An Interview with

MATT BISHOP

OH 429

Conducted by Jeffrey R. Yost

on

6 June 2013

Computer Security History Project

Davis, California

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

Matt Bishop Interview

6 June 2013

Oral History 429

## Abstract

This interview with computer security pioneer Matt Bishop discusses his doctoral research (access controls and the Take-Grant Protection Model) working with Dorothy Denning at Purdue University and subsequent career as a computer scientist and computer security specialist at the Research Institute of Advanced Computer Science, on the faculty at Dartmouth University, and on the faculty at University of California-Davis. Bishop's research is wide-ranging and the interview touches on his work on Unix security and vulnerabilities, network security, intrusion detection, electronic voting systems, and other areas. Bishop recounts the project he launched to provide public (Web) access to seminal early papers in computer security, an important effort to facilitate computer security history and learning from the past.  And he relates the evolution of the UC-Davis Computer Security Lab and its influence on the research field and education in computer security. He also discusses his role as an educator and the authoring of his textbook *Computer Security: Art and Science* (2002).

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 University of Minnesota, the Charles Babbage Institute, and I'm here today at UC-Davis with Matt Bishop and this is for CBI's NSF-funded project, "Building an Infrastructure for a Computer Security History." I'll begin with just some basic biographical questions. Can you tell me when and where you were born?

Bishop: In New York, on September 29, 1956.

Yost: And did you grow up there as well?

Bishop: Until I was six, and then my family moved to San Francisco and thought it was a reasonable idea for a six-year-old to go with them. So I lived in San Francisco until I was about 12 or 13, then we moved to San Rafael.

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

Bishop: Oh my word. There were a couple of teachers. One of them was a Russian teacher named Boris Ilyin and a French teacher named Carole Crane, and a science teacher named Mrs. Hicks. Then, I went to summer school in the seventh grade, and took a class that introduced you a little bit to eighth grade math. It turned out I was fairly good at it, so the teacher had me for the eighth grade but let me run ahead of the class. By the time I was done, I hadn't gone through set theory, but I had done the regular algebra, algebra II, and trigonometry by the end of the first semester and the teacher did some special work with me. His name was Mr. O'Brien — I never knew his first name — but he was an absolutely wonderful teacher. I had basically science, math, and languages with these. And then in writing, there was also an English teacher who was also superb, Jane Boisseau.

Yost: With your aptitude for math, was it something you greatly enjoyed?

Bishop: Loved it. Basically, it addicted me.

Yost: You went to college at Cal-Berkeley.

Bishop: Yes.

Yost: Did you have any exposure to computers before Berkeley?

Bishop: Not beyond reading the occasional article in the science magazines, or wherever. The first exposure I had that I remember was in a physics program. I was an astronomy major; later on I changed to astronomy and applied math. In the upper division analytic mechanics class, in the first class, the instructor walked in and basically said, ladies and gentlemen, by the end of this quarter you will have written a three-dimensional simple harmonic oscillator or you will have failed the course. So we learned FORTRAN and they had a computer — I think it was an IBM 1608 but I'm not sure — in the basement of Birge Hall that we had to use. And then I did some work on it over the summer, as part of a summer project with another physics professor there, calculating trajectories and things like that. So I learned a little bit about how they worked that way.

Yost:  Did you enjoy working with computers?

Bishop:  Yes.

Yost:  You initially decided to major in astronomy, and later to add applied mathematics to do a double major?

Bishop:  Yes, what happened was that I was an astronomy major but I started falling in love with the mathematical tools astronomers were using, so I ended up shifting my interests to do both. The reaction from most of the astronomers was oh, you want to major in math, too? You want to be a tool maker? And I just think if they could only see me now, because that's what mathematicians often think of computer scientists.

Yost:  Did you give any thought to what you wanted to do career-wise when you were in college?

Bishop:  Professor or basically researcher of some kind. Or teacher. Or both.

Yost:  Did you take any courses in computer science at Berkeley?

Bishop: After I graduated, I went into the math program for two years as a master's student. And while I was there I took a course in programming PASCAL. That got me very much interested in it, so I did assembly language and a couple of other computer science courses as well, and by that time I was pretty much hooked.

Yost: And so it was by the latter stages of completing a master's degree in math that you knew that you wanted to follow on with staying in computing?

Bishop: Partly that, but I also did some work with a lawyer in San Anselmo and got interested in the law and privacy then, too. Security seemed to be the place where the law, mathematics, and computer science all met. So by the time I was finishing up my master's in math I knew I wanted to do work in computer security, which greatly limited where I could go.

Yost: What places did you take a look at?

Bishop: Purdue, because Dorothy Denning was there. I also looked at M.I.T. I don't remember other universities. I don't know if I even looked at Cal Tech because they didn't seem to focus on computer security much. There were a number of others that I don't remember.

Yost: Did you visit these schools before making a decision?

Bishop: I think I visited Purdue after they had admitted me, when I was trying to decide where to go, but I'm not 100 percent sure of that.

Yost: With M.I.T. were you aware of Multics?

Bishop: No. And I wasn't admitted to M.I.T., which was probably a blessing in disguise at the time.

Yost: You knew Dorothy Denning and or knew of her work?

Bishop: Oh no, I didn't know her, but when I was looking through the catalogue, it said one of her interests was computer security.

Yost: At that point you hadn't read her lattice model?

Bishop: No. Basically, I was a complete novice. At the time they didn't even have the computer science advanced GRE test; you took the math one. I basically just went down the catalogues and looked at people who seemed interesting and tried to pick out places from that. I wasn't aware of the literature or anything like that, at the time. The other thing that I remember is I did write a note asking a question, and Dorothy wrote me a very polite note in response.

Yost: Did you know that Peter also did some work in computer security, at that time?

Bishop: I know it now; I don’t know whether I knew it at the time. I think it was more general knowledge of, well, there is interest there; here’s a faculty member, let me write to her and ask her a couple questions. But Peter does a lot of things other than computer security and Dorothy seemed to focus more on that. My memory, by the way, is somewhat shaky because a) it was so long ago, and b) it’s so colored by what I’ve learned since then. So please bear in mind my answers may be completely off here and this may not be what I was thinking at the time. I don’t remember the application process very well. I just remember I got interested in computer security because of the intersection of the three subjects. I went looking and looked at a couple of places — another one I looked at was Yale University, by the way — and wound up, after very many peregrinations and such, at Purdue. You always wish you had a steel trap memory, but fortunately or unfortunately, I don’t. [laughs]

Yost: Yes, definitely. Can you talk about the curriculum you had in your Ph.D. program at Purdue?

Bishop: I had taken a couple of the classes elsewhere but the courses that I remember taking were computation theory under Paul Young. Computer security, of course, with Dorothy Denning. I did operating systems and compilers elsewhere, so I had those coming in. I had numerical analysis under Professor Walter Gautschi. I don’t remember the title of another theory class, it may have been a follow-on to the one Professor Young taught, by Professor Buchi; and I did analysis of algorithms under Michael O’Donnell.

Then I know I did a graduate course in mathematical statistics and another one; here, they would call them 289s. They're kind of group courses in a particular subject; topics courses would be a good way to put it. One with Doug Comer on building Xinu. I did quite a bit of work on Xinu along with my classmates. There was a class in programming languages, too, and I don't remember the teacher. I think it was James Morris but I could be way off on that. I think that one, aside from the security class, was the most fun because it was a comparative languages class. The way he did it was he had us learn five languages over the course of like 10 or 11 weeks, and then the last couple of weeks was just a comparison of languages. It was fun learning the languages.

Yost: It sounds like an interesting class.

Bishop: Yes. Usually it's taught, well, here are the theory and principles, and here are the languages that exemplify it, but he did it backwards, which was great.

Yost: Can you tell me how you came to your dissertation research topic?

Bishop: I was always very interested in systems building. And so, for example, when I was at Purdue, the first semester I was there I was a math TA because of my background in math and the math department was hurting for people. I found out that they also hired students to help out with the department computers so I talked to Doug Comer about it and mentioned it to Dorothy, and apparently she talked to Doug as well, so I got pulled into the group. I can't remember, it was initially on a volunteer basis, I think. But then the second semester — I was going to be a math TA for the first year fall and second year fall — so the second semester Doug picked me up as a paid systems administrator type so I got very interested in UNIX protection, it was a UNIX system. Dorothy had me doing database work and for whatever reason, I didn't like it, but I stuck with it. I think I started on that project a second year and after about a semester or two — and I don't remember the exact time frame here — she basically sat me down and said look Matt, you're having a lot of fun with systems work, you love doing the Take-Grant model. Larry Snyder was the professor there. I had known him before I came to Purdue and I had done a project with him on the Take-Grant model, where we had extended it in a certain way. It's like, oh gee, here's an idea, Larry can we talk about this? And we did; and I got a paper out of it. Dorothy said, you really like that work. Why don't you forget the database work and try applying the Take-Grant model to the UNIX system and see what you come up with. I pointed out that I'd have to extend it in some ways and she pointed out, yes, that's your topic; that would be a good Ph.D. topic. That taught me that Dorothy was an incredible advisor because among other things, what that showed me was, look, follow your interests in research because that's where you really learn things and it becomes much more fun. And I wound up turning it into a Ph.D. thesis.

Yost: You partially anticipated my next question, which is to describe Dorothy as a mentor.

Bishop: I can't think of enough good things to say about here. She was very supportive, a very, very good advisor, and absolutely brilliant at analyzing things or critiquing things that I was doing, and very good at leading me through paths that I probably would've tripped over without her. Very supportive. I don't know what more to say.

Yost: And was Snyder . . .

Bishop: Larry Snyder, yes.

Yost: . . . was he also on your dissertation committee?

Bishop: Yes.

Yost: And Peter Denning as well?

Bishop: Yes.

Yost: Can you speak about the contribution and mentoring of those two?

Bishop: I had much more interaction with Dorothy and Larry than with Peter. Larry also was absolutely fantastic. I was at another university for a year before I went to Purdue. And Larry was my advisor there, and he was absolutely superb. The first paper that we wrote we submitted to SOSP, and basically, they were requesting changes and things like that, and he got quite upset about it. The paper finally made it. It turns out the conference was at Asilomar [Pacific Grove, CA], and since I lived in Marin, it was easy to get to.

Larry basically said you are not presenting this paper because it's going to be very controversial and I don't want you getting the challenges that are going to come. And boy was he right. The second paper, which was on the same topic, I did end up presenting and this will give you an idea of Dorothy's style, which, by the way, I absolutely love. I presented it and I got a couple of questions, which were mostly friendly. I remember Roger Schell was one of the people that asked the questions, but I don't remember the question. And at the end, when no one else was going to ask a question, Dorothy gets up and says something to the effect of, in your talk you quoted, look before you leap, she said, but what about he who hesitates is lost? [Laughs.] It's like yes, okay. I don't remember my reaction externally, but internally, I loved it.

Yost: So was your dissertation was the first attempt to apply the Take-Grant protection model to a system for security?

Bishop: I wouldn't say first. Anita Jones applied, if I remember correctly, the model to the Hydra system and for whatever reason, that work didn't go any farther. I thought what I'd try doing is modifying the model to handle certain characteristics within UNIX-like groups and setUID ID privileges, and things like that. As far as I know, I was the first one to apply it to UNIX. I think maybe the second to apply to a system, but Anita was definitely there before me. In fact, I remember reading her paper when I was starting down that path.

Yost: Certainly, applying it to UNIX is something that creates much broader applicability.

Bishop: UNIX is much more widely used than Hydra, so in that sense, certainly. [Laughs.]

Yost: So you presented and published from this in 1981?

Bishop: Larry Snyder and I did the first paper in 1979, and the second one was in 1981. I don't remember whether it was the sixth and seventh, or seventh and eighth, or eighth and ninth SOSP, but the years are right.

Yost: And the conference presentation that you were referring to earlier, was that the 1979 or 1981?

Bishop: The one where Larry said I don't want you to present the paper was the 1979 one. But I was at the conference and that's where I met Anita Jones and a number of other people.

Yost: When you presented in 1981 at the Symposium on Operating Systems Principles, how was this research received?

Bishop: It seemed to be received very well. Some of the people that I remember talking to liked it quite a bit. I'm sure other people didn't but I don't remember anybody in that discussion. I don't remember whether it was in 1979 or 1981, but Anita Jones and I had a bit of a talk about it because she was one of the creators of the model and she really encouraged me to keep going in that area. She's another one who is superb.

Yost: Did you learn anything from her work with the application to Hydra that was useful to you?

Bishop: I honestly don't remember. Let me rephrase that. I'm sure I did but if you asked me what, I couldn't tell you.

Yost: As this research was disseminated outside of the people on your committee, are there people that stand out as strong supporters or backers of the approach you took?

Bishop: Not really. What I was doing that got more interest was looking at trying to find vulnerabilities in UNIX. That's part of what I was doing with the model, but also, a lot of it was me just studying the system, looking at what others had found, and trying to channelize it, and so forth. And that got a lot more interest than the actual modeling.

Yost: Can you elaborate on the vulnerabilities that you found?

Bishop: In 1980 or 1981, I don't remember which, I was playing around with the system and I found some problems. And then I found others who had found similar problems, so I ended up writing them up and showed it to Dorothy and basically said well, what do you think? And she said well, this is pretty good, but what would be even better is if you could try to organize them in some way and see if you can find ways to fix them. So I ended up writing a paper in 1981, which was never published, which basically described how to attack the UNIX system. The reaction was rather interesting from people, because people who knew Dorothy and who knew I was her student thought it was very good. One person, who I don't want to name because he felt very badly when he found out what was going on, I sent this to him but I forgot to identify myself as Dorothy's student. I just said I'm working with the UNIX system, here are some things I found; I value your opinion on this. And I got back a very, very strong note basically telling me to get out of the field and leave it to professionals because amateurs shouldn't be breaking systems. I showed the note to Dorothy and she said I'll handle this. After about four weeks I got a very nice note from the person apologizing and saying I didn't realize you were working with Dorothy or I would never have done that; this is really good work; we need more people like you; and so forth. I have no idea what the conversation was like between Dorothy and this person but whatever it was, it certainly resolved any problems. I know the person now, but since then, I'd never reminded him of this because number one, there's no point in doing it, and he's always been very friendly and very helpful.

Yost: But clearly, Dorothy commanded much respect in the field and by that time was very influential.

Bishop: Exactly. I also found out I'd made one mistake because I labeled the paper confidential, meaning don't spread it around without my permission, and I kept numbered copies and everything like that. I sent one to Roger Schell, and I saw him at a conference and asked him what he thought of the paper. He said, well, the confidential is not a military designation is it? I said no. He said okay, good; then I loved it. [Laughs.] The paper, by the way, since then has been published. I believe it was about 2009, I was asked to give a talk at ACSAC [Annual Computer Security Applications Conference], a retrospective, and I used that paper as the basis for it. "Classic work" was what they called it. And basically everything on there, at least on current systems, we tried again and the only ones that worked were the administrative ones because somebody didn't follow the directions on how to configure the system. So I didn't have many qualms about using it. Also, I talked to people whose opinion I really respect, and their comment was of course you should publish that. That should've been published a long time ago.

Yost: While you were in graduate school, you obviously attended the operating systems principles conference. Were you also attending the Oakland one?

Bishop: I went to Oakland whenever I could.

Yost: Do you recall what was the first year you went?

Bishop: 1981.

Yost: So the second one?

Bishop: Yes.

Yost: Can you describe that conference and the atmosphere and the culture of the conference?

Bishop: Very small at the time. I can't remember how many people were there but I think it was under 100. Very, very collegial. People generally knew each other and those who didn't were very quickly introduced. As I recall, I met Jim Anderson there, Roger Schell, and a number of others. And the papers that were presented, people were very constructive in their comments on their papers. I don't remember any attempts to tear anything down. I remember a lot of good natured teasing. At the time, people used overlays. Dorothy had nine overlays on one talk and I remember her being crowned as queen of the overlays because that was more than anyone else had ever done. [Laughs.] The papers also were pretty much breaking new ground because the field was so new at the time. And so a lot of the conversation in the hallways and such was well, how do we push this work farther? What are the implications? Here's a project that we're getting going on, would you like to work with us on this or how would you suggest we proceed? That sort of thing. That was a very, very exciting time.

Yost: You mentioned Roger Schell, as well as Jim Anderson. These individuals were involved with important work in the 1970s, the Anderson Committee, and the follow-on work of Roger's Air Force research program on computer security. At that time, were you aware of the Anderson Report?

Bishop: At that time, no. Not in the 1970s.

Yost: But when you first met these people at the symposium, were you aware of their background, what they had done?

Bishop: Oh, okay. Certainly knew about Roger's MULTICS work. That was very widely known and I did know about the Anderson report. And I did know about some other work that Jim Anderson had done that suggested using auditing mechanisms, that were back then used mainly for accounting, as tools for doing security analysis for basically what would later be called intrusion detection. So, yes, I knew of their work.

Yost: In studying what was done with MULTICS, did that provide any insights to you in looking at vulnerabilities with UNIX?

Bishop: In particular, there were a couple of studies that were done that included MULTICS, and those certainly provided inspiration. The MULTICS work itself did, as well, but sort of from the other way, as in how do we try to do it right? The idea is knowing where you want to go, and then trying to go there, as opposed to just building it and then saying okay, what can we do with this? I don’t recall any specific vulnerabilities in MULTICS that inspired anything but I recall reading the final Protection Analysis report — don’t remember which one. Anyway, what it did was it showed how to build generic descriptions of vulnerabilities and it turned out that I guess it had a bigger influence than I realized because part of my research now is looking at vulnerabilities in a certain way. And when I described it to Marv [Schaefer] many years ago he said oh, that’s what the Protection Analysis guys were doing too. And I thought I should go back and reread that report. When I reread it I realized, what I’m doing is a little bit different, but certainly the nub of the idea is there. I wasn’t conscious of it when I thought of the approach because I was doing the work in intrusion detection and the approach is derived from that. But on the other hand, it may well have come from that.

Yost: Were there any other graduate students at Purdue that were there at the same time as you that started to work on computer security?

Bishop: Jeremy Epstein is the one I remember.

Yost: Was he a student of Dorothy’s?

Bishop: I don’t remember who his advisor was. I do know that I was the only Ph.D. that Dorothy graduated at Purdue but she may have advised at the master’s level. There were a few others but they were more on the systems end, and they did security kind of when it impacted their area. For example, Paul McNabb, I don't remember whether at the time he was in security at Purdue, but he's certainly done a lot of security since then.

Yost: During parts of 1982, 1983, and 1984, while you were in the doctoral program at Purdue, you worked as a systems programmer for Mega Test Corporation?

Bishop: Yes, during the summer. And then the last year I was working there part time while finishing up my thesis.

Yost: Can you talk about both that corporation, provide a brief description, and also the type of work you did?

Bishop: We built VLSI testers. And they were trying to build an advanced tester that used UNIX to control the test equipment. And, let's see, I got the summer job there because one of the assistant programmers at Purdue, Steve Stone, moved out to the West Coast and got a job there, and they were looking for interns. I don't remember whether I approached him or he approached me, but in any case I got hooked up with them. I was there for two summers and then I moved. Dorothy went to SRI or DEC. I don't remember; it was one of the two. They [Peter and Dorothy] were both in Palo Alto or Menlo Park, around that area. Peter became the director of RIACS so there wasn't much point for me in sticking around Purdue, and I also had personal reasons for wanting to come back to California. So I got a job. Basically MegaTest offered me, I can't remember what percentage it was, but it was a job where I was also told, yes, you can work on your thesis and use company equipment to do that. MegaTest was a small startup. At one point, someone was interviewing a new employee for TV, and he asked the new employee what he thought of the president of the company. He shrugged and said my T-shirt's nicer than his T-shirt. So that'll give you an idea. Its philosophy was very much as long as you're contributing, have fun. I did a number of things involved with keeping the systems running. I basically built a driver for them. They had a laser printer but there was no driver for it yet so I wrote the driver, which was great because somebody in the sales department, her husband was finishing up his Ph.D. or master's thesis, and so they were helping me debug it. Again, all this was with full knowledge and approval of the company. The company basically wanted to get that driver working so they said go for it. Then I spent a lot of time building a pattern matching assembler which would be driving the outside tester. And in order to test it, I had to do a disassembler in order to make sure the codes were being assembled correctly. And while I was doing that, a friend of mine across the cubbyhole was having problems. So I took the assembler and I built it in a modular fashion. I changed a couple of things, changed the instructions. We were able to run it on the VAX and disassemble things. Then he was having trouble reading the core file so I wrote a program called Core Dump that would take your core dump and print it out in an understandable form, including memory mapping and contents of registers. He had been trying to fix something for about three weeks that he just couldn't capture the debuggers. He ran that and within two minutes he had found the problem. It was just that the debuggers were tampering with the registers in such a way that he couldn't see where something was pointing. So I did a lot of that sort of thing, Yost: Sounds like both interesting work and also a great environment to continue on your studies.

Bishop: Yes, it was fun. Some of the people were also university types so we would talk about fairly advanced ideas and how they might apply it to what they were doing. Also, everyone understood I was working on my thesis and they were very supportive. When they found bugs they would let me know and I would see if my model could catch it; if I could incorporate that somehow into my thesis, that sort of thing.

Yost: Was MegaTest in the Bay Area?

Bishop: When I got a job, it was in Sunnyvale, and it subsequently moved to San Jose.

Yost: Can you talk about your job search, after you defended your Ph.D. dissertation?

Bishop: I went back to MegaTest and wanted to do research. So they made me the offer of you can work for us half time and we will support the research for the other half. But I decided I really wanted to do it full time. So I talked to Peter at RIACS, and they were very interested in hiring me. We were able to work things out and so I went up to RIACS and worked there for three years. There were a couple of people at RIACS, Mike Raugh and Barry Leiner, both of whom very strongly encouraged me to become a teacher because, they said, you'd be great at it. In 1986 I got married, and the lady I married had a 12-year-old daughter. We decided that it would be good for me to teach but the environment where the daughter would go to school if we stayed, would not be conducive to study. So I wound up applying for a university job at a number of places on the East Coast. Jeepers, I don't remember all of them, but one of them was Dartmouth. Dartmouth made me an offer and we went out there and looked at it, and liked it, and decided to go. So I was at Dartmouth for six years. Let me emphasize that I did not apply to anything in California and in a way, I regret that. Had I applied here, I don't know whether or not I would've been hired, but if so, it would've been very interesting.

Yost: You mentioned RIACS.

Bishop: Research Institute for Advanced Computer Science at NASA Ames Research Center.

Yost: Right. Can you talk about that organization? What was the atmosphere when you arrived?

Bishop: When I first arrived I was doing mostly computer graphics for various reasons. The organization was basically your typical research lab. I wound up doing a fair amount of system administration too, because I knew how. I'd been doing it at Purdue and I was quite comfortable with it. RIACS was basically a contractor. It was run by the University Space Research Association. I was there [and] I did computer graphics and such for a year. And then some of the people who were working on the new supercomputing center found out I was there and said would you mind helping us a little bit with security? So I did mostly computer graphics the second year but some security. Then the third year I think it was about balanced evenly, maybe a little bit more security. And it worked out very well. The atmosphere, at least from my point of view, was quite friendly and helpful. The only issue I can remember involved bureaucracy. Because it was working with the federal government there were all sorts of bizarre rules. I think at one point, I was on study group and we wanted to host a meeting at Ames, but it turned out that there was absolutely no legal way to pay for the sandwiches in the breaks. The office staff came up with a way — I do not remember what it was but I wish I did — but it worked. The other thing I remember administratively was at one point, everybody was taking off for secretary's day and what we were told was you can have one alcoholic drink at the expense of the government. I said what about those of us who don't drink? They said you have to pay. The administrator had it exactly backwards so a number of us tried to convince him that really, the milk was from a drunk cow. It didn't work. I have no idea how that one got resolved. [Laughs.] The work environment itself; the people were delightful and were very, very supportive of each other. The only disagreements that I remember were typically professional.

Yost: How large an institute was it?

Bishop: When I got there, there were about seven or eight people. I don't remember how big it grew. There were a number of us from Purdue; Peter Denning, as I mentioned; Bob Brown, George Adams, Dave Curry I think was there for a little bit of time, too. Those are the ones who I remember the best because I knew them for years.

Yost: And were there others with computer security interests besides yourself? Other than Peter.

Bishop: Of course, Peter. Bob had a little bit but not that much. When Dave Curry came, he was quite interested in it. That's really all I remember. Most of the security work I did was with the supercomputing folks and a lot of it was operational.

Yost: When you say RIACS was a contractor doing various contractual work for different customers other than government?

Bishop: Well, for NASA Ames.

Yost: For NASA Ames only.

Bishop: Let me rephrase that. As far as I am aware, for NASA Ames only. Certainly all my work was funded by NASA.

Yost: And was it an environment where NASA Ames put forth what work they needed or were the researchers at the institute proposing projects?

Bishop: I think it was both ways. I tried to stay out of the management as much as I could, which was very easy since Peter was such a good director. Peter handled all that.

Yost: As I looked at a number of your publications in the first half decade after your dissertation, I was struck by the range of your work. You seemed to be developing many different interests. Can you talk a bit about how those research interests evolved?

Bishop: I'm not sure I can because I'm not sure they've ever evolved into anything. I'm still quite eclectic. For example, a lot of the work that I've been doing lately is on vulnerabilities analysis. Some of it is on the insider problem, which is really a form of vulnerability analysis. But I've also been working on attribution on the internet, and many years ago a colleague and I did a paper on how you distinguish between — I'm going to call them cyber attacks although I don't like the term — between cyber attacks launched by individuals and small groups, and cyber attacks launched by nation-states or their equivalents. I'm just fascinated by the whole field and I tend to go where my interests are. The people who I work with, if they're working on a particular topic, I'll often dive right into it with them. That way, I learn an incredible amount. Hopefully, I'll retain it.

Yost: I asked you about the IEEE Security and Privacy Symposium. Were there other events? Did you go to the National Computer Security Conference, as a graduate student?

Bishop: As a graduate student, no, I don't remember going to that one. I do remember going to the National Computer Security Conference when I was at Dartmouth, and when

I was here. I don’t remember if I made all of them, but I surely made most of them from that point on.

Yost:  Can you compare and contrast the environment of that conference with the IEEE Symposium?

Bishop:  That one was much larger and much more oriented towards government and industry. In other words, there was some academic work there but there was also an awful lot of operational here’s-how-you-do-something. So, for example, one of the papers I had there discussed how you build an isolated network for testing really malicious stuff. And I’m not sure if that would’ve been an appropriate paper for Oakland because the paper was extremely operational, as in here’s how you build switches, here’s how you configure the network, and so forth. For an Oakland-type paper, which I probably should’ve written — now, of course, it’s far too late — it would’ve been, here’s how you model the escalated networks, under what conditions you break isolation, and so forth.

Yost:  So, broadening it out to the theory behind it.

Bishop:  Right. Or narrowing it down, depending on your point of view.

Yost:  Right. When you went to Dartmouth, were you the only faculty member interested in computer security?

Bishop: Oh, yes.

Yost: It was probably a fairly small department?

Bishop: I was in the Department of Mathematics and Computer Science, so there were a lot of mathematicians and a few computer scientists. And at the time, they were really pleased to have me because, they said, you really enjoy the systems. The thrust of some of the comments at the time was that computer security is not an academic topic, which was both a little bit exhilarating because how do you show it's academic, what academic things can come from it, and so forth; but also, a little bit unnerving because one of the issues within the department was it was so heavily focused on the mathematical end that the problems that arise in practice were basically [treated] like well, that's just details. The problem is in computer security it's all details. You can build the most beautiful mathematical theory but if that theory doesn't map exactly into what you're doing, the little details can kill you. So security, to a degree, has to be operational; it has to be focused on what's actually there so that you can build a model *from* it, instead of building a model and then trying to apply it and close the gaps. You can do that too, and it is done; it's just that in my experience it's much harder to close those gaps. So anyway, there wasn't anyone else at Dartmouth who was particularly interested in security during my six years there.

Yost: This would be moving back a few years prior to that move, but The Orange Book comes out, how did you view that at the time, both in terms of was this the right approach and what did you see as the impact of that document?

Bishop: At the time, I didn't think of it in that way. I looked at it and thought that it was a good step forward because it was actually trying to codify things and, of course, whether or not it would be useful depended on the policy. I do at one point recall seeing a draft of the network version of it and this is where they took those things and tried to apply it, and I didn't think that part was well done. The one that they *did* put out was quite well done, but I remember that the draft that I saw was not that good. Also, Oakland and a number of other conferences, one of the issues is that many commercial firms, in particular banks, didn't like the idea of multi-level security because their approaches to security are completely different. Within the government it was very good. It was absolutely necessary to try to codify some of this and get companies interested in building the systems. And I think the key thing that The Orange Book brought about is not the idea of multi-level security, but the idea of assurance — that the companies had to show that their systems did what they were supposed to do. And that certainly carried over to what commercial firms were interested in. So that part of The Orange Book was fantastic.

Yost: Willis Ware and the 1970 Defense Science Board Report really emphasized the importance of computer security research taking place in an open environment, sharing ideas and partnering with industry. He really stressed that industry would have to play a critical role with the great challenges with multi-level security if they were to be met. And then this appears as the start of a trajectory of Schell's Air Force research, and Bell-LaPadula, and then The Orange Book. The goal with The Orange Book clearly seems to be that incentives, certifications of systems, must created for industry but

Bishop: C2 by 1992.

Yost: . . . but relatively few companies developed high assurance systems . . .

Bishop: Still true.

Yost: . . . and in that sense, were the financial incentives just not there?

Bishop: You should ask an economist, really, I don't know. If I had to speculate — I need to emphasize that this is pure speculation on my part — I would guess that the financial incentives weren't there in the sense that it cost a lot more money to go through the certification procedure. Even though you didn't pay for it, it would take a lot of time to craft the system in such a way that it could get a fairly good rating. And then you sell it to a fairly limited market. I mean, your average consumer, your average company, wouldn't need the multi-level security stuff. If the home PCs and such started coming out, well, why aim for the government at all? Aim for the hobbyists. So I suspect that economically, that was part of the problem; that the economics were a large part of the problem. The other part, too, was probably the question, well, why do we need this? I suspect that was being asked in government as well. Managers would say well, we're required to do it. Okay, we'll go ahead and do it. But now we have to hire people who actually know how to administer the stuff. That drives up the costs again. So, ultimately, it's the resources, because now you're using people for more complex tasks than administering an ordinary system. It wouldn't surprise me if that was not part of the issue as well. That being said, I want to re-emphasize I do not know the answer, I am not an economist. All I know is that to put it bluntly, it didn't work. I do know that the goal was everything in government was to be at level C2 by 1992. But people kept asking for exceptions for various reasons, and as a result, it never quite made it.

[BREAK]

Yost: And C2 is fairly minimal security.

Bishop: Basically, object reuse, and discretionary access control, and protection of authentication data.

Yost: So, in terms of the government's classified information structure, that seems insufficient, doesn't it?

Bishop: I would assume so. I should tell you here now that I have no security clearance whatsoever. So I know the rules for handling unclassified material and that's about it. I've been on a couple of study groups where we got something that was unclassified but sensitive. And on those, basically, as soon as the study finished I destroyed the media they were on or I deliberately caused the IronKey to wipe itself. But beyond that, I don’t know the rules but it does seem to me that C2 would be awful weak for anything very sensitive. Of course probably what they use now is awful weak anyway.

Yost: IBM had a research program in computer security with much of it devoted to cryptography.

Bishop: Yes, at Watson.

Yost: But IBM Research also, in the early 1970s, did some access control research that they presented to customers at the IBM SHARE meetings in 1974 and they spun a off product, after they got feedback, that became RACF, and then a competing product that ACF2 grew out of this, and later one called Top Secret. As an academic researcher in computer security and with a lot of knowledge of research to theorize and develop truly secure or far more secure systems, how did you view what was going on commercially?

Bishop: That was before my time so basically, I had no view. In 1974, I think I was a first year Berkeley student.

Yost: But RACF still exists today.

Bishop: Oh, how do I view it now?

Yost: Well, how did you view it when you first learned about it, would that have been the 1980s?

Bishop: Cool, people care. [Laughs.] I don’t remember when I first learned about it, probably in the 1980s. And it was good to see commercial firms doing research and trying to build products that would improve security. My view is a little bit weird. The Orange Book — and actually, I do remember hearing about some of that work — I didn’t know it in the guise of The Orange Book, because The Orange Book hadn’t been done, but I do remember a lot of interest in how to build secure systems at Purdue. And my question was always well, what about the existing systems, you’re not just going to be able to get rid of them. So my interests have always been much more on how we deal with existing systems and vulnerabilities. Although I’m very much interested in building assured systems, on the other hand, in the 1980s there was this feeling that we could just solve all the problems by building high assurance systems and then we’ll be able to get rid of all the stuff that doesn’t work. RACF is essentially an admission that no, we’re not going to be able to get rid of all the stuff that doesn’t work. We’re going to have to try to build security into it or onto it. So when I heard about it; I really appreciated that point of view. That I think is a more complete answer.

Yost: The other point of view, that seemed to more typical of responses I was getting from Roger Schell, he seemed to see the only answer as building a secure systems.

Bishop:  Don’t misunderstand; I agree completely with Roger. My only point is, in terms of deploying those systems to replace existing ones, in the government, no problem. You can probably just do that by an order. But in industry, you’re going to have to make a business case and, in most cases, you’ll have trouble succeeding. In home computing, you will not succeed simply because those systems are going to be more expensive and people will want the cheaper one. In fact, one of the questions is how do we balance security and usability? And the problem is that most people will go for the added features over the added security. So I agree completely with Roger that the goal should be to build systems that are secure, that meet the specific security policies that you have in mind. For various reasons, I don’t think that deploying only those systems is practical. The other problem is, of course, what about different policies? RACF implements a particular access control policy or set of policies. What happens if those aren’t the ones I’m interested in? Do I still have to install it and still have to use it? Or what if you build a multi-level secure system that’s very high assurance, but I don’t care about confidentiality; I care about bank account and transactional integrity. That system is not going to help me with transactional integrity. Do I still have to buy it and then modify it to meet my needs? The moment I start modifying it, it will probably introduce problems. And so you’d have to build another system, which again would be the right thing to do. But now you’re starting to get a lot of systems — very different — that are high assurance. If you can do that, I’m all behind it; I think that’s the way it should be done. The problem is that in my 30-something years, my experience tells me that that simply will not fly in the world today, unfortunately.

Yost: Well I think, from what I've studied of what happened at that 1974 SHARE meeting is that IBM was receptive to more robust security in the product, if customers were willing to pay for it.

Bishop: Right.

Yost: And what they heard from sellers, that the market, which at that time was still mainframes and minis, customers simply weren't going to pay for it.

Bishop: Right. That's pretty much true today also. Most people see security as an add-on that either slows things down or hinders their work, and that's the mindset that needs to be overcome to facilitate widespread deployment of secure systems or high assurance systems. In areas where they're critical, like for the Space Shuttle and so forth, those systems are very high assurance. But in, for example, the average Windows, Apple, pick your favorite system so to speak, the assurance there that I've seen is not particularly high. In order to make it high, the data companies would basically have to revamp the inside of their system. I remember at one conference when somebody from Microsoft was giving the keynote, and he said something to the effect of we've cut the number of lines of kernel code in Windows 2000 down from 47 million to 33 million. I thought back to about 10 years earlier when Keith Bostic was giving a presentation about security in the UNIX BSD system, contrasting that to UNIX System V. He said System V is 1.5 million lines of code. One point five million. You want to bet I can't find a bug in 1.5 million lines of code? His whole point was that the system was just too large to be able to make the claim that it was secure. It would approach being secure but it would never be completely secure. Contrast that to the Microsoft person who was saying, in the interest of security, we've reduced it to 33 million lines of code, the implication being that that was more secure than the 47. And I was thinking okay, 33 is about 22 times larger than the 1.5. [Laughs.] But those product lines were necessary for what Microsoft was selling; what people either wanted to buy or once they bought it, found it actually useful. How can security compete with that? It's going to take instances — large corporations getting there is one thing, because they can afford extra security — I hate to say this, but I suspect it will take a lot of ordinary people getting hurt very, very badly in order for anything to really happen about security. And it's not clear to me that training people will help. My dad could barely use a computer. He was a writer. He used a computer like a typewriter. If you asked him have you installed your antivirus, he would say, what? You explain to him and he goes, why would I care about that? I finish the book, I print it out, if it gets wiped, I just retype it. He can type 120 words a minute, just [whoosh].

Yost: The U.S. has been less proactive with privacy legislation than many countries in Europe. You said privacy was one of your interests.

Bishop: Yes.

Yost: Is that an added hurdle culturally? Do Americans care less about privacy?

Bishop: I honestly don't know, I'm not that much of a cultural anthropologist. For whatever it's worth, just from my own observations. Again let me emphasize this is just very personal stuff. I'm not sure whether it's that they care less; I think it's caring differently. The loss of privacy gives you a lot of extra things, like when you go to the store and you swipe your card, you get a discount. But they also have a record of what you buy and so there's an offset, there's compensation there. I suspect most people feel, who cares? Who's going to see that, and so forth. That said, also I think as times change the notion of privacy changes. For example, my daughter's generation — she's 21 — she and friends post things to facebook or public media that I would never post because I feel it's too much of an invasion and would reveal too much private information. You see stories of people who were turned down for teaching jobs or such because the recruiter or hiring office will google them on the web and it turns out that they have a picture of themselves naked or drunk or something, and so they get turned down. Again, that's kind of a clash of privacy because the older generation will say gee, we don't want a person like that teaching. The younger generation says well look, that's just being human, it's on your own time, and so forth, so who cares? So the notions of privacy are changing.

Yost: There's also the problem that it's not always that person posting. Nearly everyone has a cell phone camera.

Bishop: Right. Also, it may be a misidentification. As an example, I know of one person, when you google his name, the first thing that comes up is a porn actor. The second thing that comes up is the right person. In another case, there was a very famous case where, I believe it was McCarthy — I can't remember if it was HUAC or the Senate committee — but one of them identified a worker at the Pentagon as a communist. And so, of course, big hearings and such, and after about 10 minutes into the hearing, it was very, very clear this woman had never ever heard of Karl Marx. It turned out that she had been fired as soon as she had been subpoenaed by the committee. And finally Symington — so I guess it was the Senate — took over and basically said, what do you do? And she said I work in the cafeteria. Do you have any access to top secret information? No, just the menus and the recipes. And basically what he ended up telling her was look, if you can't get your job back, see me and I'll see what I can do. Basically, they identified the wrong person. So one of the things that I am concerned about with the loss of privacy whether it's really your loss or someone who's a lot like you being lost? And if you don't know, that takes a lot of time to correct. That said, some European laws go overboard, like the right to forget, the right to oblivion. That's a great idea. I mean, everybody — if you'll pardon the crudity — everybody screws up when they were a kid. And to hold that over your head when you're 20, 30, 40 years old is absurd. On the other hand, you have to accept that this stuff will be on the internet and to say it has to be removed from the internet, that's not going to happen. Much better would be something to the effect of you can't use it, I don't know how that would be done, but some recognition that look, we're not going to try to obliterate, we're not going to use the — have you ever read a story called "Paycheck" by Philip K. Dick?

Yost: No.

Bishop: At one point there's a memory wiping device that can erase people's memory. You're not going to have that. If you've never read the story, by the way, I strongly recommend it. You know the writer Philip K. Dick, the science fiction writer?

Yost: Yes, I know of him, but haven't read his work.

Bishop: It's basically a story about when little things you least expect would be useful prove to be very useful. So anyway, the idea of privacy I think is critical. Again, I'm somewhat strange because most people who got into security because of privacy go into the crypto realm whereas I went into the systems realm. So I was a little bit strange.

Yost: Those realms were quite distinct, especially at NSA, COMPUSEC, and COMSEC, I've heard from Becky Bace and others.

Bishop: So I've been told.

Yost: But are those research communities coming together more in recent years as computation and communications technologies have become increasingly integrated?

Bishop: My impression is that they are. Systems people are finding crypto a valuable tool and crypto people are finding the constraints of developing crypto systems that can actually be useful, are also quite a challenge. Dan Boneh and Matt Franklin — I know Matt and Dan — recently got the Gödel Prize for developing practical identity-based encryption. And that's an example of something which is really cool theoretically, but also has immediate practical uses. They may well have come up without the incentive of how do we do public key distribution easily? But the public key distribution structures basically don't work well for reasons I don't want to get into. The identity-based ones seem to have the potential of working a whole lot better.

Yost: When you were at Dartmouth did you teach courses on computer security?

Bishop: Yes, and cryptography.

Yost: Can you talk about the courses that you taught and your philosophy of how to get these ideas across to undergraduate students?

Bishop: Undergraduates and graduates, they were mixed classes. Basically, I used Dorothy's book and followed her guideline, following the way she taught me. The one course where I didn't do that I co-taught with Jeff Shallit, I think he's at Waterloo [Ontario] now. Anyway, he was a computer scientist who did a lot of cryptography and his crypto was the modern public key cryptography. And while I like that, I've always been very fascinated by statistical attacks and classical cryptography. So I taught classical cryptography for the first five weeks, then he did modern cryptography for the next five weeks of the 10-week term. I learned an awful lot from him, from his part. And I think the students liked the course. I probably should have focused on the statistics a little bit more on my part. But anyway, that's how it went. I don't remember that much about how

I taught the undergraduate classes, the regular computer security class. I have a vague memory of projects, but if you asked me what any of them were, I could not tell you. They may have been research reports, or things like that. But I essentially followed what Dorothy had done, and I had them do a lot of reading of research papers, too.

Yost: I notice that you had some grants from the institute at NASA Ames in the Dartmouth years. Was that carrying over from research or did you [pause]

Bishop: Some of them were new. One of them from Dartmouth that was new was to look at auditing, trying to find some basic principles about auditing. One from NASA carried over very directly. I don't remember the other grants. I'd have to look at my CV. But there were some new ones. At least one I know I brought with me when I came here.

Yost: So there were some outside researchers that applied for grants to the institute?

Bishop: I'm sorry, what institute?

Yost: The NASA Ames Institute, RIACS.

Bishop: Oh. Maybe, maybe not. I don't know. The grants I got from NASA Ames all went through RIACS. Oh, but then after that, I was at Dartmouth. Yes, I did apply— I remember there was one grant on password protection and such that I remember. I don't know whether outside researchers got things. This was for some very specific work that I did earlier but it took it in a couple of new and interesting directions. It also got me my first taste of patents. Basically, somebody called and asked for a description of what I was doing so I sent them a technical report. Next thing I know I received a certified letter saying that they had applied for a patent in exactly what I was doing, and if they got it their lawyers would instruct me to stop working in the area. I have no idea whether or not he got it; I never heard from him again.

Yost: In 1993, you moved from Dartmouth to take a faculty position here at UC Davis. Can you talk about that decision and that change?

Bishop: My wife was born in California; I lived most of my life in California. In New England, while Dartmouth was a wonderful place to be, we felt that we weren't as comfortable as we were in California. I had known Karl Levitt since the mid-1980s and he was kind enough to invite me to take a sabbatical out here for a term, and that went very, very well. And then something opened up at Davis, so I applied and was hired. The chairman of the Dartmouth math department, an absolutely wonderful, wonderful person, basically said yes, you should take it. You are definitely not doing winters; you're definitely Californians. And he and his wife spent winters in Santa Rosa so he knew both places. [Laughs.] This was Professor, Ken Bogart. So then we came out here, I think we got here the third or fourth of July, and we've been here ever since.

Yost: And was the center also an incentive, a center that Karl had started?

Bishop: That Karl was here was a major incentive because it was a very strong academic program that he had started and quite frankly, I was just honored to be able to contribute to it.

Yost: You and Karl have partnered on a number of research projects. Can you talk about both Karl as a research colleague, and also some of the research projects that you've done together?

Bishop: A lot of the research has been on intrusion detection or on assurance. For example, property-based testing where you basically define certain properties that you want the program to meet and you define them in a particular language. That language can then be translated into code that is added to the program, so as the program runs it generates outputs. After the program runs, you then have a second engine that takes in the description, takes in the trace of the program, runs the trace through the description, and if it finds any violations it will flag those. If you do this right you can actually get the line numbers of the program where the problem occurred. We also collaborated on intrusion detection. We've collaborated on networking projects, pretty much everything under the sun. And, let me put it this way, although he was not my graduate advisor, if he had been, he would have equaled Dorothy Denning. I have never had a bad experience working with Karl. In fact, the only experiences I've ever had are good. And also, he's one of the sharpest people I've met anywhere.

Yost: I greatly enjoyed the morning I spent with him and learning more of his research and career.

Bishop:  He’s also very, very good at seeing the heart of the problem, which is a very rare facility. I mean, given a lot of information, it often takes quite a while to figure out the one or two key parts and he’s very good. One of his other students basically said that if I can get Karl for 20 minutes, a lot of things that are very nebulous, suddenly become crystal clear. So he’s very, very good at that.

Yost:  So these are obviously your two most important mentors; one early in your career and one later.

Bishop:  In terms of mentoring, yes.

Yost:  Are there other individuals, as well, in the field that had a major influence on you?

Bishop:  If I can say “influence” because I don’t know how major it is.

Yost:  Doesn’t have to be major.

Bishop:  Okay, yes. Marv Schaefer was a very good influence because he taught me a lot about the scientific method and also about asking questions. Peter Neumann was another one. I have a special affection for Peter because of his love of puns. Marv has the same love, too. And let's see, Becky Bace as well, because she supported some work I did early on, but she's a very wise woman and very much willing to give of herself to help others. So I would certainly call her — let me put it this way, not as major an influence as Dorothy or Karl, but she's way up there. So with Peter and Marv, and then I'm sure there are a lot of others that I could give you over a longer period of time. Of course, Larry Snyder, too, and Peter Denning, I can't forget those two.

Yost: You wrote a highly praised textbook, "Computer Security: Art and Science" published in 2003. Can you tell me about your decision to write a textbook and how you decided to organize it, and also why you framed it as "art and science"?

Bishop: I had to write it because basically, I couldn't find any book that would do what I wanted to in class. So I started pulling together notes and realized that others might find it useful. So, okay, we'll give it a shot and if there's a publisher who's interested in it, we'll go for it, otherwise I won't worry about it. It turned out there was. I framed it the way I normally would teach a class: an introduction and the foundational stuff, then modeling and the more abstract ideas, then cryptography as a tool underlying a lot of implementations, then the implementations, then some topic areas I just picked because they interested me, and then okay, now that we've got all this academic stuff, how do we actually apply this stuff in practice? And so I had a section on practicum. Anyway, that's how the organization came about.

Yost: The subtitle is "art and science." It's a less than typical title for a general textbook.

Bishop: The idea is that computer security is a science in the sense that you need to apply mathematical rigor, analyze problems, develop metrics, and so forth. But it's also an art because multiple things will produce the same effect so the question is which one works best. According to the scientific method applied to the technology, they're equally good but on the other hand, they have different implications that the future of the science can't tell you about because we don't know where it's going. And so it's a bit of an art to select the right one that will give you the right future actions, as well as the one that will work best in the environment you're dealing with. And we don't have that down to a science, yet. There's a lot of humanism in computer security and that's one thing that's often overlooked. Even with dealing with people in society, it's an art. And well, the pun would be that you get a lot from art, too. But the point is, if you approach it strictly as a science you'll get very good results, but they may not be very applicable. So that's why I called it both.

Yost: Are there standouts in your mind in the computer security research community that have taken the approach of looking at the humanities side of the topic?

Bishop: There's a lot of work on human factors, economics, and things like that now. In the past there was not. I'm not comfortable in naming specific people who led the field. I can name a couple, but I know all these people so I'd rather not. But in terms of actually applying art, or things like that, society and such, the people who seem to do it the most are the ones who cross the line between technology and design policy. Even they don't do exactly what I’m talking about, but they’re close. Jeffrey Hunter was probably a very, very good one there because he was an economist who wound up working in the federal government; economist, business, federal government. And he set up the first Computer Information Infrastructure Assurance Office, the CIAO. He was on the National Security Council. But he also understood a lot of implications of the technology in society and such. There are a lot of other people along those lines, it’s just that Jeffrey is the one I’ve interacted with the most. Oh, I should add also; I strongly suspect Marv Schaefer did a lot of that, although in areas that I don’t work in, because when he retired from the NSA he opened a used book store. My dad was a writer, my mom was a literary agent, so I’m kind of the black sheep of the family. [Laughs.] So Marv and I had a lot of fun talking about books, literature, and things like that. So it wouldn’t surprise me at all if he had done something like that. Peter Neumann would probably be another one along those lines because he has so much to do with policy and such.

Yost: You mentioned that you wrote your first textbook in large part because you wanted to use it for your courses and you saw it being useful to others to teach . . .

Bishop: I thought it might be; it turned out it was.

Yost: . . . computer security. Did you also see broader audiences beyond that and did that shape in any way how you wrote it?

Bishop: No, actually the only broader audience would be people who were interested in learning the underpinnings of things. It was never intended to be a book to give to system administrators to say here, this will teach you everything you need to know about securing your system. It was intended, for example, if the system administrator wanted to know why am I doing these things? Well, then I can go look at the practicum, that will point me back into the sections. But even there, I didn't want to tie it to any particular system. So the main audience was, in fact, the academic audience.

Yost: Probably it was useful to some systems administrators.

Bishop: I have been told that a lot of system administrators who are non-academics liked it too.

Yost: Marv Schaefer praised it very highly and he's not an academic—a former government research scientist, but not at a university.

Bishop: Yes. I know in certain technical circles as well, it seems to be highly regarded, nonacademic technical circles. But those people are typically not the system administrators, they're the researchers. Marv was definitely a researcher.

Yost: Can you tell me about the origin of the Seminal Papers Project, the early computer security history papers?

Bishop: At one point I was talking to some people and they had never heard of the Anderson Report, and then saw a number of papers I was reviewing in which people were basically rediscovering multi-level security. So I went looking for the seminal papers and couldn't find them. So I talked to Jim Anderson about this and he was running a project that had a little bit of money so he sent over a little bit of money to try to put this together. And we did. We were able to get quite a few contributions. I wrote to, I think, 20 or 30 people, saying can you send me a list of the 10 most influential unpublished or unavailable papers you know of? Nothing with copyright, or that's available from IEEE or any professional organization, but technical reports and such. A couple of people sent in ones that they named, and that was fine, too. And then I basically just went through and picked the ones that everybody seemed to name in common. A couple of people had ones that were very important, even though not everybody named them, or a majority didn't name them. I figured that was along the lines of well, gee, these are the ones I can think of off the top of my head. And so then I sent out a note saying does anybody have copies of these? And we were able to pull them together and scan them. We had enough funds to pay to have them scanned in, and OCR'ed, so we made a CD. I was hoping we'd have enough funds for a second one, but that wound up never happening.

Yost: Was the methodology in selection based solely on how many people named certain ones or was there a selection committee at the end or did you just make the call?

Bishop: My memory was, that what I did was I put together a — and again, this may be completely misremembered because it's been quite a while — but my memory was that I pulled together a couple of lists, and then circulated it to a few trusted very close friends in the area and said what do you think about this? Am I missing anything obvious or anything that if you saw this list you'd whup me upside the head and say, how could you forget that? I don't recall whether that list was changed or not, but that's how the list was developed.

Yost: What response have you heard from the project, both from the computer security researchers who you've long known, but also ideally, the intended audience, some of the folks coming up, the graduate students in the next generation who will hopefully learn from these pioneering works?

Bishop: I don't know whether this project was responsible for it but I do know that a lot of people now have their students read the Ware and the Anderson reports. I do that for some of my classes, as well. Certainly the project made them much more available than they were. The reaction has been uniformly good. I've not had any complaints about people saying why didn't you include this paper? I have been told a couple of times, the next time or for the follow-on, here are a couple of ideas. But everybody seems to like it, just have it, we distributed various CDs for many years, it's now up on the web. So it was very good and it was a fun project.

Yost: It seems like the type of project that given how important and successful it's been that it wouldn't cost a great deal to extend it.

Bishop:  I don't think it would cost a great deal but the problem is going to be finding the papers. What I would like to do is close what's there by finding the ones that I don't have. But for whatever reason, afterwards there wasn't interest, or funding, or whatever in trying to gather others. I suspect because now so much is on the web but I don't know.

Yost:  In interviewing Steve Lipner, there are a number of MITRE reports on there [available on your project site] but there are many more security reports from projects that MITRE and others — of varying significance — but some that were significant that aren't up there. So Steve was trying to get the ear of someone at MITRE to potentially donate them to us to make them public. We're trying to collect archival material on computer security but if it makes sense for them to come to you, then that could be a possibility, too. That's probably a site where computer security researchers go before a history research institute.  And no reason why they couldn't be available at different places different researchers might come across them.

Bishop:  Perhaps it would be possible to do something jointly. I'm not real comfortable saying don't take them, give them to me. I'm much more comfortable saying great, you've got them, let's see what we can do together. Yes, I'd be very interested in extending the project.

Yost:  I noticed on the site, the website for the project, that you thank Blaine Burnham at the NSA. Can you talk about the role he played?

Bishop: Blaine, I believe, provided some funding and also made suggestions for the papers, as I recall.

Yost: Can you talk about the graduate students that you've been the primary advisor of, and what have been some of the important projects that they have done?

Bishop: The property-based testing work I think was extremely important and very well done. That was George Fink and he completed a Ph.D., and then Mike Helmke, he was in the master's program. In fact, I'm pretty sure he got his master's; it's been a while. Eric Haugh, who also got a master's, did a lot of that work in conjunction with the NASA project, and did it beautifully. Another piece of work that was really good was the electronic recordation, what goes on in California when real estate is recorded over the internet. We built a model for that to analyze that they wanted to do and determine conditions under which you could get certain guarantees. The graduate student who did a fair amount of that work was named Tom Walcott, who also was very, very good. Another two are very recent. One of them took a physics class that's a university general interest, and the professor there was talking about modeling systems using what are called epsilon machines, which are basically probabilistic finite state automata. And the studnt said, you know, this is really cool, we could probably apply this to analyzing protocols, and he went ahead and did it. And the work was beautiful. Sean Whalen was his name, he did real good work. And then Sophie Engle also took some work that a master's student had done on modeling policy layers and essentially ran with it. She also combined that with the theory of vulnerabilities that I've been working on and produced a very good thesis on how to analyze systems for vulnerabilities. It's quite theoretical but it has implications for practice and that in fact turned into a paper that got accepted in the *IEEE Transactions on Dependable and Secure Computing*. She's now on a tenure track at USF and she's a fabulous teacher. And another student who got a master's here and then went to San Diego for his Ph.D., did some very nice work on reconciling logs. You have two separate sets of logs with differing time stamps, messages pass between the two systems and the clocks are off; what kind of bounds can you place on when events occurred? That was again really cool work. And then there was another student who — you said primary advisor but I wasn't really his primary advisor, but I helped out — I met him in 2003, if I remember correctly. He knew me through a mutual friend. We talked a bit about security and it turned out he was going to turn that into his thesis area and I knew his advisor from way back. So his advisor asked me if I'd be willing to kibitz on the thesis because while he's interested in security and knows about it, it isn't his primary area. What the student did was took work that involved the requires-provides model for attacking, flipped it around and started using it for forensics, and he built quite a formal model of how to do analysis and such. That was Sean Peisert; he does real good work. And then the last one that I'll mention because otherwise we'll be here all night, Karl and a student named Steve Templeton built a model called the requires-provides model that describes attacks where you go through stages. You want to reach certain goals so you use some capabilities to get to a subgoal, and that provides new capabilities, and you keep going until you get to the end goal. The student I'm taking about basically did some work on vulnerabilities database on his own time and then said, you know, I'd like to come into the graduate program. So he applied and he got in, so he started doing work with me on a funded project. He ended up taking the requires-provides model and taking the intrusion detection events and using that model to combine the events to develop large-scale pictures of attacks from very small minor attacks. And the work was excellent. He got his Ph.D., but just as a highly informal test, okay — this is not scientific for anything, though — we had a HoneyNet at the time, so we got the traces from that and had several grad students who were doing intrusion detection sit down and try to find the attacks. They came up with I think three or four, I don't remember which. Then the student ran his stuff over the traces and it came up with one they had missed. So his correlation engine had found an attack that experts — I mean, these students were the best of the best — had missed. So that was really heartwarming. So anyway, yes, of them, the one who's work I'm following up with most closely right now would be Sophie's, the vulnerability analysis work.

Yost: I would imagine that having a center here is not only a boost to opportunities for sponsored research but also really helped advance graduate education.

Bishop: Yes, it's nice to be part of what is arguably one of the top centers in the country, or in the world and to know that the center you're part of is recognized that way. Karl did a fantastic job founding this place and leading it. I doubt that he will tell you that he is the leader but everyone here looks up to him as a mentor and leader. So blame him. [Laughs.]

Yost: It's a center with some incredible talent and obviously a very thriving graduate program.

Bishop: The graduate students we have are absolutely wonderful.

Yost: By just looking at the website for the center, and I'm only interviewing you and Karl, but can you say a little bit about your other colleagues?

Bishop: First of all, I'm a little bit embarrassed because the website is out of date. As you know, we've had budget cuts and we don't have special administrative support for the center. So it's whatever we can cobble together and then sweet talk the main office into doing. [Laughs.] However, my colleagues, again, are exceptional. Sean Peisert is an adjunct here, he's a staff member at Lawrence Berkeley Labs now. He is a lot of fun to work with and really good, a very good mentor for students. Felix Wu was the person who came in after me. A leader in intrusion detection, now he's breaking new ground in social networking, and security of social networks, which is, right now, very critical. I don't know how hot the topic is but it's certainly a critical one, nowadays. Hao Chen, who came in after Felix, his thesis work was interesting because it was basically a static version of a property-based tester, although it was done completely independently of our work. It wasn't, well, let's take this work and apply it statically. It was well, let's do this. And when he gave the talk I thought, hey, that's cool! He does a lot of work with Android phones and portable phones and security in that realm. The work is both spectacular and very much of interest to the vendors and to news people, which is great because it gets great publicity for him and through him for the whole group.

Yost: One area that you that we really haven't explored in which you have published a lot is on security and electronic voting technology. Can you talk a bit more about it?

Bishop: I actually got into that whole line by mistake. The work on electronic recording, through that I got to know Tony Bernhard, who was the Yolo County Clerk-Recorder. And then when he retired, his successor was Freddie Oakley who I also got to know when I was working with Tony. And so one day, Freddie asked me if I knew anything about electronic voting machines and should she be reluctant to use them. So I went out and did a literature survey and basically wrote a report for her that said, it's the wave of the future but it's not quite here yet. Then I went to Maryland, visiting a couple of universities to give talks. Tom Walcott, a student of mine, was interning at a company called RABA, that Mike Wertheimer was heading up the research arm of. Tom was there, so we got together and he introduced me to Mike, and we seemed to hit it off fairly well. They said look, we're going to be talking to the state about something and probably do a penetration test; would you be interested in coming with us? Do you have a little bit of time? I said yes, let's go. So we met with the state people and they were talking about they needed to run a penetration test on a mock election on voting machines. Tell us about how you would do this, and such. Mike asked some questions and then I asked some and so forth. And afterwards, Mike said gee, if we get the contract, we would definitely like to involve you in the penetration testing. I said sure, it would be fun. So to me, the whole thing started just as a penetration test, not as a voting test. The penetration testing was actually a lot of fun because they had really good people; Bill Arbaugh, and Mark McLaren, and a number of others who I had known personally, or known about. All right, so it took us five minutes to control the machine you vote on, and 30 minutes to control the machine that counted. The only reason that one took so long was when we saw what it was, we were too lazy to write the program ourselves so we got it from someone else. [Laughs.] We then kept on finding other ways to cause problems. After that, I figured okay, that was a fun exercise but that's it. So one day I'm in my office and I get a call. I pick up the phone and I heard, "what are you doing on the West Coast? I thought you were in Maryland!" I said, ah, I've been on the West Coast most of my life; who is this? It turned out to be a research scientist down at Livermore, who was working with the state of California to check voting machines. He led an advisory group and he came up [and] we chatted. [It] turned out actually that he knew a lot of the people that I knew from MegaTest, because he went to CMU and those people went to CMU, and so forth. So, that went extremely well. He invited me to join the group; so I joined that. And then, the Yolo County election officials were getting machines so Freddie Oakley asked my group to test them and we did. Actually, I think slightly before that, a friend of mine in Florida was asked to look into a problem with an election. What happened was there were 18,000 undervotes in a hotly contested election, 18,000 people didn't vote and the question is, what happened? Could the software have caused the problem? Alec Yasinsac, the leaser of the study, knew I was very much into software analysis so he pulled me in [and] pulled in some other people. We analyzed the source code and basically concluded that the software did not cause or contribute to undervotes, as far as we could tell, but if you want to rig the next election here's exactly how you do it. Parts of the report, too, were sealed. There was one paper that was written that essentially said here is something that demonstrates that the machines have a flaw that would've caused the undervoting. Well, we had found that a few days earlier and I had actually walked through the code, which the people who wrote that didn't have. It turned out that they were confusing correlation with causation. They said these events correlate, therefore, one event must cause the other. It turned out both; the correlation was correct but there was a third cause, which had nothing to do with the undervotes. So, apparently, a House committee — this was a congressional race — asked Alec specifically about that. He showed them the redacted report and they said okay, and moved. And then Debra Bowen got elected California Secretary of State, and I got pulled into a top-to-bottom review of all California electronic voting systems. I pulled in a couple of my grad students, who were fantastic, and some colleagues: Dick Kemmerer and Giovanni Vigna from UCSB, and a bunch of other people did. Ever since then, I've been associated with electronic voting. One of the big mistakes I think that everyone's making is looking at the machines. What you need to do is look at the machines as a part of the process. I've had some funding from NSF to do that. I'm working with people in UMass Amherst who do process modeling and model how processes work and look for single points of failure, which are equivalent to vulnerabilities in that realm. In fact, the work we've done, we've worked very closely with Yolo County and we've suggested things that have helped them. We started a couple years ago working with Marin County, and they loved the work, too. So we're hoping to continue with it.

Yost:  In talking to Peter Denning, who is quite skeptical of the possibility of a fair and safe computer voting system, one thing he stressed was even if other elements of security can be addressed that he had the issue of people voting not at a polling place, but at homes, offices, etc.

Bishop:  Internet voting.

Yost:  Yes, internet voting. There can be coercion, the selling of votes, and the verification for people that are manipulating elections, it simply isn't possible because if you pay someone to vote a certain way they can go into a voting booth and, of course, vote however they want and no one would ever know.  But that's not true without voting places.

Bishop:  There are two problems. First of all, I've not seen any evidence that we can do internet voting with the technology we have and I have seen a tremendous amount of evidence that says we cannot. Personally, based on what I have seen, given the state of the art, if one of the goals of the election is secrecy of the ballot, anonymity of the voter, accuracy, and integrity, the internet does not make it. The internet — and by internet I'm including the end point — simply does not have the mechanisms to support that yet. But there's a much bigger problem, too, which is transparency. One of the key issues, at least to me — and again, we need to emphasize this is me, personally — what I said earlier about the internet not having the mechanisms to support it, that I can defend scientifically. This part I'm not sure I can but it's, to me, critical, is the credibility of the election. With a paper election, for example, I can go to the Yolo County police station or to Election Central the morning of the election, watch the polling inspectors come get the voting material that is waiting for them, follow them to the polling station, watch every step of the setup, [and] stand back and observe the entire day's election. I can watch how everything goes — I just can't go in the booth with anyone, of course, or stand behind the shield of the machine — and then watch them tear it down, watch them check the ballots, count them count, do the post election verification, watch them seal the bags, watch them drive it up to Election Central or if they're far away, watch them put it in the drop box and watch the drop box for the sheriff's car to come by, then drive to Election Central, watch them unpack the bags, and stand behind the barrier and watch them count it. Everything is open and I can see every step of it. But with an internet election there's no way you can do that because you don't know what happens to the vote once the person types it in on the computer. You don't know if there's a virus or some other type of malware or Trojan horse that will change the vote before it's transmitted. And I don't care how good your software is, when I click on something it may show one thing on the screen but that screen may be a fake screen with the software actually reading what's underneath it. So unless I have a trusted system to vote with, I don't see the transparency and to me, it's not enough to know that your vote was counted. You have to know the entire process that it goes through. And in fact, in Yolo County, a lot of what they do is completely unnecessary but if they ever go to court, if anybody ever sues them over a question of the elections, they can say yes, we did exactly this, this is how we validated it. I don't see how you can do that with electronic or internet voting. Would I like it? Yes. Do I believe it will come? Ultimately, at some point it almost certainly will. Are we ready for it now? That's really up to the body politic but from the requirements that I know for an election now, the internet does not support it and the computers, quite frankly, don't support it. So anyway, that's my feeling. I'd love to be proven wrong and in fact, a lot of the voting work I've done is to try to prove myself wrong.

Yost: Earlier, we talked for a while about your work and publications on UNIX security but I wanted to ask you if you could place the body of work you've done on UNIX security within the broader literature. What have been your largest contributions and what are some of the most important contributions others have made to the topic of UNIX security?

Bishop: Oh my, so many people have made so many contributions. Mine have maybe been in the area of vulnerabilities analysis, and also analysis of passwords, speeding up the password guessing tools. Actually I should say proactive passwords where the system will control what you pick as a potential password. I did do a little bit with encryption on UNIX, also. I think Dennis Ritchie and Ken Thompson, of course, were the ones who made the biggest contribution. [Laughs.] But beyond that, I think so many people have done so much good work I think it would be unfair to identify any particular set as being the main contributors to it. So let me weasel out of that with a squeak.

Yost: Okay. A number of computer security pioneers that we've talked to have used different language, but talked about different perspectives or world views of security or factions in the field. Do you see yourself as fitting into any particular category? Looking at the range of your publications, you seem extremely broad and have done work in so many areas.

Bishop: Thank you. I try to avoid any kind of faction and generally go where my interests are. If you have absolutely have to have a particular faction that I'm not a part of, it would be the database people. But even there, I'm doing work on data sanitization and its applications to the database realm. I'm just not in that realm myself. I've always thought that computer security work is so much fun, I've never wanted to tie myself to anything in particular. I seem to stumble into stuff by accident, which is half the fun. For example, the work I've done on attribution really came about because Jeffrey Hunker and I at one point had a conversation, and Carrie Gates happened to be in the room, heard it [and] joined in. The next thing we know, we're sitting down at eight o'clock in the morning at a workshop, before the first session, working out some interesting paths of research, and the same with the work on the insiders. That started at a conference in Italy. I couldn't sleep for some reason, so at 11 o'clock I went down to try to find a soda and Carrie was sitting there. So we started talking and we went across the street or next door to talk about security and such, and we wound up basically coming up with an idea for a model that we've been working together on for about four, five years. Completely by accident, and if I hadn't come down there we probably never would've talked. So I guess that's kind of the story of my career, I stumble into things by accident and I sometimes get lucky. I know there are factions. I try to ignore the factionalism [and] not be a part of any of it, just follow my own path. My ideas often take years to mature or to take effect. Hopefully, I've made the discipline a little bit better. In a way I kind of feel like

Archibald Cox, if you remember that famous quote of his. “I tried to do the right thing, I tried to make it a little bit better and if I failed, what the hell.” [Laughter.]

Yost: What do you see as some of the greatest computer security vulnerabilities or risks that we face as a nation in the world today?

Bishop: What time is it? [Laughter.] They change so rapidly. Actually, a more serious answer is I think the most serious vulnerabilities are those that involved people interacting with systems or policies that are inconsistent with the way that people think, or the way society’s using computers. That disconnect is huge in some places. A couple of good examples are the design and implementation of software. The state of the art could be very good but in practice it’s quite bad. And that’s not because we don’t know how to improve it. We do. It’s because to improve it would require changes to market forces and to the way that people think, and to the way companies and organizations work. That creates vulnerabilities at the system level. There are also vulnerabilities in that people want to lock things down that they have no hope of locking down. A good example is when countries try to cut off internet access to the world. Egypt tried that, and look what happened. Egypt could’ve turned off the phone system but then they would’ve lost all contact with their police. Other people were using satellite phones. There’s no way Egypt could’ve shut that down without putting a bowl over the country. It’s only going to get much worse. Is that a vulnerability? Well, to the government it is because they don’t have control. To the people it certainly wasn’t. That brings up another point; how do we define a threat? For example, in the United States, restricting access to the internet, or the internet to the world would be seen as a threat. In China, failure to do so is seen as a threat. That's why I say I think the biggest threats we're dealing with are people oriented. The insider has suddenly gained a great deal of currency, but that's kind of another topic because I'm not sure the insider is a real problem. I think it's a mischaracterization of several other problems that exist, although the idea of a human betraying a trust is certainly a problem. The internet infrastructure in the U.S. is probably quite vulnerable. There are systems in parts of society that depend on the internet. What would happen if it failed? They're vulnerable to distributed denial of service and other attacks. Okay, so you asked me for vulnerabilities and threats that were the biggest. So now that I basically said that what I'm going to tell you is not true, I'll go ahead and tell you. [Laughs.] I think the first biggest problem is our attempts to restrict information. I consider that a vulnerability because often the people who need that information can't get it. We see this a lot with reports of vulnerabilities that are floating around. The ones who should get it in theory are the vendors and the people with those systems, but how do you distribute that information so that it reaches everybody so that they can respond? And in these days of zero day attacks, how do you immediately respond to those attacks? The inflexibility of these systems is a vulnerability and the problem is that it's a double edged sword because if you make them too flexible, then they may not respond correctly. So how do we balance all this? That's one issue.

Another issue, as I said earlier, is the poor quality of most of the systems and the knowledge to lock them down often can be quite arcane or require a tremendous amount of information. And the problem is you have to know which information applies, so this lack of guidance also is a vulnerability. The problem is you have all this guidance out there but how do you know what applies? And that, again, is a people-oriented vulnerability. The insider problem is probably, to my mind, one of the top two issues or vulnerabilities. And part of that is because I'm not sure we're characterizing the problem correctly. We're looking for people who cause problems. It might be better to look at what information will cause a problem when leaked? Or what system will cause a problem when compromised? Then how do we figure out who can do that? We probably should be working backwards.

The other biggest problem, I would say, is simply the lack of assurance in systems. Third big problem, credibility. Often people believe that computers will do things that they won't. I'm thinking digital forensics, and often they don't dive beneath the surface. As an example — I don't want to name the company or the people involved — but somebody mistyped a name and it went to a porn site. It was an elderly lady, she panicked, and she started clicking to try to get it off the screen. Well, the company has what's called the two-click rule. If you go to a porn screen and immediately leave it, they assume it's an accident. If you click on it twice, they assume it was deliberate. And so she had to go through orientation to find out why pornography was bad when it was very, very clear to everyone involved that she was horrified with it. Again, the problem was taking the computer literally and not going behind the scenes. My concern about this is digital forensics. There are all sorts of tools for analyzing systems but my question is, can those tools be defeated? And how do you ensure proper interpretation? I'll give you a very, very good example. Have you heard of this singing group called A-Teens? This was about 10 years ago. It was a group that was really popular with tweens, which of course my daughter was, and she wanted to go see their web page because she was just learning about the web and such back then. I said, okay, I'll try, so I typed in ateens.com. Thank heavens she was looking the other way because I immediately backed out of it; it was a porn site. The right one was a-teens.com. First of all, if you didn't know me and my daughter, and just saw the computer there, would you see a father looking at porn with his daughter? However, here's the real problem, if you analyzed that computer — I didn't realize it until later — but you would see that page on the disk because it was in the web cache. So imagine someone going into a porn site at work by accident and then it being stored in the cache. Or going to a child porn site by accident because of confusion of names or something and going oh-my-God! [Phump], I'm out. The disk later gets seized and they do an analysis, they'll find child porn on it. Is that a crime? I would have a very hard time convicting someone of accidently going to a site and bouncing out immediately. But the problem is that the tools wouldn't give me that interpretation. I don't know whether the tool would say yes, there is one instance of child porn here, it's in the web cache, and if you look at the time it's clearly not intentional. It was viewed for a second. But I don't know if the tool will tell me that. So the blind faith people have in certain aspects of computing is a very serious vulnerability. And the other aspect of this that makes it very bad is, what happens if we get evidence of a cyber attack from someplace like China? Or actually, let's pick Iran. How do we know it's not someone spoofing those addresses? That can be done very easily. Question is, will the policy makers and the people who are to respond or the people who are interpreting the data know this, understand it and be able to say no, let's make sure of what's going on before we decide what to do. And this is why talk of cyber offense really bothers me because when you defend, if the response or defense is to launch an offense, most soldiers would say that's the right strategy but the problem is you want to make sure you hit the right people if you believe that that is acceptable. But in the virtual world, not everything is as it seems. In the physical world, you can pretty easily tell where something came from.

Yost: There's no clear friend or foe in the computer world.

Bishop: Exactly. It's the old saying, when you're on the internet nobody knows you're a dog. And those are the vulnerabilities. I don't know if they're the most serious or not but they're the ones that scare me the most.

Yost: What do you see as some of the most important lessons from the longer history of computer security that haven't been learned?

Bishop: Santayana: those who cannot learn from history are doomed to repeat it. For example, a lot of the stuff that's being used now and that people are touting as brand new is actually very old, like the whole idea of virtual machines for security. That was being done back in the 1970s, yet now it's been rediscovered, and people are rediscovering a lot of the problems and a lot of the issues that were known back in the 1970s. To their credit, because the early papers are out there and more is known about history, a lot of them are pointing back to that work. Multics had, for example, memory mapping. In fact, everything in Multics was memory mapped. Yet, the UNIX community when it discovered it, treated it as the greatest discovery since sliced [bread]. [They] treated it as a breakthrough. Maybe putting it on UNIX was, but the ideas were well known. Buffer overflow attacks, we've known how to counter buffer overflow since the 1960s and yet there's still so much work going on there and people are discovering different ways of handling them. But again, a lot of what they're discovering was either known or hinted at in the 1960s and 1970s. For example, the whole notion of tainted data. When you read in data it's considered untrusted so it's tainted. Then after you check and validate it you can move it up to untainted and then do other things with it. That was known in 1978 under the guise of the Biba integrity model, and yet I remember in the late 1990s when people started doing this I was told that this was brand new. I asked what about Biba? They said oh, this is completely different. To this day I don't understand how it's different. It seems like a version of Biba, that is, multi-level integrity with two levels. I think that's probably the biggest lesson, that we don't learn our lessons. On a much more prosaic level, I think one of the lessons from history that we've forgotten to a large degree [is] how important people are and how controls have to be designed in such a way that the system remains usable. And the symbiosis of hardware and software that will bring that about. With the right hardware we could eliminate buffer overflows tomorrow, but for whatever reason that hardware is not being produced. I don't know whether it's too expensive or whether it's just that nobody would use it or whether you have to write new compilers to take advantage of the hardware, of the security features that would be built in. But I would say that would be a very important lesson, too, that security has to be dealt with holistically. It can't be, here's a vulnerability, we have to focus on that. It's, here's a vulnerability

[and] here are the causes and effects, so we need to focus on the whole thing. That's what makes it so hard.

Yost: That is a term that came up multiple times in my interview with Peter Neumann, a holistic approach.

Bishop: I would very strongly agree with Peter on that. That's why I'm emphasizing people so much I think, because if you see security in a vacuum, you're going to have very good security for a vacuum. It's like the old Groucho Marx line about he'd invented a cure for it so there was no disease. [Laughs.] It's like a lot of the laws nowadays, they work real well for the law abiding people, but not for the criminals.

Yost: With your prolific career there are some topics I haven't brought up. Are there things you'd like to talk about that I haven't asked about?

Bishop: Quite frankly, you've been so thorough. Simply that there are an awful lot of really good people in security and there's an awful lot of good work to be done and an awful lot of people who can do it. Oh, actually this goes back to the lessons we haven't learned. One area that's vastly undervalued — security nowadays tends to focus on what can we use in the immediate or near term future. And that's very much an approach that's going to cause problems because we have to look at the far future. To do that, we need fundamental, basic research and I guess this is kind of what I'd like to talk about for a couple of minutes. We need two types of projects: the ones that are solving today's problems; the ones that will figure out what tomorrow's problems are and how to deal with them and that will build a basis for us to do that. And all too often, the funding we see or the problems we see posed are the ones we're facing immediately. That's entirely understandable but it's very short sighted, and I think that's another one of the problems that we haven't learned from the past. Other than that, I can't think of anything. Everybody is trying their best and hopefully things will work well. I'd much rather be optimistic than pessimistic. In fact, I am optimistic.

Yost: Well thank you so much for taking the time to give this interview, it's been extremely enlightening and helpful.

Bishop: Thank you for inviting me.