without revealing something that would lead to covert channels, namely, whether or not a specific piece of classified information exists in the system. So we came up with this idea; there were so many variations of it that we toyed around with; Roger, and me, and Dorothy, mostly. We came up with all kinds of crazy things, like you could have in the same cell multiple values and stuff. But what we finally ended up with was you would just enter a second row for that person. There'd be one row

that you would retrieve that was secret that said Bill has green hair, and then there would be the row that you would retrieve if you were unclassified that says Bill has brown hair. And so basically, you'd have these different versions of reality, depending on what you knew, right? You can see how it kind of matched with the real world, as we speculated that it would be if there was a secret thing out in the world. You could see part of it, like you could know about the person but not their hair. There would be some story that they put out there about the hair not being visible, or even a cover story about their hair being brown and not green, then everything makes sense, right? You know that governments do this, right? How do you explain that the troops are massing at the Czech border? Well, you could say there's a big exercise going on; there's cover stories like that put out there. At some level this could make sense, but the fact is that for labeling cells in a multi-level table, it was such a granular level that a lot of people had problems with the whole concept, but there was no way they could get rid of it without introducing channels, and still have labels at each cell. One way to reduce it would be: well you can't have labels now on each attribute, but you can have labels on a whole entity, or a whole type of entity. But still it would come up and there was just no way to get rid of it. Say if an entity like Jane Doe working at this company was classified, but now an unclassified person enters another row for Jane Doe; it's the same problem. I think it was one of the identifying kinds of moments that we had; we called it polyinstantiation. We were all out to dinner, we were calling it multiple value blah blah blah; we had a million ways to refer to it. We were all out at dinner one night and Peter Denning was with us. He came up with the word. He goes, "its polyinstantiation!" It was meant to be a joke; he was always — like Peter Neumann — he's a punster, right? So I'm sure he was thinking

polyunsaturated or something like that. But it just stuck and we started referring to it as polyinstantiated. It became this huge, controversial thing. The fact was that everything led you to it and it was a way of dealing with covert channels. If you're going to be A1 and you wanted to be rid of the covert channel, you're going to have this. If you were not going to have this, then you're always going to have the risk that people could infer or discover the missing pieces.

Yost: Were there principle critics in the research community for what you had labeled polyinstantiated?

Lunt: Yes, most of them were the database people. [Laughs.] Because you know, relational databases were being invented, had been recently invented, and they made these — relational databases anyway —they made these rules. They were just built on a bunch of simple rules like you had a table, it had a primary key, the primary key uniquely identified a row of the table, you can't have two rows of the table with the same primary key value. Obviously, if you're a database person, this thing, polyinstantiation, was no good. So what does a primary key mean anymore? We had to redefine all of that stuff. At the time we were building just a model of this thing, and that was Dorothy, me, and Roger, mostly; we had to define what does primary key mean now in this multi-level database? What does foreign key mean in the multi-level database? How do you reinterpret all of the definitions and concepts. So we did that, and then the access rules, like if you're a person with this much access what do you get when you retrieve data, given all the different "versions" of the data that exist? So that was the beginnings of the

work on that that I did with Dorothy. But I think she left before; I came in somewhere in the middle of that; but she left before we started building the prototype. We got funding from Rome to do; it was all funded by Rome. We got funding to build a prototype of this thing. The whole idea was, like I was saying earlier, you would have different copies of the database application running; the whole database process and its application running, like one top secret version, one secret version, one at every level. "Higher" versions could see the data of "lower" versions. So you could have a "high" ***view*** that merged in data from lower versions. That was also part of the "View" in the name SeaView. And the A1 system underneath is going to provide all that separation between processes at different security levels, so you didn't have to prove anything correct in the actual database system. That was sort of the invention that SeaView is; to minimize the amount of trust you had to have in the database system, since you already had this trusted operating system; just leverage that trust so you don't have to prove anything about the database system, which was immensely complex and would've been very difficult to prove anything about. But then once you've got that thing separated then you've got your top secret view of the world, your secret view of the world, and all that. Okay, we have the whole architecture ready to go, we just needed the A1 operating system. And that's where Roger Schell's company came in; they were building Gemini. They never finished it, to my knowledge, but all the time we were building this thing it was going to be done "any day now." They were porting it to some new processer or something that was slowing everything down and it got to be obvious to me it wasn't ever going to be delivered in time for my project anyway, so what I did was I switched from using Gemini as the trusted operating system to using Trusted Unix, which was a new commercial

product at that time that was multi-level secure. It wasn't Class A1 or anything, but we were treating it as a stand-in for a future A1 operating system. I went to Oracle and brought them on as a subcontractor I already knew a bunch of their security people. I met Bill Maimone there, a fabulous guy working on the Oracle team; and Linda Vetter, who led this Oracle team to implement SeaView in Oracle, because Oracle, by some piece of luck, already had an architecture for a distributed database system where it would have a set of different Oracle processes running on different processors. So say you had three database processes running: they had a way of combining the data from those. This was the same kind of idea we had for the SeaView architecture, only in SeaView these processes would be at different security levels, and needed to run on a multilevel operating system.. Oracle's design wasn't done for security reasons, it was done for other reasons. So they had the right kind of architecture in Oracle that would map to SeaView if we could port it to Trusted Unix, and Trusted Unix could treat the separate database processes as having different security levels. So I went back to Rome and said we can't; we don't have an A1 operating system; we're not going to have one so we're going to just assume there's one under there, and use Trusted Unix instead. When you finally get one, you can put one there and this whole other thing we're building on top has the right architecture. The Oracle team came in and built the MLS applications because we were delivering the secure operating system, the MLS database system, some applications operating on multi-level data, and showing all the labeling as fine-grained element-level labeling, showing the polyinstantiation, and all this stuff. So we delivered all that to Rome and at the end, I remember a conversation with Dick Metzger, who was head of all the research there and, you know we talked about how Oracle, based on all

this work, had decided to come out with a commercial secure database system, and some other companies were, too; Sybase; but they had sort of watered down the idea. It wasn't element-level labeling, it was going to be row-level. But I had this really interesting conversation with Dick at the end of our project; he felt that the whole thing was not successful because it wasn't being commercialized as an A1 database system and why was that? We talked about that for a while and it just seemed that industry was not ready. They didn't own this problem; there was no market for it; it was too complicated. We might be happy to see labels all over our data but the business world wasn't. They wanted to see the data, not labels. They didn't think about data that way. They didn't think about people at security levels. There's personnel data, there's financial data, there's secret projects nobody knows about, but many of these things are never merged together in a single system so you just don't have the same problems, so nobody was going to build this thing, it couldn't be commercialized. The fact that Oracle and Sybase were even considering building these things was, I thought, real progress but it wasn't enough for people who wanted these A1 things. So then I really started thinking about what was worth spending my time on? [Laughs.] It's one reason I went to DARPA. You know, really what good was it to work on a problem that could never actually have any impact on anybody, right? You find, I think, in a research community like that, that most of the performers are academics and maybe 90 percent of the work will never have any impact. It may have indirect intellectual influence on something but it's a very tenuous line that you draw. Maybe you have to pay all that money for all that work in order to explore a big space and then figure out what you're going to ***really*** do. Maybe not. I don't know, but I started getting really kind of disillusioned with this stuff. I talked with people at

Rome and asked can we just take a bigger view of some of these things and help decide what are good problems to work on and what are not good problems to work on? But then I had an opportunity to go to DARPA, and that's where I really expanded; it forces you to think bigger; much broader even than your own field. So I was in two fields then; I was in intrusion detection and database security. But then going to DARPA and being responsible for security you had to sort of own the whole problem and what did the military need? What did the whole country need big-picture-wise in security; it gives you a whole different way of looking at things.

Yost: You brought up SeaView. Can you talk a bit more about that; its origins and your role at SeaView?

Lunt: Yes. I guess there had been; even when I was still at MITRE, there had been some Air Force study at Woods Hole; it brought a bunch of people together for three weeks and they wrote a report on some recommendations. There had been some projects leading up to that report. There were like two or three main ideas on how to build a secure database system. One was this Hinke-Schaefer model, which assigned labels to columns in tables. And then there was some other model, which I'm going to forget what it was called or whose it was. This other model assigned labels to rows in tables. And then there was some idea of, I can't remember if it was in that report or not; but somehow there was this idea of labeling elements that were in intersections of rows and columns. And someone else was just labeling entire tables; maybe Jim Anderson or somebody. These were all still just speculative; nobody had built anything or thought about how you would build

them. But there had been reports; Rome had paid for some studies; and there were some technical reports on just kind of the preliminary thinking about stuff and how do I apply Bell-LaPadula. Everything was how do I apply Bell-LaPadula. So then there was this study and they issued a report, and it was very influential. And I guess from that, was this whole research agenda. So Rome became the place that was funding everything, and I'm sure it was NSA giving them the money, but they had a lot of money in those days and so they were funding a whole bunch of projects on secure operating systems and secure database systems. So by the time I got to SRI, Dorothy already had one of those projects going. That was the SeaView project that she was doing with Roger Schell. There were others. I'm trying to remember. There was one that was involving Sybase; I can't remember; I'm pretty sure Rome was funding that. So there was a whole set of different ways of doing secure database systems laid out in this research report and Rome was funding a few of those variations. So it was a big part of what we were doing at the Oakland conference, reporting out on these things. I think SeaView was in many ways most ambitious, looking at element-level classification, the most complex version of this stuff, to see how you could make that feasible.

Yost: Do you recall what year NIDES started and were there any fundamental changes between IDES and NIDES or was it more a continuation in structure and the application of theories from IDES?

Lunt: It was basically just a straight line continuation. We called it Next-Generation IDES. [Laughs.] You know Dorothy had the idea before she left, of making it distributed.

So that was one aspect of it, so that it would be monitoring a distributed system, not just single workstation, while IDES was just monitoring a single workstation. So she was able to see that that was not going to be able to scale up as everybody had their own workstation. There was no organizational view of the thing or the systems being monitored, so NIDES was a distributed thing where you would have the ability from a single point in the organization to see what's the big picture of what's happening. So, yes, it was just a continuation. It still had the combination of expert system and anomaly detection. And it was scaling up that idea and creating the extensions to the algorithms that you needed to tap into this much more diverse environment, many more features that you wanted to look at that had to do with connections between people, connections between machines, connections between my machines and machines outside my organization, and stuff like that where you could start seeing if there were patterns.

Yost: What impact do you see that IDES and NIDES had for subsequent research on intrusion detection by others, both in research and in practice?

Lunt: Well I think one of the; maybe the main impact is it kicked off a whole bunch of related work and you know the stuff that eventually turned into the products that are ubiquitous today. You can trace all of that back probably to Dorothy's initial; I mean, she came up with the initial paper; actually to Jim Anderson, really, but then Dorothy, and then from that, we started building IDES, and from that people had a whole variety of ideas on how would you analyze data to find the bad stuff in a huge collection of data. So I think even though it was never directly commercialized; well, you know it did lead to

other work at SRI that did become commercialized, I think, more recently. Yes, it led to a whole bunch of stuff like the work of Phil Porras, who came from Dick Kemmerer's group at Santa Barbara, which had seen what we were doing and wanted to do it differently; and yes, so there was a whole bunch of stuff like that. Typically, you see that kind of thing in academic communities; there's a thousand flavors of solving a problem. So that was happening.

Yost: And were you starting to see that happen before you left SRI or was that part of the frustration that it took so long to become adopted [interrupted]

Lunt: Well that was happening before I left, yes. I guess part of my frustration there was people didn't have any good access to examples of attacks, right? So they had all these hypothesized attacks that they just kind of extracted from their heads but no direct knowledge or experience of the people; they weren't actually studying the people who had the problem, although you started to see the hacker-type people were starting to join the community and they were contributing. But basically, there wasn't a whole lot of bad stuff in the wild as compared to today, and people who knew about it were few and far between. And they had their own little proprietary collections of known attacks and nobody was willing to share that stuff. From time to time, there were people like Karl, I think was one of them, Karl Levitt, of wanting to put a machine out there as a honeypot, you know, a tempting machine to attack, and just collect a bunch of stuff so you would have some examples of attacks. So that was a big gap in the research community that we didn't have access to that. And that was one thing I hoped to do when I went to DARPA

was to somehow have a collection of these things for use by the research community so they could build better systems.

Yost: I'd like to mention several intrusion detection systems and just get your opinion and feedback . . .

Lunt: Sure.

Yost: . . .Haystack.

Lunt: Yeah, that was one I mentioned earlier. You know, there were all doing really close to the same kind of stuff. Haystack was the one that was commercialized or sold to TIS; and then they were bought by some other company. Yes, it was a good system.

Yost: Midas?

Lunt: Oh yes. Midas was part of our work. Actually became part of our work [when] we hired a guy from NSA, Alan Whitehurst who was doing that work. So it was another way of looking at the expert knowledge that you could bring. An expert system is not just an expert system; it's all about the knowledge, right? So he had another way of thinking about, reasoning about intrusions and so we included that in NIDES because he was working with us for a time.

Yost: And W&S developed at Los Alamos?

Lunt: Yes, that was interesting work that was based on machine learning, and I think that was really promising work.

Yost: And that was contemporary with NIDES?

Lunt: Yes, all of these things were contemporary with NIDES, I think. Not so much IDES because IDES was just very limited. I think it's when we started working on NIDES, I think that all of these things had started popping up by then. There was the stuff at UC Davis, and I can't remember what that stuff was called; DIDS or something like that. That was very interesting. And also, the work that Dick Kemmerer was doing called U-something. You probably have it in your notes. It's so hard to remember these things. But he had started with one system and then he had another extension of it to make it distributed so people started getting interested in distributed attacks, and could you see an attack progress across a bunch of machines and you know, stop it before it got to all the machines. That sort of thing.

Yost: In 1992 you became director of the Security Systems Research Group. Can you tell me roughly how large was that group and the scope of the activity? Were there other projects that we haven't talked about?

Lunt: By that time, the group was only 10 or 12 people, and that didn't include people outside my group who were working with me on these projects. We probably had six, seven contracts going at any one time. And then we also had subcontractors on most of them. We had one on secure real time operating system with Doug Jenson and his team back in Concurrent Computing Corporation. That was also funded by Rome and used people in my group and people in another group at SRI. Doug's group was a subcontactor. We had a couple of database things going; a couple of intrusion detection things; you know, we had multiple of these things; Peter always had something of his own going on and it was just, you know, a collection of things going on. But the biggest things that we had was the core database research, database security research, the intrusion detection system, and the inference control kind of stuff that we were doing.

Yost: Having such a group of talented researchers together at SRI must have been a wonderful experience. There wasn't any university, certainly, that had that many talented researchers in computer security at that time, was there?

Lunt: No, because it's really hard to do that kind of work at a university because it's crossing disciplines, which is hard. Usually, some professor gets the money and you know, they're in some department, and these things, if you're security in databases or security in operating systems, you need to bring these different people together. And then if you're going to build something, you know universities are going to lose interest very fast when the bulk of the work gets to be about building, experimenting with the data. So I think unless you have serious government funding and a place that can work on

something for the long term, with a big enough team to actually build those things, then you just can't do it. So that's why you see more of the academic stuff will be smaller projects and more limited in scope sometimes; more fundamental, maybe, very early thinking about stuff.

Yost: So SRI was probably even more about bringing different disciplines together than even RAND was? It was more diverse.

Lunt: Yes, I think RAND had maybe different emphasis than SRI had. We were really fortunate to have the AI center there with, you know, they had basically invented the field of AI. [Laughs.] So some of the really best minds were right there, and then they had a big business group, back when I was there so you also had people who were working with companies. Hal was in that group and Al Valdes so a lot of the right ingredients were right there.

Yost: You went into your decision, some factors in your decision to move to a program officer or leadership role with DARPA. Can you talk about the different research projects in computer security that you funded? What are some that stick out as especially interesting or influential?

Lunt: I continued a whole lot of work in intrusion detection, but what I tried to do there was foster a research community that was not each building their own; you know, their own intrusion detection system. I know from having done it that most of the money is not

going into the new, creative thinking, but a large part of the money's going into the system side to make that thing work. And I didn't want to have to pay that price at every institution I funded so what I wanted to do was just pay for the core research and then have a common infrastructure that it would run on. And then what I also asked them to do was develop ways, interfaces, so that I could combine these things together and anybody else could as well, in a flexible system so that if I wanted to have an expert system and a machine learning component, and a this and a that, and whatever, I could combine them. But that was probably a bit of a pipe dream. What we did; I still remember one PI at my first PI meeting, one PI in the front row raises his hand and says, can't we all have our own standard interfaces? [Laughs.] Like, why do we have to work together on this? It's hard to get all these people to agree on things. But we did start trying to standardize on names and formats for expressing the findings, and the attacks, and all that. And so that was; a guy at Davis was leading that part of it. I had various people trying to help pull all this stuff together and then I had Lincoln Labs, who had a test facility. I was going to see how well did these things perform on actual data and then how much improvement was there over the life of the program. So we were able to show most of them didn't perform very well. Most of them, you know, left a huge set of stuff undetected and their progress was slow. Progress was hard. And you can make rapid progress against known attacks. Once you know what to look for, you can look for it. But we always asked Lincoln Labs to include new attacks so nobody was going to be able to anticipate those and people had a really tough time detecting that. It was kind of a perfect result to keep getting money from the government because there's a big way to go; but that progress was slow and was not encouraging. There was progress but it wasn't as fast as you'd like. So that was one

aspect. I got interested in network attacks so I was talking to telecommunications people, people in government who were actually involved with this stuff; setting up, managing computer networks. There were, of course, networking programs at DARPA so I interacted with the program managers there.

Yost: Is this the information survivability program?

Lunt: Yes. So we were looking at how do you get security into those networks? How do you get security into the new emerging applications of the World Wide Web, like e-commerce and stuff like that? So that was one whole line of work. So there was also; I had some responsibility for some work that Kirstie Bellman had started, and she moved off to do something else; these project she started were more in formal methods for hybrid systems and those are mostly being used for safety applications. Some security applications too, so it was a little bit of a mixture; I didn't create it but I was sort of inheriting it from her and continuing it. And then, for information survivability I started this line of thinking that drew from those projects; and I included people I drew in from other fields like fault tolerance and stuff, to consider the problems that we can't make 100 percent secure systems and there will be malware, there will be intruders at any given time. When we are; when the nation is engaged in some kind of a war, you know, some kind of an adversarial relationship with others, these adversaries will be not only doing these standard kinds of malware like we were experiencing back then, but they would be figuring out what assets that they wanted access to, or what effect they wanted to create, and they would craft special-purpose attacks for that, or special-purpose malware for that.

So it wasn't enough just to have something that could detect everything that we see in the wild, but you had to also anticipate that people were going to try and thwart you in particular ways and they were going to seek access to those particular things. And that you had to be able to operate even though those things were happening. So you do everything you can to prevent those, and stop the ones you can detect while they're still in progress. But mostly, you have to accept that they're going to progress faster than your ability to detect them, and that there will be, your systems will be in an insecure state probably all of the time, but especially a lot of the time that the nation is actively engaging with an adversary. And that you needed to have the system keep going, you can't just stop. Everything can't fall apart at that point. So you had to be able to contain, and keep going; and that's why I called it survivability. You had to be able to survive and keep going, limping along or whatever. And then I started looking to see how was the military thinking about this in say, command and control systems, and I started learning about how they were testing their new command and control ideas in these military exercises. They started trying to test computer attacks as well as other kinds of attacks. But they wouldn't let the security people in to try these attacks until they finished trying out all their other things, because they knew that once the security people got in there it wasn't going to be working properly and they wouldn't be able to test it. [Laughs.] So it was; it looked to me like they weren't actually facing reality. You know that these people can attack your system, and it's not going to work as you planned, and you're not doing anything about that, right? You're not willing to test under realistic conditions? So you're not actually having an active part of the system try and compensate for that? So if your command and control system isn't working properly you're just going to give up. I mean,

you don't have a fallback position. Like you would with any kind of a physical attack you would have: here's my plan, here's my plan B, here's my plan C, and all of that, right? So that just wasn't in their thinking for computer attacks, so that's what I was trying to get going in my later years at DARPA. I was there for four years and so I launched that information survivability thing.

Yost: That was 1995?

Lunt: I was there 1994 to 1998.

Yost: Who was the director of, of the DARPA IT office, it had originally been IPTO, back in the ARPANET days, but then it became?

Lunt: It was CSTO when I went there, and it was John Toole who was the director, he was the office director.

Yost: Yes. Of course, I know him from his time leading the Computer History Museum.

Lunt: From the museum. Then, of course, it was Howard Frank who came in. And then we joined with SISTO; I can't remember if Howard joined before or after there was a reorg. There was a new director; I was there under three directors. There was some kind of a reorg. We combined with this other office. Then we were renamed ITO, Information

Technology Office, and then David Tennenhouse and there might have been some name change in between there, but David Tennenhouse was office director when I left.

Yost: To what extent was the focus to fund defense-related research versus kind of broader impact research beyond just defense?

Lunt: DARPA is part of the DoD so it's all got to be relevant to DoD. So that's just the bottom line going in. It is recognized in the office I was in that these technologies have broader applicability and that one way to affect the military was to have stuff take hold in the commercial world and then the military would buy it, rather than have purpose-built systems for the military. It would have many advantages, one of them being cost. So that was one of the strategies we tried to do was to have one part of the technology be attractive commercially. The challenge there is it gets so watered down when it is commercialized. The security needs aren't the same in the commercial world so you don't end up with the thing that the military needs when they buy it back so then you have to figure out okay, what else do you have to have purpose-built for the military to complement these leaky systems that they're going to buy commercially?

Yost: The computer science field, in the past couple decades, there has actually been a substantial decline in the percentage of women in computer science. Just looking at intrusion detection, a number of women are very prominent; you, Becky Bace, Dorothy Denning. Do you see any reason for this; was it the underlying mathematical background, what do you think was influential?

Lunt: You know, many people have tried to speculate on not just in computer security but the whole of computer science. It's a well-known problem and I'm not sure what it is. It's true [that] back in those days there were quite a few women. Cynthia Irvine comes to mind. You should interview her if you haven't. She worked on SeaView. She's the one that came up with; I think between her and Dorothy, came up with the name SeaView because we were having monthly meetings in Monterey, always by the sea. We'd always lunch someplace with a view. Yes, there were quite a few women and now, if you go to a security conference, I feel completely out of place going there, not only because I'm one of the few women but also because of my age, too. There's all these young, nerdy guys. When that starts happening, it doesn't matter what kind of gathering it is, it feels unwelcoming, right? And you don't feel comfortable; you don't even want to be there. Once the critical mass reaches a low enough point, then no more women are going to want to go there. So I don't know how that started but now you definitely can feel that. I don't have any explanation for it. Has anyone else offered you any?

Yost: No. At the Babbage Institute we held a conference on computing and gender and what the problem is, historically. And really, that was our best conclusion; that a non-conducive culture and unattractive environment happened gradually. A lot of surveys have been done and women students basically just don't like the culture and the environment in computer science is very unattractive.

Lunt: Yes. Nowadays, when you have things like you can work on health care applications, or you can work on energy; I mean, now it seems more; it could become more attractive.

Yost: Yes, and more and more the former; many of the former library schools have evolved into information schools; the information school at Michigan, the information school at Indiana; there's more of a gender mix within these schools; at Indiana, the informatics school is also computer science, so that might be a better way to change the atmosphere and the culture, but broadening it to the study of information.
Can you talk a little bit about your work here, you have become the director of the CS Lab at PARC.

Lunt: I came here as principle scientist in 1999 and then I was part of a new security group that had formed. And then I became manager of the security group because the manager was not a security person. He felt that we should have a security group here at PARC, he hired a bunch of security people, and then he went back to what he wanted to do. So I became manager of that group and then when my boss — at the time, Richard Bruce was managing the lab — he wanted to go back and do research in biomedical stuff so he became a group lead in the lab and I became the lab director. I'm not sure what year that was; it was like 2003 or something like that.

Yost: And how large at the origin was the security group, and how large has it become?

Lunt: Right then, when I joined in 1999, it was maybe four people. And now, like the security group is now still like three or four people. It had been bigger at one time; it had been maybe twice that size but then Google hired all those folks and I had to start replacing them. Google's hiring everybody [laughs] not just security people.

Yost: What area of research has been the focus?

Lunt: Now, one of the big foci is on cloud; secure computation in the cloud. When I came here they were working on trusted printing, so you could somehow authenticate a printed document as the actual authentic document and we were looking at a physical means of doing that. There's a variety of system security related things, you know, security in the service of other kinds of projects at PARC. What we then called ubiquitous computing was a big one, where you're starting to have mobile users, giving them context-aware services that are aware of their location and other things, who are they with, and so privacy and security were a big part of that. And then we did useable security, that was something that Diana Smetters — she's a person you might want to talk to, she's at Google now — she kind of launched this area called useable security and there's now a whole field of useable security, about making security easy to use. Diana's thesis was that one of the reasons why security isn't adopted, most of the great ideas are not adopted, is because nobody's taking the trouble to figure out how to make them easy to use. So she worked on, one of the things she worked on here was what we call near-zero administration networking. So she had a way of allowing devices to be securely connected to a network by provisioning them with certificates and do it in a way that

would just take seconds rather than hours to do. The best thing you could do before that was a thing that would just take hours and it was menu after menu after menu of stuff you had to answer questions to, stuff you didn't know the answers to and nobody had any confidence that they'd done it properly, and so you couldn't count on the security of the thing, the final result. But she came up with this thing that basically just made it so it was completely automatic and all the user really needed to do is be there in person, or a registrar be there in person with the device, so that there was no way that some third party could interrupt the channel as the provisioning was taking place. So useable security was a big one. We have a big group of scientists here; you know, ethnographers; PARC's invented the field of corporate ethnography using anthropologists, sociologists, psychologists in the workplace to understand work. And so we had those people — and they work in my lab — they were part of getting this useable security thing going because you could observe people and see the issues that they're having, and you could figure out how to make the technology work for them. And then you could evaluate how useable it was, you know; so that took off here. And then recently, the other big focus for the security group is we have this big project here called Content-Centric Networking that we started around 2006. The idea is to get rid of IP addresses as the way we think about communicating on the network. Instead, the network is about the information that you want to get or that you want to publish, and you don't need to know where it is on the network. So Content-Centric Networking has you just deal with the content. The content is named and the network knows the names so you can start routing stuff just based on their names and you don't have to know where you put them. You just put it out there. The network becomes like a big storage layer; the information can be stored anywhere

and you don't protect it through protecting the pipes. So it's not about end-to-end protection of a pipe, which doesn't really give you security for the content, but with CCN it's the information that has to be protected, it has to protect itself. It has to be possible for a consumer of the information to directly authenticate it so that it came from the authentic source or the claimed source. So security's a big part of that and what makes CCN possible. What I think is important when I give guidance to our security group now is that we shouldn't look for some academically interesting problem to work on. That's not to say that when you're a commercial business, it's just about making money. Money is the side effect of having a big impact, right? So you want to have an impact, you want to do something that people are going to use that is going to make a difference in the world. It's not just going to sit on a shelf in some library. So we shouldn't go looking for some intellectually interesting standalone security thing. We should see where there are important technologies being developed that have the potential for a huge impact but they aren't being adopted. One example today is cloud. The biggest reason for non-adoption of cloud today is security. So what can we do? And the people who are adopting it have to take unnecessary risk, I think, because it's just the way technology is built, right? You build the core functionality first, you worry about security later. There's no point in thinking you can change that. [Laughs.] It's the way it happens. CCN for us, security is mission-critical for us. Nobody would ever use a thing like that if security weren't built in at the fundamental, bottom, supporting foundation of the thing, right? So that's a good choice for a security group to work on something like that. Ubiquitous computing was another one, because if you can't solve security and privacy issues there, you're just asking for trouble. You can have all kinds of personal information being available to the

world, and as a society we don't want that. As individuals, we don't want that. And so it's a great choice, security's mission-critical there. And we should be looking for: where is security mission-critical? Where is it one of the top one, two, or three most important things and that without it, adoption either will not be possible or will be very limited. So that's how we can really multiply our impact. And, of course, there will be security applications like intrusion detection, virus detection, malware detection, those are all very important things, too. And there's standalone security things. Right now, I don't think it's a great place for PARC to be working; it's a very mature area. So those are the kinds of things; that's how I think about what's a good problem for a research group like PARC to be thinking about, and investments in security research and where we can have a big impact.

Yost: How closely is that tied to Xerox's current computer security business and to what degree is it just looking more broadly toward future applications in businesses that Xerox might get into?

Lunt: We're not Xerox PARC anymore. We are PARC; we are a subsidiary business. So, it's 2002 and Xerox created us as a subsidiary company. They had been paying for all of our operations up until that point. At that point they stopped doing that. They still pay for almost half, I would say, of the work that goes on at PARC, and part of that is contracted work, and part of that is more speculative work. Xerox is our customer now. They are our biggest customer. They still own PARC but we're a separate business and we do work for other companies. So we do work for lots of other commercial companies and also do

technology licensing. That's a significant part of our business. And we can also create spinouts, or license technology to spinouts, and so that's also a way that we can have impact. So there's many, many ways we can impact the world. You know, as a member of the security group, and then later leading the security group here at PARC, before this happened it was really challenging to see how we could have an impact on Xerox. They are not in the security business. Security is a ***requirement*** on many of their printers and office services, but not a big driver. It's not mission-critical. Many, many things are way more important. So we were always feeling like there was no good way for us to impact the company, but now it's different. Now we can find customers anywhere and we have some great relationships. Some of our first commercial projects were around security for various other commercial companies.

Yost: Finally, are there any topics that I didn't bring up that you'd like to discuss?

Lunt: I think you hit a lot of them.

Yost: Okay. Well, thank you so much. This has been really helpful.

Lunt: Great.