An Interview with

ROGER R. SCHELL, Ph.D.

OH 405

Conducted by Jeffrey R. Yost, Ph.D.

on

1 May 2012

Computer Security History Project

Monterey, California

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

Col. Roger R. Schell (USAF, ret.) Interview

1 May 2012

Oral History 405

Abstract

Dr. Roger R. Schell, a retired U.S. Air Force Colonel and current president of Æsec Corporation, is one of the foremost contributors to and authorities on "high assurance" computer security. In this oral history he discusses his formulation of the secure kernel and reference monitor concepts (in the early 1970s), his work that led to security enhancements to Honeywell-Multics (mid-1970s), his role as deputy director of the National Computer Security Center (including leadership on TCSEC or “The Orange Book” in the early to mid-1980s), and commercial (high assurance) computer security enterprises he’s led since retiring from the Air Force.

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 and I'm here today on May 1st, 2012, with Dr. Roger Schell in Monterey, California. This is part of the Charles Babbage Institute's NSF-sponsored project, "Building Infrastructure for Computer Security History." Roger, I'd like to begin with some basic biographical questions. Could you tell me where you were born and where you grew up?

Schell: Right. I was born in Richey, Montana; which is in eastern Montana; and I grew up through grade school there. And then when I was entering junior year in high school, I moved to Belgrade, Montana; which is in the more western part of the state; and grew up there. I went to Montana State College, which is located nearby—graduated from Montana State.

Yost: What did you study there?

Schell: Electrical engineering.

Yost: At what point did you first develop an interest in electronics?

Schell: Well, probably around age of 10 or 11. When I first started school, we didn't have electricity and so there wasn't electronics around, although we did have battery-powered radio. So I suppose my earliest interest in electronics was associated with a wind-charger that charged a battery, which we used to run a radio for a couple hours a day. Then, after we got electricity, I got interested in radios and got the handbook of the

Amateur Radio Relay League. I sort of self-studied; I looked in encyclopedias. Then when people around the neighborhood began to have radios, tubes would go out, things like that. I developed some ability to just basically troubleshoot radios. So I was—with absolutely no knowledge and probably limited skill (laughs)—I was sort of doing tube replacement and that sort of troubleshooting in the neighborhood.

Yost: And did you go right on to do graduate work at Washington State University, after completing your degree at Montana State?

Schell: Yes, I did. I graduated in the ROTC program at Montana State, and the Air Force, at that time, was looking for people with advanced degrees in electrical engineering. Since I was in that field, they wanted to send me there. I had to choose between a full fellowship offer from Stanford, and the Air Force's sponsorship to Washington State. But since I had pretty well committed to the Air Force, I went with the Air Force program.

Yost: And at WSU, did you have exposure to digital computers?

Schell: Well I did. I actually had exposure to digital computers at Montana State. I worked my way through college and one of the part time jobs I had was in the electronics lab at the university which had a project that was sponsored by IBM. They were doing work on digital disks, or recording magnetic disks. And I was just a technician doing that. By that time, and actually beginning back when I was in high school, I was involved in a

radio station. I'd gotten my First Class FCC license as an engineer for a radio station, so I was involved fairly deeply in electronics. Another one of my part time jobs was that engineer job at the radio station. Then, my senior year at Montana State, in the ROTC program one of the jobs I had was the cadet personnel officer. One of the things I would do for the ROTC professors—as all the personnel officers did—was to compute their grades. They graded on the curve for the test scores and so you had a lot of computational sort of things. And the school had an IBM 650, with drum-based memory. I took that personnel project, which would take many, many hours with a hand calculator, etc., and put that on the computer. I punched the cards and put that data into the 650 and programmed it to calculate the scores. And so I was able to do that job much more quickly than my predecessors had. That was my first exposure to computers, both in the research lab and running the 650.

Yost: Did that create a strong interest for you in the field at the time?

Schell: It was a reasonable interest; I thought it was neat. My primary interest was electromagnetics and radio, and that sort of thing; where I had been most of my career. So it was not such a strong interest. At Washington State, my research involved antennas, and antenna pattern prediction. And at that time, there was just emerging the log periodic antenna. I don't know if you know much about electronics, but you see most TVs today use the log periodic when they have an external antenna. They have multiple elements. And at that time there were pattern predictions for things like rhombics, another antenna

type. We would do these predictions based on the elements size and shape and you'd go through a mathematical calculation to predict it pattern.

There was some research work being done at the University of Illinois that tried to do predictions for the log periodics, and they were not able to get reasonable predictions for the patterns. And I said, well, it's just science out there; somewhere, there's got to be a way to predict that. My hypothesis was that in electromagnetics, there's something called the Near Field and the Far Field. All the antenna pattern predictions only dealt with the Far Field. I said that since the log periodic elements were next to each other, I hypothesized that the Near Field Effect was dominant. But the problem was that the actual calculations of doing this using numerical integration and things like that were just not feasible with pencil and paper. The school had a 709 IBM computer that wasn't overly used, so I used that to do the antenna pattern prediction. And, in fact, since Illinois had published measurements for the antenna patterns, I was able to use that to confirm I quite accurately predicted the antenna patterns, which hadn't been done before. My advisor thought that was cool. I think he put together and published some paper—I don't even remember the details—related to that.

I had concluded a lot of graduate level courses by the time I graduated in electrical engineering from Montana State. So when I went to Washington State, and was in the electrical engineering department, I ran out of electrical engineering courses to take. Because I had to have a full load, I said well, I'll go look at nuclear engineering; and so I took a lot of nuclear engineering courses. Well, in nuclear engineering you also have a lot

of patterns; radiation pattern, and things like that of a different sort to deal with. They were just getting into computers, but nobody on the faculty really knew much about computers. I had self-taught myself a little bit about the computer FORTRAN language, which was just emerging at that time. So I got in the good graces of the professor by teaching FORTRAN for a week to the students in his class. Then he gave FORTRAN assignments because, it was just coming on campus. So while I got involved in computers, it was always tangential to the main thing I was doing. It was just a tool to do computation.

Yost: And in what year did you complete your master's program?

Schell: It would have been 1963.

Yost: What was your first assignment in the Air Force, coming out of school?

Schell: My first assignment for the Air Force was something called the Electronics Systems Division, which is at Hanscom Air Force Base, in Bedford, Massachusetts, near Concord. At that time, the Air Force was just installing their Ballistic Missile Early Warning System. The Russians had a long-range missile capability. Then, a few years after Sputnik, we were worried about Intercontinental Ballistic Missiles (ICBM), so created our missile detection system. It was the largest government procurement ever, up to that time. In those dollars it was in excess of a billion dollars for a radar system to detect missiles. Since it was a radar system and I was experienced with electromagnetics

and antennas, that was my assignment to go there and work there in that program office. I started out primarily in the test division, where they were just going through various tests. They had sites in Alaska, Greenland and in England, and I went to some of the sites. I looked at what they were doing there, and did some things related to the antennas. But it turned out, because you have this huge, literally 50-foot-across radar dish antenna, to control that you needed computers. So, they had an IBM 7090 computer, dual for redundancy, that ran the radar. Not many Air Force people knew much about computers. Because I had programmed and had done computer sort of things, I ended up getting a lot of extra duties associated with the computers. So that was additional exposure to the computer, and that was my first assignment. We wrapped that up as the system was completed and went on to my next assignment.

Yost: Were the sites for this ICBM Early Warning System the same as sites for the Semi-Automatic Ground Environment (SAGE)?

Schell: They were not. It turns out that my next assignment was related to the SAGE system; and obviously, you've heard of the SAGE system. At that time, the Distant Early Warning System, it was called the DEW Line, was a tier of very northern radar sites that were detecting aircraft, as opposed to missiles. It was a lot older, of course, than the missile warning, which was a new deployment. And they were getting installed, a system to semi-automate the radar sites. Until that time you had the thing that you see in the movies, the old movies, of an operator who would sit at a radar screen and visually see something on the screen. Then he'd get on the phone and call down to the headquarters.

They had the major headquarters at Anchorage, Alaska. They did the typical write backwards on the back of plexiglass, and that sort of thing. And that was the way they operated. It obviously was slow and had limited capacity. So they wanted to improve the through-put—the amount of things they could report on—as well as improve the speed. And so they semi-automated that. They didn't have the budget to fully automate, so they took the things that they'd done with SAGE, and they did an intermediate kind of thing because the SAGE system—you're familiar with that, Jeff—had these huge computers that were blockhouses, and you couldn't (pause)

Yost: the FSQ-7s

Schell: Yes, AN/FSQ-7 computers. And you couldn't put those at every radar site in Alaska. So their semi-automation was giving the operator sort of a track ball that he could position something over a target that he saw; click on it; and it would generate a teletype message. And then the teletype messages would be processed by a computer to generate a track. And so they had communications; the network communications was teletype, literally. So we were installing teletype machines and that sort of thing. And the computers that were doing the process; taking the teletype message and processing it. They in turn took that and replaced the writing on the screen backwards with digital projections on the screen, with a high precision optics grid. So it was an interesting project, and it got me more into computers. It was a place where I began to encounter the problems of what today people would call software configuration management. Actually, I believe the term "software" was derived at that time from the fact that when you bought

things like radar systems or any military equipment, you bought two sets of things: you bought hardware, and software. And the software was the manuals, the tech orders, as they called them; that sort of thing. And all the computer programs were written up as technical orders; as documents, and instructions. And so you ended up calling these things software. So, out of the SAGE era, which was the same era, they called them software. The way you manage documents, of course, is you have revisions to documents and you have updates and such. But, that didn't fit very well [with] the way you did computer programs, which tended to be more like engineering. But people didn't treat it that way, and the people that were largely involved in the major companies doing computer programs were, from my perspective, not at all in the vein of engineering discipline. Software just didn't go to discipline or it just didn't exist. They were very much just independently doing things and even though they were part of projects, they didn't have an engineering discipline. The results were that, for example, in SAGE, every site had to have custom built programs that ran on that site. And the same was going on in Alaska. This is crazy. You can't do that. I said why can't we just have site-specific adaptation data, and in SAGE, they did have a notion of adaptation data—then why can't we just have one program and just have it adapted that way. Well, and so I, with an engineering background, I had a different perspective than the contractor people that were building these systems for us.

Yost: Was that [the Alaskan system] part of the SDC contract?

Schell: Well, at that time, the Alaskan system was a separate contract, run by Philco Ford. And they were using a computer that was actually the computer from the missile submarine program that Admiral Rickover had installed; they had introduced the notion of strong configuration management on the hardware side. But they had no idea how to deal with it with software side because you already had an idea of how to do updates in software with manuals.

So, one of the things I thought was an important distinction was rather than calling these things software, distinguish them as such: I said, we'll call this computer programs. And then we'll have hardware; and programs; and software, which in the traditional meaning of software, were the manuals and things like that. And so I began to introduce some computer program configuration management kind of things into the Alaskan system. The system development had faltered for a year in the installation and deployment. I'd worked on a system previously in its manufacturing, in the Philadelphia area, where the Philco computers were built. And when I got to Alaska, I just saw the lack of applying engineering discipline on the computer program side. For all the hardware and everything, you had all sorts of configuration management and controls. It didn't exist for computer programs, really, not at all. And much to the annoyance of the contractor, I began to implement those kind of controls in the testing and the integration—and with good results. As a result, we did succeed. The project had actually been, by several of the Air Force management, written off as unmanageable, and a failure. They were just going to scrap it. And we, in fact, successfully delivered it—much late, because of these things. But with that, I was much more involved in the computers. I was no longer really

involved in the radars, even though it was a radar system, and that's why I was there. And as I was sort of wrapping that up, the same program office was doing the SAGE systems.

Yost: And what year was that wrapping up?

Schell: That would've been, I'm going to say, order of 1964-1965; 1965, I guess, probably. 1966 maybe. End of that era. And SDC, the System Development Corporation, was doing the software for SAGE; and it was also doing the software for the follow-on system; essentially a transistorized version of SAGE. It was called the Back-Up Interceptor and Control System (BUIC). And they were just putting together the software for that; and since I'd successfully improved the management structure with the Alaskan system, I got assigned as the project engineer and software engineer for the BUIC system; the backup; the transistorized SAGE. And, you know, as time went on; maybe six months or nine months; there hadn't been a lot of management, really, at all of the SAGE system. The government was just largely hands off; it just sent money to SDC. I was given the job of also managing the computer programs for the SAGE system. So it was still operational; still getting updates as such; but the primary goal was to replace SAGE with this transistorized system, which was much more efficient. Although it was a backup system on paper, everyone understood that over time, it would become the primary system. And so that was something that I did for several years. So I sort of naturally moved from the Alaskan part of the broad SAGE system—the air defense system—to the transistorized, to the SAGE maintenance. And there my ideas, in terms of configuration management and organizing, and engineering, had become much more

solidified. I was rather distressed at the way that the cost and schedules got out of control, in terms of the software things. From my point of view, I could see why that was the case. It was the total lack of engineering discipline in the software. And I'd become familiar with, in the ballistic missile system, the configuration management process. The Air Force, at that time, had taken what Rickover had done; had institutionalized that to a set of regulations and procedures for configuration management for hardware. And I said, "I can do that for software," you know, computer programs, even though it wasn't done. And that created a lot of turmoil. SDC, in particular, did not really get enthusiastic about the structure that I wanted to impose. But I had the same boss that I'd had in ballistic missile program. He moved and became my boss, who's also the program manager then for the SAGE and those things. And I just told him, I said, you know, this is very inefficient. It not only wastes money, it wastes time, and we can do better. He knew nothing about it but he knew I'd delivered for him before and so he said, "Fine," you know, "do what you need to do." And I said well, we're going to actually translate the computer hardware configuration management notions into the computer program area. And even as recently as a decade ago, what we established then became a set of military standards for the configuration management for software, and those persisted. And I was the one who directed the preparation and creating of that configuration management process for initially the military, and then the IEEE, and other people have picked up and used those same sort of notions; obviously a parallel development of similar sorts of things. So that's sort of what I did there. It was definitely one in which there was a lot more push-back than there was cooperation from the contractor. I declared that when we ran a test run on something, we were going to use that as a baseline; the notion of a

baseline and then they would make the next version and we would be able to do regression testing. And so okay, we can run the test against this; and we'll run the test of the next one. And, you know, it was ridiculed; it was various criticized; this sort of thing. But, I said that's what we're going to do.

So I ran a test and I came back a couple months later, and I took with me the test deck. I said I've got here the test deck of the test data. I want to run that test data on the new version. It didn't run. So I said well, fine; take me back to the version you did before. So okay; here it is. It didn't run. And I said okay, well why doesn't it run when you said it was the same thing. Well, they said, no, it is the same thing. And we had a fairly, you know, knock down drag out kind of thing; and I said—the end of the story—I said I want to lock up the deck in safes and only I will know the combination. And they're going to stay locked until I come back. And when you have the updates, then you're going to apply the updates to that base, not your base. Well, as you can imagine that caused a great deal of consternation. And I said well, too bad. And so we did that and that solved the problem of that sort of regression testing. And now, I said, that we've done that in terms of the test; now we're going to have similarly controlled baselines that go to every site and we're not going to build this thing site by site. We're going to send out one base version to every site and then we'll apply the adaptation data and we're going to build it so they can install the adaptation data on site. And you would've thought the world was going to end because it was a very major revenue stream. The local site support team (as I recall was something like 30 percent of the contract) was actually doing the site support and a good deal of it was about all of this process. And so I had pointed out to me that

well, you know, that was going require more than half their time; the work had to be done. Of course, it was put not in the manner that I was, that they were going to do less work. Rather it was put in the manner that all this work wasn't going to get done and therefore, everything was going to fail because we were just doing what was needed. My decision, you know. And so this resistance was to the very thing, trying to install configuration management. When I enforced the baseline by putting the version into the safes, I got called to the [SDC] president's office after I'd declared this was what we would do. They had locally, an Air Defense Command representative; a military guy; the user command came in. They lived in the same plant, and what I would call had "gone native". So they came in and, of course, he far outranked me; and he said no, we can't do that. If you install that process, we've checked it with our programmers, and a third of our programmers are going to quit. And the nation will be at risk. And the world will collapse.

I said, well, that's the way it goes. He actually called my boss on the phone and said, ah, we seem to have a misunderstanding here. My boss said something to the effect, well, what do you mean a misunderstanding? You don't understand what Lt. Schell is saying? Oh, we understand it. Well, we just think you need to do some adjusting- Lt. Schell is being unreasonable. And my boss said to him, he's the project engineer and if you've got a problem, discuss it with him. And I fully support whatever it is that he says is necessary to do. Click. And, you know, panic broke loose. They said well, a third of the programmers will quit. And my response was, that's about what I think you'll save. And I said, this is a good process because the right third will quit; the ones that don't want to do

this are the ones that are part of the problem. Said it is a good solution. And so we proceeded; and indeed, it was the first major project that, at the end of the day, the project was delivered on schedule and under cost.

Yost: Resulting in a major economic benefit?

Schell: Major economic benefit; major cost benefit; and we've instituted the whole process that became a military standard for software so that all the software shortly thereafter began to impose these standards. So we used this as a test bed. I'd make them write the documents in military standard format. We refined it; we revised it. And they started by saying well we don't know how to do this; and so I actually had to write some pieces of it and they saw that I was going to be serious enough to make them actually do it and then they would do it. So, yes, it was an interesting turn. By that time I was heavily involved in computers. (Laughs.)

Yost: Moving back to the ICBM Early Warning System, when you first started working on that, were you thinking about computer security at all and was the Air Force thinking about it?

Schell: No. Not as you really think of computer security today. Obviously, the information was classified and so it was transmitted encrypted. And so we had that kind of encrypted communication security, and it was a typical air gap security. We had, you know, unclassified things, and we had air gap, and we had the classified. And as far as

that ballistic missile part, there was no real what you would call multi-level environment where you had a mix of levels. And so there was no computer security, *per se*. When I got to the SAGE system, and the BUIC system, yes, we had computer security as a recognized problem but one people didn't want to talk about because there wasn't much you could say about doing anything about. But the problem was that all the air defense system ran over unencrypted telephone lines, the network; because even though, some people think networks came later, the SAGE was a network system. It was a distributed network system. And the networks were unclassified networks. And yet the tactics for example that I was responsible for putting into the programs were classified for the control of the aircraft. And so we had secret level algorithms running connected to the unclassified lines outside. Our rationale — and probably reasonable at that time — was that there wasn't any adversary that had sufficient knowledge to exploit those kinds of things and so we sort of pushed that aside. But we recognized that as a potential problem. I don't know how much you want to go down that path, but one of my first introductions to the need to have to do something about the computer security problem was actually with regard to the missile control. Part of the SAGE system was that it was an air defense system, of course, and they were primarily designed to deal with the airborne threat. And part of our strategy for dealing with a massive airborne attack was to do nuclear detonation in the flights coming in. Before they reached the mainland, if they came with a massive flight coming over the water you would have a system called BOMARC, which would launch a nuclear missile and detonate it to destroy the bomber force. So now you had missiles, nuclear tipped missiles, on U.S. soil being controlled by computers. There was always—at least for a long time—the issue of nuclear safety that had been there.

Somebody raised the question of, so how do we know that where that missile goes is properly controlled. You've got the usual stories; well, we've got two keys; and they are switches; and all of that sort of stuff. And somebody wrote up this report; you know, some manager someplace that says, there's not a problem because of all of that. Since that time, and as part of the SAGE system, in the broad, and I would seen as the manager for the software and I was asked to review and sign off on the report. And I didn't really know this was going on very much. I wasn't involved in this study. Somebody had done this study, some scholarly group had put this together. I looked at it and I said it's garbage. You can't say that. No, all that stuff doesn't matter because at the end of the day, the keys that you are using are not in the electrical wires, their input is to a computer and the actual launch command is given by a computer. And the controls that are provided are provided by a computer. There's no positive control; it doesn't matter what the human did with keys or didn't do, it's a computer that sends the signals.

Yost: The thought that it might be redirected to a high-value domestic target, it would…

Schell: Yes, that's what I was saying. Yes. And I said, there's things that you're claiming are going to prevent that that are absolutely nonsensical. They don't work. I won't sign off. Well, this got the usual sort of; well, you can't say that; you've got all these Ph.D.s and everything; they all know the answer. Well it isn't true. I said, if you don't want me to sign it, you know; I'm not part of this project. But if you want me to sign off on it from the standpoint it's adequate; this accurately represents what the SAGE system does—it doesn't. And again, back to my boss. As you imagine, he and I had a fairly good

relationship. And they said you know, we don't really need to have Captain Schell sign off on it. He said, no, you do. And he said, "It's not going to get my signature if it doesn't have his." And so now they had a problem and so we spent a fair bit of time; and at the end of the day, I said well, you're not going to rebuild the system, but you are going to accurately portray what the problem is. And you've got a problem by your standards, that you don't provide, you couldn't provide nuclear safety because it's just controlled by a computer. And so that was my first encounter, I think, with computer security as an actual operational issue and we didn't have a solution, *per se*, but at least we're identifying the problem. And the problem was that you couldn't effectively evaluate. People wanted to be able to inspect a program and claim we can say with assurance that this only does this **and nothing else**. And I said, you simply can't do that. We simply do not know how to do that. And at that time I didn't have the understanding of scientific formulation as to why. We just knew that as a practical matter, I could have as many as a million eyes looking, but it doesn't matter. You cannot look at a piece of code and say that it will do this **and only this**.

Yost: At that time, what was your sense of adversaries' ability to exploit such a vulnerability?

Schell: Well, at that time we recognized that; and it would not have been popularly talked about. But the issue of an adversary sabotaging activity and putting things in the software that weren't supposed to be there was recognized as something that was very doable. It was actually dealt with, for example, in all the software development. They

were done in classified areas by cleared programmers. And the primary reason you did that was because you were concerned about an adversary being able to subvert the software, otherwise you wouldn't have done it. And so that was the primary defense was cleared programmers building every piece of software from the metal up. And an adversary could also be putting things in the hardware. But software is more malleable, so that was clearly the more attractive, from an adversary point of view. So yes, that was recognized as a major threat. I mean, you're talking now Cold War era; you're talking about the Russian capabilities, which were very substantial, technically. And given the opportunity, they could have; and we believe, would have. So all we can do is reduce the opportunity by having these essentially an air gap enclosed development environment. So, that was very much recognized as part; as you said, you had the clearance. And the only reason you had to clear everything was to deal with that problem.

Yost: The Philco programmers on the ICBM detection system and the transistorized [interrupted]

Schell: Yes, Philco did the Alaskan system, right.

Yost: The Philco programmers on the Alaskan system and the SDC programmers on the SAGE system, did they all have clearances at that time?

Schell: Yes, I think so.

Yost: They all did.

Schell: I think they all did. The Alaskan ones, I'm not so sure, because since it was dealing with sort of a second tier of the operational impact was, you know, more bounded. But, I believe they did. The SDC programmers did have clearances.

Yost: Can you provide some context about the war in Southeast Asia and how that was impacting such systems and security?

Schell: Yes. So, the war in Southeast Asia was ongoing and, of course, when I started in the missile timeframe, it was just an assistance activity. There wasn't really an active, major military operation. But as time went on, that obviously intensified so that by the time I was either wrapping up the backup interceptor, the replacement for SAGE, and were deploying that, we were suffering in Southeast Asia a fair number of aircraft losses, because the enemy launched Russian SAMS, surface to air missiles, that they had. Our aircraft being controlled and the controls that we were providing to our pilots were not the tactical air controls needed; they were not that effective. So one of the things that they wanted to have was to provide effective tactical air control. I was part of some discussions that said well yes, we do that here in the US in terms of our air defense. We're able to provide fighter control, tactical air control here and we could, in fact, provide that in the theater of war if you had suitable physical facilities; because these were not tactical computers. These were just plain old, you know, industrial computers. It just happened to be the Burroughs D825, a computer that the transistorized version ran

on. And I said, we could take that and actually install that in Southeast Asia. And they looked at the alternatives variously and the conclusion was yes, that was clearly, well, the only way to quickly get the capability there. And so I was the system engineer on the engineering end, and software engineer for that deployment in Southeast Asia.

Yost: And what was the time frame with this?

Schell: That would've been 1965. And so we took, literally the BUIC system we'd done in the U.S. Obviously, we engaged SDC to do the reprogramming necessary to put this, you know, in a whole different environment. And just essentially picked one of our BUIC sites and put systems in Southeast Asia, and provided the tactical air control. And it was, I think, highly effective in terms of reducing fighter losses because we could in fact get information, and control them, and manage that. And was just sort of a straightforward picking up the control of the SAGE system, as opposed to the air defense portion, taking the control of the tactical control, moving that to Southeast Asia. And so my job there in terms of software was primarily just getting the specifications, moving that forward. But in that same time period, they observed (and I was not part of the strategic assessment there) but the information was the intelligence community knew information that could help us avoid things like SAM sites and other sort of threats. However, that was not made available on a timely enough basis to actually be used to control the tactical air. And so that was a problem. Well, it couldn't be sent, obviously, because it was sensitive intelligence data. And so they said well, how could we possibly provide some version of that intelligence data that we could use to control this on a timely fashion. Now, it was

being done manually; and they were taking it and manually transferring it; and of course it always— or frequently —got there too late. And so I said well, we need to have an automated way. You may have a human in the loop or not, but you have to have a way of automatically transferring that data; making that part of the same tactical control. Well, since I was responsible for tactical control, I also became an assistant engineer responsible for; and then the project manager for that and software for that interface to the intelligence community. So, at that point; that was the first point that you might say we had to have a full, multi-level system in which I had highly sensitive intelligence data being processed in the computer; and the output, some of it, would come out as unclassified. And so you had Top Secret, Sensitive Information, to Unclassified out of the same computer and that was the proposition. And the answer, of course, was no, we don't do that. That's never been done. The people who were in control of intelligence resources said no. We can't do that. And so part of my job in the Air Force as a consumer of that intelligence, was to interface with the intelligence community, which had a different perspective than we did. I mean, we had a war to fight and fighters to protect; and they had sources and methods to protect. So I became, at that time, substantially involved in terms of understanding what the sources were. My problem was to build and deliver this system in a short period of time. Well, we understood that to do that we were not going to be able to take the software and the hardware that we had been using for our air defense system. But again, the issue of subversion was a dominant issue. If you look from an adversary perspective, that was going to happen. And in the intelligence community, they did sensitive processing. They actually had their computer hardware as well as software built in classified environments with classified specifications, and with

more just built with designs instructed from the ground up.. And, of course, in Minneapolis, they built those things with special computers. Those computers were not for sale outside that community and so I became introduced to that community. And, of course, everybody always starts by saying well you can't do that. But I understood what they were; learned about their instructions; gave the ESD (pause).

Yost: Were there guards against electronic emanations in computing equipment prior to that?

Schell: Yes, very much. So that was a multi-level system that everyone had basically declared could not be built. And it turns out the intelligence community had a set of essentially evaluation criteria for software; of what they considered good software practice for doing that, as well as the hardware. And everything was built, including the hardware and all the software; cleared program; highly cleared programmers, in a closed environment. That was a major part of it, along with the software standards and criteria, which I had looked at; which were probably informed by some of the things we had done by the configuration management work I had done earlier. But, you know, I wasn't particularly impressed with it, but it was better than ignoring the problem. And so that multi-level system was a challenge and obviously in a short period of time — you know, a war is going on. And I successfully brought that through to completion, deployed that, ran that, and it did in fact run as people claim. I have no way of knowing, but probably the first really full multi-level system that ran with intelligence data connected to an open

unclassified environment. And we had to program to those standards in everything that we did. And so that was probably the first system.

On a personal impact of that; the Electronics Systems Division was part of the Air Force Systems Command, at that time, which had the headquarters in Washington, Andrews Air Force Base. And every year they designated an outstanding junior officer for each division and then for the command. And it was always the case that the Outstanding Junior Officer of the Year for the Air Force Systems Command was somebody in the aircraft or missile business because that's what the Air Force is about. And I was the first junior officer that was awarded the Outstanding Junior Officer of the Year for the Air Force Systems Command from Electronic Systems Division, which is about electronics and computers. And that was based on the work I had done with building and installing this multi-level system in Southeast Asia that quite effectively protected our forces there.

Yost: In addition to the personal honor, did that have an impact on how ESD was seen within the Air Force?

Schell: I don't know to what extent that particular program did, but ESD was certainly seen; electronics and computers were seen as increasingly important. The rank of the commander, you know, had a star added, and such. And so, in fact, the rising people like General W. L. Creech, who went on to become the Commander of the Tactical Air Command, was the commander of Electronic Systems Division for part of the time. He

was obviously a rising star in the hierarchy. So the importance of electronics and computers was more highly recognized.

Yost: So this takes things through the mid-60s.

Schell: Yes, the late ‘60s; it’s at that point, yes.

Yost: In October 1967, RAND’s Willis Ware and NSA’s Bernard Peters presented a paper entitled “Security Considerations in a Multi-Program Computer System” at the spring joint computer conference. Were you aware of that paper?

Schell: Not at that time.

Yost: Not at that time?

Schell: No.

Yost: And were you aware of such work before the report of the 1970 Defense Science Board came out with the Ware report?

Schell: You know, it continued to exist in military leaders at that time, that the research community, which would be concerned with that, and the engineering and acquisition for our operations are really a long ways apart. And whenever somebody would talk about

research activities, as an engineer, I was well trained; if you got a researcher and they wanted to have him join my group I'd say yes, there's the door— out —because you don't make research part of what you have to deliver. Because it's research. And so those things were things that I wouldn't even have had a lot of interest in because, you know, the Ware Report was—if I had known about it, which I didn't—it was, you know, so what does it tell me I can do about this problem, right? I mean, nothing. There was nothing I could use there. And so consequently, no, I was not aware of those and wouldn't have been interested. I mean, somebody might have talked about it; whatever; I wouldn't have been interested. I wouldn't have read it.

Yost: Does this basically take you through to the point where you returned to graduate school, or were there other things you worked on before you went to M.I.T.?

Schell: Well, there were, you know, a variety of things I worked on, you know. (Laughs.) But, yeah, that's probably the major things.

Yost: With regard to computers?

Schell: Right, with regard to computer security. So the Air Force periodically had this problem about having qualified engineers that were organic to the organization. And so when they had an engineer they wanted to get you to go to graduate school. And I have never enjoyed going to school; and so they would; I would; almost seemed to me like annually, would get this "opportunity" to go to graduate school. I just threw them out.

Not interested in going to school. I just never enjoyed being a student. And finally, they got more aggressive and somebody, I don't know where in the personnel system, decided that they'd state it differently. So, "it is in the interest of the Air Force" that you go to graduate school. And you have to decline this offer; otherwise you will be assigned to graduate school. The reason they couldn't order you to graduate school was you had a three-for-one active duty obligation for every year you spent in graduate school, you had to have an additional three years of active duty. Indentured servitude doesn't allow you to do that without you having to agree. And so they'd ask you to agree. But the thing that you said; and you had the opportunity to disagree by saying essentially, I recognize that I'm rejecting this opportunity to contribute to the best interests of the Air Force, yadayada. I showed it to my boss and he said, well yes, they're essentially giving you a choice. You can either resign or go to graduate school because you will have no career if you have that in your record, so you decide. And so that was a decision that I really had to face; as sort of the act of decision; as to whether or not I would agree to go to graduate school or not. And since I disliked being a student I said well, I really don't want to go just any old place to graduate school. And so there became a fairly extended period of time in which we arm wrestled about whether I was or was not going to go to graduate school; and where that was going to be. And since it was electrical engineering, I could only go to an electrical engineering program, not computer science. They had more computer people than they needed, but they needed the engineering background. But I'd obviously gotten involved in computers; and so I was among other things, interested in where computers were taught in the electrical engineering departments. And I had an interest in M.I.T. When I graduated from high school, I'd been accepted to M.I.T., and

for financial reasons wasn't able to go there. And so I thought; obviously many years before, decided that was a place I'd like to go. So I said I want to go to M.I.T. And we played through this bit of drama with the personnel system. And at that time what they told me was we have three tiers of colleges that practically exist. And there's the small upper tier, places like M.I.T. And there's a second level, not quite as small, places like UCLA, and Purdue, and places like that; and those are good schools. And we've got, you know, a lot of the state schools, and such, which is where you did your master's and bachelor's. And they said if you were really an exceptional person we might move you one step up, one tier jump. And so, since you went to cow colleges, you could go to the second tier. No, you're not a candidate to go to the top. You can't do it. It is not a possibility. And, of course, their reason was that people that make that sort of a jump are unlikely to graduate. And so you've wasted our resources by sending you someplace that you're going to fail. Not to mention the fact that you probably can't get in. But, I said, that's the only place I'm really interested in going. So we arm wrestled and they resisted; I was at a cocktail party with the commander of the Electronic Systems Division when this was going on. He came over; congratulations on your Junior Officer of the Year; that sort of thing. I said yes sir, you know, that and a quarter will get me a beer at the Officer's Club. He said what do you mean? I said, they tell me I have to go to graduate school and they say I'm just not authorized to go to a good college like M.I.T. And I said so what good does it do me to do a good job when you say I'm not qualified. Well, he says, that's just not right. And his aide was there and he said, remind me I want to call the commander assistant's man and see what's going on. Well, from a cocktail party discussion you don't expect to hear the next day. But the next week he did follow

through. So I was what they called a “13”, which meant that the maximum score you got on your performance reports were a nine and the maximum graduate grade point was four; and so that’s a 13 when they talk about their system; you know, just a shorthand. Because I’d had a 4.0 on my master’s degree, and I had top nines on my performance reports

Yost: So as high as you could get.

Schell: As high as you could get. So 13, that was it. And they understood that jargon, at least in the educational community. And so he called up the commander, at Systems Command, and said, you guys; you know, we won the Junior Officer of the Year; our Junior Officer here in your Air Force; and the people down there in Air University say he’s not qualified. And I’m not sure I understand that. And they say he just can’t even be considered. And the four star commander of the Systems command said, well, that isn’t true. He says, anybody can be considered anything they’re qualified for. Well, my commander said, it didn’t happen; so what’s the story. Well, he said, I’ll check into it. So I got a call from the Air University about my tinkering with their system. They really didn’t want this to happen. But I got a friendly call from a fellow Air Force officer. The guy called me and said this call never happen. But he said, you’re going to get a call with all the reasons for why you want to accept these other schools—UCLA, wherever they’re sending you to—but if you choose not to accept it, then they will have to take some action. So I got the call and I said no, I won’t accept that. So now they had the problem; you know, the good graces of the commander. And so they said well, okay, but you know

you have to get admitted. So I had two, three months from nothing to get admitted. Well, obviously, a normal admission process doesn't do that; nobody's going to cut any corners. But they could move things faster through the system. And they did. And I didn't know what would happen; I didn't get accepted to M.I.T. until maybe two weeks before the start of classes; and I already thought I was moving to California. I had sold my house.

Yost: What year was this?

Schell: '68. Yes. So that was the primary benefit I got from the Junior Officer of the Year was it served me well to get me into the graduate program I wanted in to. The nuclear engineering professor at Washington State that I'd taught FORTRAN for was valuable because, of course, being in the Air Force I had kept no contacts with the academic community and M.I.T., of course, wanted an academic reference. And I was like wiped out; like, how am I going to get an academic reference? Right? (Laughs.) Impossible. But you've gotta have one, right? Just gotta have one. No choice. And so I remembered this professor that I had taught FORTRAN for; I thought well, I helped him out. So I looked him up; he was still there; called him up. Well it turns out when I had left that he had, in his nuclear engineering class, liked my lab reports. My wife was a good typist; she typed; and I drew my curves with India ink because I, you know, took engineering drawing and that sort of stuff. And so when I got done at Washington State he said, are you going to do anything with your lab reports? Do you keep them? I said, I keep them but I don't do anything with them. So he said well, can I have them? I agreed.

So I called him up and as soon as I said my name oh yeah, he said, I still use your lab reports as samples to the students to show them how they ought to prepare lab reports. Oh sure, I'd be glad to give you a recommendation. And so my FORTRAN course served me well (laughs) from having done him a job.

Yost: Great. At the time were you aware that the FSQ-7 computers of SAGE grew out of Project Whirlwind at M.I.T.?

Schell: Yes.

Yost: So you knew about M.I.T.'s major contributions to computers and computer networking at that time?

Schell: Yes I did.

Yost: When you arrived at M.I.T. what was the state of the Multics Project and were you involved at all in that project?

Schell: The Multics Project was at a relatively early stage. I wasn't involved. I came looking for a topic; thesis topic to get done. And so I went looking for an advisor, and among other places went to Project MAC. The professors in Project MAC taught the early systems programming sort of courses that I was interested in. And so it was a place that I knew about, Project MAC. I was just a graduate student looking around. Obviously,

I looked at AI; looked at the whole range of things. But the CTSS, the time sharing system, sort of the predecessor of Multics, was up and running as part of that community, the computer support that they had. So that was an active project. Multics was emerging at that point. They had a machine; the GE645s were running; they had the capacity I remember well, of 16 users as the maximum number of users. So if you wanted to use it you had to be one of the first 16 to get on because that's all it supported. And so I went looking for a thesis topic and I did talk to the Project MAC people, and other people; and Bob Fano , who was, you know; obviously electromagnetics. And since that's where my background was, and such, I went and talked to him. And one of the things that I thought I was interested in when I came to MIT was security. And so I had a couple of the professors say okay, look, you gotta decide as a student. You've got one of two goals that we find students have. Some students like to be a student forever, and it goes on and on; and security's a good topic for them because there's no answer. But if you want to get done, it's not a topic. Understand that computer security was interesting to me because I had finished up this Southeast Asia project and believed there's an important problem.

Yost: Which faculty members were sending you that message?

Schell: Oh, essentially all of them. I mean, certainly Bob Fano, and Fernando Corbató, I guess would have been; and Michael Dertouzos. And so, they were all basically agreeing, you know, one way or the other; and some of them more. I guess Robert Fano was probably the most blunt about it; just put it the way I sort of put it; if you want to get done, you don't do that. And so I just gave that up but I was interested in operating

systems because, after all, if you think about it, of the systems I had worked on, none of them had operating systems that came with them. For them all operating systems were part of what you had to build. And what I had encountered was that the operating systems were a major cause of failure. One of the hardest problems, seemed to me, just engineering wise, was building the operating system that had good performance and good reliability. And we succeeded in doing that. In the BUIC system we had a multi-processor system; it was one of the first, if not the first major command and systems that had multi-processors. And obviously, the contractor, Burroughs, had put that together, but I'd been deeply involved in the operating system design and influenced the design decisions since it was essentially designed to support our particular activity. And so I had a lot of interest in operating systems, because I had been involved in that. For the one we built from scratch for Southeast Asia, I was essentially the architect for much of the operating system because there was nobody else in charge. I mean, there wasn't an operating system, really. It was here's a piece of iron; here's a problem; yeah, we got a lot of examples of executives and stuff. And so, that was an area I was interested in. So I fairly early on, since security wasn't a thing, went off and looked at operating systems and that naturally lead to Project MAC, and CTSS, you know, understood what that was; and then Multics, which was ongoing. And so I joined Project MAC, with the hope of finding a thesis topic there. And I did. And my thesis topic was addressing the question of dynamic reconfiguration — of adding in or moving hardware modules while the system is running. And I remember some of the people on the programming staff saying no, you can't do that. You can't; you've got two processors, you can't take one away and add back

another one; and you can't take a memory away and add that; no. It can't be done. That's what I thought was interesting. So that's what my dissertation was about.

Yost: And was Fano your primary advisor?

Schell: He was not the primary advisor, but was on my committee. It turns out that Corbató was who I wanted to have. In some sense, you know, he'd done CTSS as an operating system. Clearly, that's the place. But, he was, I think, fully subscribed. He had a fair-haired junior professor, Jerry Saltzer, that was his student and I think I was Jerry's first student. And so he had; you know, being a junior professor, had to find a student, bring him through that process, right? So I think I was Jerry's first graduate student.

Yost: And when you were exposed to Multics at that time, were you aware of the security features that had been programmed into the system?

Schell: Yes, very much so. You know you started by, of course, looking at the 1965 papers that the joint computer conference had on Multics; four or five papers about Multics. And then there was the paper on Multics Ring that was done by Graham, and things like that. And so, yes, I was quite familiar with that; and discussed those things. From my point of view, having come out of the military background where they said, you know, the determined adversary is going to primarily concern himself with subversion and that kind of thing. What they were dealing with struck me as security features that were interesting, but they didn't do anything about addressing the problem that I

would've seen from the military point of view as the main threat that I had to face. If I were going to attack a computer, that's how I would attack it; is from the subversion point of view. And the thing that they were doing wasn't dealing with subversion significantly. So I was aware of those but since I had been well advised to stay away from the security thing, I didn't involve myself particularly. It turned out that in the course of my research work in the operating system, I discovered, as typically a hacker would today, some flaws in the security system, which I brought to peoples' attention and, you know, took the usual amount of sort of hacker joy in trying to show somebody else "look what I can do." I can get on to anything that's in the system and I can control it totally. And so I found at least one major flaw. But I didn't give any further real attention to that. And the only other place that security showed up was when I did my oral exams, Bob Fano was on my committee and after his good advice to avoid security, well, he ended up asking me several security questions, which after telling me not to think about it, (laughs); so I didn't think was quite fair.

Yost: Questions coming from left field.

Schell: But I knew what the security features were in Multics and I understood, but it wasn't an area of concentration in any way. It was just one of those things.

Yost: And those flaws that you identified with a hacker mentality, did those lead to changes?

Schell: Oh yes. They fixed them; they patched them. Sure. Because they understood the value proposition of Multics is that it's a computing utility. And that's the whole thing that Multics is about. If you go and look in the 1965 papers, the whole notion is what's different about a utility? Well, one of the major differences about a computer utility is it has to have security. And even though, at that time, most of the mainframes, and even the time sharing systems, were used by a community that was at least cooperative, in some sense. And when you talked about a Multics notion of a computer utility, you're talking about computation that's out there, available as a utility and adversaries are using that utility. And so, yes, security was one of the major distinguishing properties of Multics.

Yost: Did you get the sense that that was the environment at M.I.T. and their contracting work with the military had an influence, because the Dartmouth Time-Sharing System, as I understand it, didn't have any security features?

Schell: No, the military wasn't; you're talking '60s now. There was no love lost with the military. And so, no. It was not driven (laughs + pause)

Yost: And Edsger Djikstra was a visiting faculty member when you were there. Can you talk about any interaction you had with him?

Schell: Yes, he was one of a couple of people that probably had a major influence on my perspective on computer science. You know my background was FORTRAN. I mean, that's what we used. The military programs were either assembly language or

FORTRAN, or JOVIAL. JOVIAL was a military program language—sort of like FORTRAN. When I joined Project MAC; well, one of the things that you do with graduate students is they're free labor for various things that have to be done. And one of the projects I had was what we called the Bootstrap Project in Multics, which brought it from essentially the bare iron up to the point where it could begin to run. It had been written an assembly language. And in Multics, the whole story was to write things in a higher language. That makes them inspectable and viewable, and that's what I was working on, looking at that problem. Okay, rewrite Bootstrap in PL/1, from assembler language. About that time Djikstra came to campus and he taught a class. And I took his class and his claims, of course, seemed to me pretty outlandish claims in terms of the ability to know that essentially you had, call it, bug-free programs. And I, of course, had read the T.H.E. paper that he had written about that and I was largely a skeptic. Well, it turned out his office was about three doors down from mine. By that time I had gotten an office in Project MAC; they had a cluster of graduate students that I was with. Down at the other end was Djikstra; they'd found a place for him. And so I'd go down periodically and sort of challenge his claims. Well, he was not a particularly patient person, and obviously, I had a lot of respect for him so you wouldn't debate him as you would another graduate student; but still try and learn where he was coming from. So I sort of threw out to him; because his examples; although the T.H.E. was about an operating system, when he taught his classes they were about algorithms that were essentially applications. And I said well, I'm not sure that I am persuaded that I can get this approach really to work; he described it as putting beads on a necklace and you put them together, and that sort of thing. I said I'm not convinced. So he said to me, what are you

working on? I said I'm supposed to rewrite this assembly language thing in PL/1. And he says well, you understand what I do? Well, I think so. Well, you try it. And so, I said okay, fine, I'll do that. I'll modularize it that way; I'll introduce those sort of layering constructs; and when I had questions and such—it isn't hard, and not a science as to how to construct that. And occasionally I'd say I don't see how to do this, so I'd go down and talk to him and say well, okay, I don't see; you know, it doesn't seem to me like this goes; and he'd patiently, or impatiently, put me on the right track. And so I did that and produced that program in that way, and of course, with a Bootstrap program, you can't do much more than a desk check and then go run it. And so you'd have to construct a new boot tape and it booted from tapes. You'd create all these files, then you put them out, and they had this process of creating these tapes, and that was the job. And I was familiar with that process; I had worked in Multics; I had learned how to boot things up. I had written at that low a level. So when the system would come up, if it incurred a flaw early on in the process, it would type that the operating system was dead; this bombed; this is not here; you'd get some sort of very cryptic message that the operator console—which was a typewriter—would type out. And so I put my PL/1 version of the bootstrap program, built the tape, and so I'm going to run my first test, and well, it's where it's going to crash first, right? I mean, the usual kind of question, is can I know what happened? So I start the tape running and went to the operating console and waited for the message to type out. And I waited; and I waited. And I said oh boy, I'm in trouble. The thing; it isn't even able to give me an error message or anything. Nothing happens. I'm really in trouble! And I turned around and looked back at the processor. Well, when the system's fully up and running, the background thing will flash the lights and a pattern

on the console lights. And I looked back, and it was running. And, I said, what'd I do? Run the wrong tape? So I looked down and made sure I had my tape. I booted it again. What? And it came up! Well, you know, from my FORTRAN experience I'd never written a thousand-line program (which this was) that ran the first time, right? It doesn't happen. And so that was one that definitely changed my perspective from a software engineering point of view, in terms of Djikstra's impact, which is your question. Yes, it had a definite impact on the way I view software engineering; I became obviously very much of an advocate of that way of thinking about the problem.

Yost: And beyond just Djikstra and M.I.T. there was a growing interest in software engineering in the late 1960s; there was the famous NATO conference.

Schell: Yes.

Yost: Were you aware of that research and did it influence you?

Schell: I was aware of it; and having been at the M.I.T. environment, though, I was reading what I would not have read otherwise in the Air Force. So yes, I was aware of those; and the NATO conference in particular. I had heard about it; don't think I knew anybody that had been there, or whatever; but yes.

Yost: Are you ready to take a short break or do you want to go on 'til noon?

Schell: Your choice.

Yost: Let's take a quick break.

Yost: We were finishing up talking about completing your doctorate at M.I.T.

Schell: Yes.

Yost: Were there any other major influences that we haven't discussed at M.I.T. that we need to talk about?

Schell: Well, I think Jerry Saltzer and more indirectly Corbató, definitely had an influence in terms of thinking about computers and the whole concept of the computing utility is one. That is not the way from my mechanic control background that I would normally have thought of computers, and so yes, the fact of a computer serving multiple people and the ability to think about things in a structured and orderly way. I mean, Jerry Saltzer was a product of M.I.T. and very much, you know, emphasized that the process was important, not just getting some result but being able to explain and understand what was going on. So, yes, M.I.T. definitely changed the way I looked at particularly things in the academic arena. I'm in a dramatically different environment than what I had experienced in my previous education. When I went to Montana State, I carried around a slide rule in a sling, along with all the other engineers in a classical engineering school. And at Washington State, it was a similar sort of engineering thing. But at M.I.T., nobody

had slide rules. All the answers were small integers. And it was all about the process; show your work. To take an engineering exam when people were not sitting there just cranking out numbers, was not the way I had thought about things before, academically. So it was a very different way of thinking about the world. I guess the other thing; the area, in terms of the studies that I had not been familiar with that I got familiar with at M.I.T. was the issue of computability; and the Turing thesis, and things of that nature. I'd just not done that. I'd played a game when I was on the BUIC project with one of the SDC engineers and I was familiar with recursion. But I didn't understand a lot. I understood Ackermann's function, and things like that; but I gave one of the SDC engineers, who would sort of claim well, I can program anything. And I gave him something like Ackermann's function to program on FORTRAN, because that was all we had. He spent several months trying to figure out how to get this to run. I give him a number and I expect the answer to come out. And, of course, if I give you a sufficiently large number, in a recursive function, trying to do this using FORTRAN, I'm going to run out of something, right? Because FORTRAN, at that time, didn't support recursion, natively. And so he's building all of this infrastructure. And finally his boss, who came over to see me one day, and he says, I don't know what you did with Jay, he's wasting all of his time; he's not getting his work done; you know, turn it off. And so I said to him, I said look, what I gave you is a recursive function and you don't have the tools in your programming background to deal with a recursive function. So I said, just give it up. That was the extent of my knowledge of that, it was just something for entertainment, more or less. And then at M.I.T. to actually understand the issues of computability and the fact that to an engineer it is really counterintuitive to the engineer, where they say "there does

not exist an effective procedure for taking a set of code and producing a specification for it." An engineer who had dealt mostly with hardware; and if you give me a piece of hardware, I can bloody well tell you what it does; I can write the specification. I can reverse engineer it; I can give you detailed specification for it. The fact that it doesn't, and there's never going to, exist the ability to do that for software, was something that to an engineer is just counterintuitive. And so that was something that didn't come in a flash, it came after working with those concepts over time. You come to understand yes, this is different. And even though the roots of non-computability are because of infinity, at the end of the day it is nonetheless, practical. The number of states in a program are sufficiently close to infinity that it doesn't take it very long for a program to essentially have the results apply. (Laughs.) And so that was probably, out of the studies, the thing that had the biggest input.

Other things at M.I.T. that I can think of; I did my minor in the Sloan School of Management. So, since I had been in the management business, I found it interesting to see how the management school was taught, since I'd never been, academically in a management school. So that was interesting.

Yost: Did you feel you learned things that you later applied from the Sloan School?

Schell: Yes, definitely so. I think, you know, one of the things that they did, whether deliberately or not I'm not sure, but they would contrast their somewhat quantitative method of management to, say, the case study method that's done at Harvard. And it

turned out that while I was at school, I had a next door neighbor that was studying at the Harvard Business School at the time. And so we would compare about his case studies and at M.I.T., we were trying to provide essentially quantitative approach. At that time, at least, there were people that were heavily into providing models, essentially, of processes, that would be management processes. I'd never been persuaded that one can do that very faithfully; one can't measure baloney with a micrometer. But nonetheless, it was a way of thinking about the problem that I had not thought about before, so that was something new. Also the use of computers, which was beginning to be more emergent in management activities, became things like Microsoft Project, and things like that. Precursors of that.

Yost: And you completed your doctorate in 1971?

Schell: Yes.

Yost: During your time at M.I.T., there wasn't a lot published on computer security, but there were a few things, including Ware and Peters' 1967 paper. Were you not following that scholarship?

Schell: I was not following that. I'd taken my advice to stay away from security. I was interested in operating systems. So to the extent that it would show up in things like the ACM's special interest group on operating systems, those papers I would read and understand. But no, otherwise not.

Yost: What did you anticipate would be your career opportunity in the Air Force, coming out of M.I.T. with this degree?

Schell: Well, since I'd been in the acquisition business and had done well. I expected that that's where I would be. I mean, after all, the Air Force had sent me to get a doctorate in electrical engineering—from their point of view—the expectation was that there would be a payback tour in the electronics area. You know, probably computers, as a practical matter, because I had made it into the computer science area. So I expected to be back in the acquisition business someplace. Since in the Air Force, the airplanes and missiles are the real things, I looked for some opportunity in either the space missile area, or the airborne area. So, Wright-Patterson [AFB], or Vandenberg [AFB], or some place of that nature. And, you know, probably project management of some sort or another. I was interested in some of the operational opportunities of using these things, and that would've been fine, as well. But didn't know where that would go.

Yost: And when you did graduate, I assume that the Air Force assigns; but do you have a say in your assignment?

Schell: People would say that I had much more say in my assignments that most people, and that's probably true. I think nature abhors a vacuum, so when somebody has resources and they're not in control, I may take control of them, and that includes the personnel system. Yes, my assignment area has definitely been not typical. My son, who

was in the Air Force for several years, kept pointing out that no, it doesn't work that way. But my first assignment, when I finished at Washington State, I got it almost by accident. The Air Force has something called regular officers and reserve officers, but this may not mean anything to you, particularly. But the reserve officers may come and go and maybe get put back to the reserve though it doesn't actually happen that way. The reserve officer training corps (ROTC) was for reserve officers, and the regular officers come out of the academies. They would offer distinguished ROTC graduates regular officer commissions as well. And the advice was that if you took the regular officer commission, you didn't get a uniform allowance, you didn't get a set of monetary things. So they always said turn it down when it comes to the offer, because you will get another offer. It doesn't matter to you, they will offer it to you again later, so just turn it down. So, while I was at Washington State, I got the usual expected offer of a regular commission. And I said well, I'm in the Air Force, I'm just going to take it. No, nobody expects that, but I'm just going to take it. So I did. I completely screwed up the personnel system. It's because you resign your reserve commission and so I was out of one part of the personnel system; I was out of the system. And so I was within a month of graduation and there's somebody that just sort of hangs around and checks things; I mean, nobody has the job of monitoring you. But there are a number of Air Force students there and the senior guy, has the job of checking on them. He said where you going? I don't know; haven't heard. Two weeks away from graduation he said, you still don't know where you're going? Not a clue. He said that's not right! So he just picked up the phone and calls down to an area of the Air University, and he got the sort of answer, "Uh, Lt. Who? No, we don't have any assignment notice for him. No, he's not; no, who is he?" So well,

although they'd been sending me my paycheck every month, the fact is that some part of the system had lost track of me because I resigned my reserve commission. So I was two weeks from graduation and nobody in the Air Force had ever thought about where I was going. So they said to me, where do you want to go. They said, you got be in your field; you're going to be an electrical engineer; 28 is the field number. I said what are the choices? And so they whip out a map; we looked at it; okay, I said, I want to get as far away as I can. And Massachusetts, up there in the far northeast, looks like about the farthest away. So there, Hanscom Air Force Base; I want to go there. So that's how I got to Hanscom, it was because the personnel system lost me. And I was almost in the same situation coming out of M.I.T. I don't have as much knowledge about what went on, but for whatever set of reasons, I didn't seem to really exist; I didn't have an assignment notice. I probably checked a couple months before; this time I checked a little earlier. So I went looking around for where I could go. I had a fair bit of influence, at least with candidates of places to go. And one of the candidates was Hanscom; and of course, since a number of the people there still knew me I probably had people who were interested in making that match. I didn't really want to go there, because I'd been there and I wanted to do something different but at the end, I got sucked in. And when I went there for the interview, if you want to call it that—when they say okay, you're likely to go here—the deputy commander that I talked to wasn't responsible for the class of acquisitions for electronics and such. And I said, you know, I think what I can do is I can assist with implementing management information systems because of my Sloan School training, and work with computers. I said, you know you're manually typing RFPs, you're manually keeping track of financial stuff. We can do a lot better than that. And he agreed.

I gave him some examples of what I thought we could do, and so I thought I was going there to put together a management information system, working for the deputy directly, sort of on the staff. Thought it would be neat to go off and play around with management information system. So, two or three weeks later I actually showed up and reported to his office. Okay, where do I start off on this management information stuff, where do I get a little background? And he says, well-l-l, there's been a little bit of a change, which, of course, is never a surprise in the military. Okay, what's the story? I said, I can do a lot of things. I'll show you some. Is it computer-related? Oh, good, you know; fine. And he says, the Pentagon has got this problem that they support the OSD, which does all this top secret stuff and they support the air staff does only secret—and they share stuff between them. They want to be able to run on a computer where they can share their files directly so the Top Secret guys can go get the secret stuff, and yet the secret guys don't have access to the Top Secret. And they think they've identified the problem. Their contractor has told them they've got a good solution down there for what they're doing and we've got the project. As the Electronics Systems Division, it's our job to give them a solution to this problem.

Yost: And who was the contractor?

Schell: Honeywell. It was a Honeywell computer. And so this is about providing computer security. Well, yes, that's what I think they call it. No, I said that's an unsolvable problem. No. (Laughs.) No, he said, it's a hard problem. The Ware Report had come out; it was the first time I'd really heard of it. He said we got this report from the

Defense Science Board; it says this is a difficult problem. One of the members of the Defense Science Board, or the Ware committee was talking about getting him on contract and helping us formulate a solution. I've got a guy up here, Dr. [John] Goodenough, looking over this, but it's not really his area, he's just sort of shepherding along. And that's your job. And I said, oh boy, now I'm in trouble. I've been given a job that has no solution. And this is not the kind of job that I want to have. And so that's how I got with my assignment out of M.I.T.—drafted as a non-volunteer for my job.

Yost: And you had a project management position to bring together resources to solve this dilemma.

Schell: To solve the problem. And the problem was, at that point, a localized problem. It was the Air Force Data Services Center that provides the IT support for the Pentagon, including the Office of the Secretary of Defense. This was for the Air Force, not the other services; they took care of themselves. But the Air Force provided support for the Office of Secretary of Defense. They were running on General Electric's 635 computers, and the contractor that they had working for them was under contract to ESD. The man to assess their security was Jim Anderson, and he was a member of the Ware committee. I'd never heard of him before, but that was the only active contract that I had at the time; and his job was to do an assessment of the 635's security for serving this purpose of running Secret/Top Secret on the same machine.

Yost: In the published histories that have been written, Anderson just kind of appears without a lot of background. Do you know his background?

Schell: Yes, I do know a little bit about it.

Yost: Could you speak to that?

Schell: He was an engineer at Burroughs and he before that was actually involved in some of the early work at the University of Pennsylvania, with Eckert perhaps.

Yost: At the Moore School?

Schell: And Anderson was involved in that meeting for the Pentagon; he was aware of it; he knew some of the people; that sort of thing. He was a naval officer early in his career. And he liked the military, had a lot of respect for the military, and had that set of values and point of view. Then he'd gone to work for Burroughs; where we had a common connection was the D825; he'd helped them design and build the D825 computer for Burroughs. And since it was a D825 that I'd used on the BUIC system, the backup system for SAGE, it was a common machine we understood a lot about so we could have a beer and discuss the architecture of the D825. And then also, at Burroughs, someone who he knew well and associated with was Ted Glaser. Glaser was more of an [computer] architect at Burroughs, and he also was on Multics Project. Glaser was one of

the co-authors on one of the early Multics papers. And so there was sort of a set of people that [interrupted]

Yost: So had Glaser had been at M.I.T.?

Schell: He had been associated with the faculty there. I'm not sure whether he was a professor, an adjunct, whatever; I'm not sure. I think he would've been associated with Project MAC, as part of the project; he probably already had his doctorate at that point.

So I understood Anderson [had] been around and worked in the intelligence community area in the security area for several years. NSA had their own network that was essentially a mini ARPANET that they used internally. And he had, for years, been the primary security architect for the guy at NSA that ran that. And so that was a pretty steady job. After he'd left Burroughs, he'd gone out on his own and he'd worked for Auerbach working on publications for a little while; and then he'd gone, essentially, into consulting, primarily in the intelligence community. So he had worked in both the CIA and NSA. Those were security related—computer security. And he was on the Ware Committee. And he was part of a project that many people have talked about—you'll find references to it— the McDonnell Douglas Project; which was an effort at McDonnell Douglas to run, I believe, secret and then unclassified on the same machine. And they were the project team that repeatedly went through the process of fixing the "last bug" and then, of course, they would find another flaw, and again, fix the "last bug". And as I understood the legend related to that—I wasn't there—when they did their lastround,

they essentially installed a trap door; after they penetrated it, and they essentially told the sponsors that sure, you guys, this is a fruitless thing to do. That if somebody provides and subverts the thing, no amount of your penetration testing is going to discover it; and here's a demonstration of what that is. So he'd been in that kind of security business; that's why he was in on the Ware Report, on that group because he'd been in the intelligence community. Probably one of the ones who thought about computer security as computer security. You know what I mean? That wasn't Willis Ware's field, as such, at the time but it was Anderson's field and that's what he'd been doing; that's what he'd done; was computer security in the intelligence community.

Yost: Okay. Do you know the time frame of the McDonnell Douglas Project?

Schell: Yeah, that would've been about the same time as the Ware Report, you know, late '60s.

Yost: To your knowledge, was that the first time that a penetration study was conducted —installing a trap door and trying to find it?

Schell: It's a little hard to say; even by then, as I'd indicated back in the SAGE days, we understood that that was the real threat. Was there anyone that had actually done it for demonstration purposes, I don't know. I mean, in SAGE, I would've said why bother? I mean, it's obvious that if I install a trap door, I own the system. That's why we have cleared programmers. No, nobody ever built one that I knew of, but it's obvious.

Yost: And so you're gathering a group and James Anderson's one of the members, and Ted Glaser's another?

Schell: Yes. So we had this contract to look at the Pentagon. That was a Jim Anderson contract. So I joined him in going down and looking at that. And of course, I knew nothing about computer security, really. I just; hadn't been there. I had these other experiences, and I knew the problem but I knew nothing about the solution. So Jim Anderson was very much my primary mentor in computer security. And the first things we did were essentially penetration testing. And so the effort in the Pentagon, their [GE] 635, what he had done is he'd already done much of the penetration testing effort. And, you know, he took me with him and showed me what he was doing; and since I had to review and approve his reports, we would discuss. So we spent a lot of time discussing and I was learning at the time what this was about. Then (pause)

Yost: And was he leading a small team, or was he working on this himself?

Schell: It was primarily himself. He, throughout his roles, the last decades of his life, he always insisted he was a single individual. He totally abhorred paperwork. And when he understood that if he had a payroll he would get into a whole mess of things, he would never hire an employee, ever. And he would informally lead groups, but without ever having to ever have any administrative responsibility. And that was just, he said why he left industry, he said; he wanted to be his own boss. And so he had done that as an

individual. He did participate in teams. There was another team; and I think I sent you my paper on the Achilles Heel; and I reported on some of those early experiences. Anderson was essentially associated with most of those. There was the intelligence community computer that we went in; and he was good enough to take me under his wing and to be a mentor and so I participated very early on in that assignment of penetration test tiger team for that intelligence community that he was primarily leading. Glaser was there, you know; he invited me; so I was just the junior, neophyte, in fact, that he brought along. So I learned from the Pentagon; I then went to the intelligence penetration. In parallel with that, we had the problem that since the results out of the Pentagon were yes, what you have isn't good enough. Then the obvious thing at this point; you know, my management job was to figure out okay, what are we going to do about it? And this Dr. John Goodenough, who was in my same ESD organization, we were sort of the placeholder for Anderson's contract; I took over for him. He'd also been the placeholder for a broad study. You could say okay, I'll follow on, if you want, with you on the Ware Report, sort of broadly, conceptually framed that way. The organization I landed in had a fair number of research kind of activities. And so I was a fish out of water because as I said, the research guys; no, I don't want anything to do with research.

Yost: But even after the M.I.T. experience?

Schell: Yes. Don't let them foul up real things because they are researchers, you know? And, I appreciate what the researchers do, but if I've got a job to solve, I wasn't going to bring a researcher out to solve the Pentagon's problem. But, on the other hand, I knew it

was an unsolvable problem. We had this activity that had been launched and so I'm going to have to play researcher. I know nothing about researchor managing research. So what am I going to do with this? John Goodenough was a Harvard Ph.D.; he was a general researcher in computer science, but not security, that wasn't his thing. And my boss, triple levels up, was a colonel who had been in the nuclear research business, and so he knew about research. And so I was going around finding out about research, but the job [interrupted]

Yost: And who was this?

Schell: Colonel Gaines, Ed Gaines. He didn't know; computer security wasn't his thing; but he, for example, had advice. I'd listened around to people in the academic communities, and people worried about people stealing their ideas and things like that. One of the things he said, you know, researchers are always worried about people stealing their ideas. He said, wrong thing to worry about. He said, if you've got a good idea you can't give it away because it's different. There's really a herd mentality that exists in the research community and if you've got something that's really out of the box, you can't give it away. So he said don't worry about it. Because I had no idea what the rules were, should I worry about things? He said, no, just give it away to everybody; tell everybody; try; you know, you're more like a missionary than an entrepreneur. So I learned good advice from him. So, since Goodenough had started this notion of a panel; so since it's is an unsolvable problem, why not? So I brought together this group. And it was really a group that was to be headed by Ted Glaser; and Ted Glaser was the

chairman of the panel, but he was blind, and so mechanically it was difficult for him to take notes and things like that. And so Jim Anderson was the executive secretary so when it came time to writing a report, and since he had a company, he could deliver on a contract. Glaser didn't; there was no company to issue a contract; and we had a contract already with Anderson. And so I gave him the contract to run this Anderson panel. And their job was essentially that the Ware Report told us this is a hard problem; it didn't give us solutions.

Your job is to start with the Ware Report and tell us what the solutions are, and how we're going to solve this problem. I said it with a straight face, more or less, and go at it. So that's what they were doing, is trying to find a road map. The Air Force laid out schedule estimates as to how it is they might provide effective computer security.

Yost: And who else was on the panel besides Glaser and Anderson?

Schell: Well, Clark Weissman, who had done the ADEPT-50. Are you familiar with that? This was an IBM 360 retrofit for security that was done by SDC. And Weisman was at SDC at the time and there's a published paper on ADEPT-50. And so he'd been involved in the security business from that point of view. The guy named Dr. Eldred Nelson, from TRW, was on the panel; and he'd been heavily involved with Strategic Air Command Missile Targeting. And I probably encouraged his participation because he understood; you know, given my business with the nuclear safety and such, I said yes, this is part of the computer security problem is how do I know that the target that I think

I'm shooting is really what I'm going to shoot at? They'd been working that problem of how do you get essentially bug-free targeting programs. For years, TRW had done that and Nelson had been; I think he was a vice president in TRW; at least some senior level. So he was on the panel. We had two members from NSA; since when I had participated in the Southeast Asia project I had had to work with NSA, and they were the intelligence source that I was dealing with there. And Dan Edwards and Hilda Faust from NSA; and they were in a research group at NSA, so I hadn't encountered them, *per se*, because with my feeling about researchers when I was on the Southeast Asia project. But I mentioned that NSA had this essentially a criteria for software development that you had to do if you were going to do the secure software. And it turned out that Edwards and Faust had certainly; maybe they created, or had significantly influenced it; at that time I didn't care where it came from, it was just the gospel that we had to live by with NSA. But they'd been involved with security for a long time, that I had been unfamiliar with because I wasn't interested in research. But they were known to Anderson, so they came on board. And I think those were the primary active people; because if you get a copy of the report, their names are in the front.

Yost: What type of budget was there for staffing the committee and conducting its work?

Schell: Not much. It was a relatively low budget. They met; it was; there wasn't a program, *per se*, that this could be targeted against. So it seemed to me that the budget probably was a little more then than now, but I mean, on the order of a couple hundred thousand dollars. And, you know, you obviously had travel; had to pay for the people;

and it was a brain-picking exercise. These people didn't go away with a lot of research. It was the intent to pick the best brains around that knew about computer security and to bring them together and to hope for some synergy in the course of a meeting. And then Jim Anderson, as the secretary, would go and write up the results.

Yost: Then for him, this was his paid work?

Schell: They were all paid.

Yost: Okay. But was he working on it far more hours than the others that were?

Schell: Yes, because he was actually . . .

Yost: . . . was the writer . . .

Schell: Yes, he was doing the actual writing. Now, if you looked at Volume 2 of the Anderson Report, people like Clark Weisman actually wrote appendices that had their particular ideas. And, you know, in that sort of research community, he would essentially give them a budget and an assignment; a fixed price basis; so I if you look at Weisman's appendix, he probably spent significantly more hours than what got paid because it was a fixed price basis. And different ones participated differently.

Yost: Now, you came up with the idea of secure kernel and reference monitor, is that correct?

Schell: Yes. You know, this having been an unsolvable problem, I'd not spent a lot of time thinking about it. So in the course of the Anderson Report, you know, I listened to these people who knew what they were talking about go on and discuss, and just brainstorm. And people like Ted Glaser, who had already moved to Case Western Reserve University as a professor, or he was about to; he was an academic, basically at that point in his career. And Clark Weisman, who had built an actual more secure system in a more "researchy" structured way, in terms of ADEPT-50. So I came up with the notion, from my Multics experience, I said you know, if you have some part of the system that is actually responsible for and the focus is all on multi-level security. I mean, that's what the military is concerned about. I've got different levels and categories that I have to protect, and if I can have different pieces in a different way, I should be able to formulate and make that central part of the operating system know about the policy and enforce the policy. And then all the rest of the stuff that we keep despairing of ever being able to make behave correctly, I don't have to do. So very broadly, that was my notion and, you know, I had bounced that off of them; you know, for me, it was a wonderful learning experience to have these experts sit in these two to three-day meetings, and I can throw out ideas and since I'm the project manager, they can't blow me off entirely, right? And so, yes, I definitely got better than equal treatment, and could throw out ideas and interact and that was great. I formulated that and about this time, the IDA [Institute for Defense Analyses] at Princeton—you know they have a close relationship with the

government, of course, in some of the security areas—and a professor there, Dr. Stockton Gaines, put together a working group on computer security and he invited the experts in computer security to come to the working group. There was enough with Multics and other things that were going on; there were enough people talking about computer security. Within the academic community, there were people that wanted to come to this workshop. And there were more people that wanted to come than what he wanted. He wanted a small, you know, 20 or something workshop. There were a lot more people that wanted to come and he would just say no. And Anderson, I think, both IDA and Anderson had worked for NSA, that's where they had common roots, and so I'm sure they probably knew each other probably fairly well in advance. So I had been working with Anderson probably six months, or a year, as part of the panel. And so I bounced off of him and some of the others this sort of notion and this idea of this central thing that could actually provide this protection. And, you know, since I hadn't any chance to put any flesh on it, there was not much there so Anderson approached Gaines about having me attend, and Gaines said no. We don't need any newbies at this; this is a serious workshop. And so Anderson said okay, he said to me, go write; they ask for a quarter to a third-page description of the ideas that you wanted to put forth for the workshop. Anderson said you've been talking about this thing; why don't you go and write it up? Not a lot of substance you can write in a third- or quarter-page, but you can get the idea out there. You've been talking about it; go write it up. Well, I guess, that's more typical in a research environment, but you know, wasn't typical for me as a manager. (Laughs.) But okay. So I went away and I wrote up my position paper for that workshop. And Anderson took that to Gaines, and he said it's just fluff; which, of course, it was at that

point; no substance or detail behind it. And Anderson said well, I think the guy has something to offer us. In fact, Anderson said, if you don't let him come, I'm not going to come. And so, you know how things happen; what's one guy, you know. Alright. Since Anderson was sort of the senior security expert; once again as a mentor, he got me to the workshop; he got me into that. So that was a first expression, written down of what the concept was. I didn't know what to (pause)

Yost: And this was, you said, an IDA event?

Schell: Yes. It was Princeton or IDA; I don't know which hat he was wearing when Gaines did that.

Yost: Can you explain IDA at Princeton?

Schell: IDA is a government-sponsored research activity at Princeton [and a small number of other universities]. They do a lot of the cryptographic research, and stuff like that for the government. As a lot of universities have, they're one of their little captive activities. So I think Gaines was a professor at Princeton and also had a role at IDA.

Yost: Okay.

Schell: So one of the things when I went to write it up; so what are we going to call this? Well, if you go to write up a paper; if you're going to talk about it, you're going to have

to have a name for it, right? But if you're going to write it up in an academic committee, you gotta have a name for it. So I asked Anderson and he said, you know, call it the Schell Security or something. And so I got lots of wild ideas. I went back to the Col. Gaines, the old nuclear researcher, and he said I don't have any advice for a name for you. But, I'll give you one piece of advice, don't associate your name with it because, he says, if you've got an idea that's not in the mainstream, you'll become the target and people will be able to discredit the idea based on talking about you and say you don't have any credentials, this can't possibly be a good idea. So he said, it's got to be neutral. Don't have your name associated with it. Well, that was good advice; I understood that; it was my natural inclination. But I couldn't come up with a name. Goodenough was a mathematician and Ph.D. out of Harvard, so I continued to talk to him. He was probably still my boss, I said I gotta have a name. He understood, being a researcher, he understood it had to have a name. So he says, well you know, what you describe sort of sounds like what in mathematics what we think of as a kernel. And so he says, maybe that's the idea there; some notion of that. And so we kicked around ideas and it's about security, let's call it a "Security Kernel." So, yes, he was the one who created the name of Security Kernel so that I'd have something to write down. So, although he wasn't that involved in the substance, he understood enough to say. So that's where the name came from. The position paper I did for Dr. Gaines was called a "Security Kernel," and that was the first formulation.

The panel—the Anderson Panel—continued in their deliberations along the way. I put out that sort of notion; and, you got the senior guys thinking he doesn't know what he's

talking about; I got the polite listening-to, but not much more. But Anderson began to, I think, understand. And as a hypothesis, there was no meat on the bone to say how is this going to work, although I could generally describe it. I said well, I think the key thing is I've got the policy. We all know what the multi-level policy is I want to do; and we know we're going to be dealing with some sort of objects of information. In Multics, that notion of objects could be articulated in the notion of a segment, and so I have something concrete I can talk about. And access to it is enforced at a lower level—at the hardware—which is where I'm going to have this small kernel, and so I'm going to think of the segments as sort of my objects. And there are things that are trying to do processing; when people talked about [them] at the panel they talked about tasks, and processes, and things like that. And those are the sort of things that; this entity that, you know, might try and access information. And Butler Lampson had done a paper on the access matrix, and he talked about; I think he'd used the terms subjects and objects in his paper. And I said well, those seem like good terms to abstractly talk about this. And so if the relationship between subjects and objects were more formalized and structured, I could have something that actually is the thing that sits in the middle of access between subjects and objects, and that's what Butler Lampson had talked about in his access matrix paper. And so I said well, okay, we'll call that the reference monitor. And then, I said well, okay, the security kernel is sort of the implementation of this abstraction of the reference monitor. And, I said, well what are the properties of the security kernel? One of the panel members asked. I suppose one of the academic ones; maybe Glaser or somebody. I hadn't thought about that. And so well, okay, let's think about it. What does it have to be. Well, it has to be always invoked, i.e., non-bypassable, so that I can't get at the objects except through

the reference monitor. And it has to be small and simple enough that I can verify what it does; I have to be able to validate that it is complete. In other words, it has to review my whole policy; it has to have both positives and negatives of the policy; what I can do and what I can't do. And to be complete it has to be tamperproof. And so, we sort of kicked those ideas around and it was getting to the point where we had to actually do a first draft of the report. So I went down to Philadelphia where Anderson lived, and we said okay, we're going to spend two or three days talking about the report. Being project manager, I was always goal-oriented; I gotta get a report out. And Anderson was more, you know, we're having these good discussions; I've got lots of notes. And as a project manager; that didn't cut it; we've gotta have a draft. I want a piece of paper; I want a draft by this date. And so spent half a day or so discussing this. And although it had been kicked around in the panel, it had not been a consensus at all. And so, at the end of that discussion, Anderson said I think that's a valuable notion, a reference monitor. And I understand your three principles of what its properties have to be; the kernel is the implementation of that. He said, I'm going to write that up for the next draft and just throw that out to the panel. And I was a little squeamish because we hadn't had a real discussion; and he said they will discuss it. As program manager, I was just more structured than he was. (Laughs.) Just let it flow, and so we did. And so he wrote the first draft of what became Volume 1 of the Anderson Report; wrote down the reference monitor ideas we discussed. He did a good job in general in writing down; capturing our ideas and putting them together. So they were built on Butler Lampson's notion of subjects and objects with some mediation, and the notion that they were policy-aware, which I think was different than what Butler had envisioned.

Yost: So your experience with Multics was an influence, as well as Lampson's ideas?

Schell: Yes. It probably was only the Multics experience that allowed me to believe that I could actually build such a thing. What I had understood about computers before, there was never a handle. In other words, if I had looked at Lampson's thing about objects, well, what's an object? You can't get your hands on it; you can't make it concrete. And if I'm going to have some small piece of the thing that can't be bypassed, it gotta be pretty close to the hardware; somehow, it's got to be there. And so the notion of segmentation, out of Multics, was what I put together with Lampson's notion of his access matrix, and I said yes, the objects are segments and the subjects are processes. And since Multics had a strong typing for both processes and segments, it allowed me to say now I believe I can think of the Security Kernel that totally controls all the processes and segments in the system. And the thing about Multics that was different than my previous understanding about computers is that information only existed as segments. There was no notion of a file; there was no notion of a disk drive; there was no notion of any of that. There was only a segment; and a segment happened to provide an instantiation for all these other things; but that's all there was for an object. And that was an understanding that was probably, I think, one of the more fundamental things that came out of Multics; was to recognize that the notion of a file is not an atomic notion. That it basically reflected what I was well familiar with, of a tape drive. I mean, you look at Linux, and you've got seek and tape drive verbs. Well, seek has no meaning for an object; it has no such mechanism. So I came to understand that the notion of a segment was the object abstraction that could

be concretely realized in hardware; that there was nothing else. You didn't have to worry about writing it out to a file; there was no such; a segment was a segment and it just got stored. And the 1965 papers talked about that; one of the papers is on segmentation and a file system. And so I came to that understanding, which gave me the handle. I could never have figured out how to have a solution to the problem without that Multics experience. If I'd not seen Multics, it would've remained a hopeless activity.

Yost: And was the influence of Lampson just a paper, or did you also communicate with him about this?

Schell: No, just the paper at that point. I don't know if I'd met him; I doubt that I'd met him at that time.

Yost: Before the first draft of the report that was written, you had the IDA workshop?

Schell: Yes, well they were sort of in that time frame, and so the back (pause)

Yost: Can you speak more about the reception to your ideas at that workshop?

Schell: Oh. (Laughs.) It didn't get very good of a reception. I think Stockton himself considered it a bit of a pipe dream. That, well yes, wouldn't we all like to have this little mechanism that would completely enforce the policy and it didn't matter what the rest did; I mean, my statement, essentially my security kernel write-up for him was such that

even if an adversary were to prepare the programs it ran, they would not be able to compromise the policy. And it was like, oh yes, sure, wouldn't we all like that. It was sort of like, yes, wouldn't we all like perpetual motion. So I think that was the one reaction; was you'll never be able to build it so what good is it to throw out this crazy idea; it's totally impractical. I think that the other reaction was one of well yes, that's just an operating system; so what are you saying? You know, that's what an operating system does; it controls the information so what's new? So those were the two reactions that it got at the workshop.

Yost: But with an operating system, obviously, it's far too large to (pause)

Schell: Yes. Recognize that in that era that operating systems were not as humongous as they are today. So, to some point of view, some people said oh, it just looks like an operating system.

Yost: At least in the early stage, Anderson showed openness to the idea. Did others?

Schell: Yes. Anderson, since he had allowed himself to be my mentor, in some sense with that comes a willingness to listen and be open. So, he would let me discuss it and he would ask the appropriate questions. That's sort of the Socratic method is what he would use, and quite effectively. In the panel, since he was the secretary and sort of the leader of the thing, he would remain agnostic; I mean, he wouldn't defend or attack, he would just simply facilitate. And I think that yes, others listened. It always suffered from the

incompleteness in saying well okay, you claim this exists, can you tell us more about what it is? And out of that panel discussion came the notion, well—probably I think significantly—by Ted Glaser, who said the standard mathematics; the standard answer is where you have a launch of something like this. So let's see if we could write a model; if we could precisely define what it is you think this reference monitor is supposed to do. And we're not going to do that in this panel. So what came out of that panel discussion was not whether or not we necessarily think this is a good idea, we can at least describe it and dispense with it, right? If we give somebody the future job of writing a model, either you succeed or you don't. And if you've got a model with which you can show that it's complete, then that's fine; and if you can't, well you've failed. Doesn't say there isn't one; just says you didn't find one. And so if you looked at that first draft, it didn't mention a model. By the time it got out of the group, there came that notion of the first task in the road map is to create a model of the reference monitor. And they didn't come to that until the panel dragged on for a couple years, probably. And nobody had any other solutions to the problem. You had all these sort of ideas; you can look at Volume II Appendices; you see sorts of things. These people all understood the problem, they'd done their report and sort of things; and there weren't any other ideas. And so I think that the security kernel made it into the report by default. I mean, I bothered to write down a little bit for the workshop; Anderson had written up the draft report and he just says as secretary, okay, so here we have something for you to mark on; get your red pens out; so who else wants to write the—we don't have to have just one answer. We'll write down as many solution paths as we can; and at the end of the day, nobody had another approach. And so that's how it ended up in the panel report. It wasn't until fairly late in the process

that the rest of the panel accepted it, meaning they were willing to sign off on the report. But they didn't accept it because they were convinced; they accepted it because they recognized they needed to have a report and this could at least lay out a road map as to what to do, and they didn't have a better alternative.

Yost: Were there any other alternatives that were approached for a time but then scrapped?

Schell: Yes, well there was the usual kind of thing, not unlike the things you would find today if you were to look at the fundamentals. You'd have the AI alternatives, right? I mean, those had quite a bit of discussion. Dan Edwards from NSA was part of AI group at M.I.T. as a student, and so that got a lot of attention. You had the speculative; well, can't we have some way of figuring out what's classified so you don't really have to keep track of it, you know? If I give you this document and I run through it and I say oh yes, it's classified "X", right? Then that's really all I need. And so I think Weisman has a paper in Volume 2 that sort of hints at that. There were; Eldred Nelson's ideas, out of his experience with SAC and missile targeting, they did this by extensive testing; essentially, path flow testing through FORTRAN programs; and the kind of thing that people still try and talk about. You know, our role, the path, how do I test them, that sort of thing. And so a lot of discussion; is that sort of the best we know how to do, to build just reliable software? It's not policy-aware. If I know how to build software, then I can make it "pretty good" was the claim, right? You know, if I really exhaustively – for some definition of exhaustively, like path coverage – test it, I can say it's pretty good. Yes. And

those were the things that were in the panel leading up to the end, that was the discussion. And because, like I say, I was the manager, they'd let me have a say in the discussion. But the bulk of the time was spent discussing these other sorts of alternatives of one form or another.

Yost: Can you discuss Glaser? He was recognized as a senior figure in this research, what had he done to that point to distinguish himself?

Schell: Well, he was one of the architects at Burroughs for, I think, both the D825 and the other Burroughs machine that (pause) 6800? No. Burroughs had an Algol machine that essentially took object code, Algol-68, and translated it. They did it; and the notion of segmentation actually existed and, in fact, the name actually at some point showed up in the Burroughs architecture. This is not the D825, which was a traditional kind of machine. But this was a fairly nontraditional kind of Burroughs machine that was used in the banking community. So he was one of the architects for that machine, and had a good understanding of the issues; and also, from a security point of view, he understood way before I did the notion that if I had a segment as an object I could now talk about security and that's what I was protecting, was the segment. So he was able to articulate that and spell that out, and then as part of the Multics project, he was an advocate for security in the Multics project. He had moved on, then off the Multics project by the time I was there, so we didn't actually overlap on Multics, but fairly shortly thereafter. And so his efforts at M.I.T., I think, had given him some knowledge. And then he had floated in the same circles that Anderson had floated in, in the intelligence community, internal

security, during penetration testing and stuff like that, that generally didn't get published. But people that knew about it, knew about it.

Yost: How wide was the distribution of the first Anderson Report and what was the response?

Schell: Well, it went through several draft stages, and then had its current form. The final form of the report, again reflecting my project manger's point of view; I said I don't want anybody to not know where they stand on this. And I said when the panel was organized, everybody's going to sign off on the report. Doesn't mean you agree with everything. But if you don't, you can write a minority report and we'll include whatever minority reports you have. But everyone's going to agree that this is what the panel produced. And Anderson's the secretary; and everyone agreed to that. And we'll know when we're done, I said, when the report is signed by everybody. And if you get a report, it literally has signatures in the front, which is the way I managed specification concurrences and reviews, and the configuration management process. Came to find out that doesn't fit very well with the research community, but they had to put up with me. So that's how we knew when we were done; when everyone signed the report. So we put together the final report; everybody signed off on the final report; and it was going to be distributed. There began to be questions raised about why is the Ware Report was classified and this wasn't, because this would to some seem more problematic to distribute than the Ware Report. This is actually talking about where we might go to produce solutions; shouldn't this be a classified report? I had dealt with the classification issues my whole career; and I said

there has to be some damage to the national interest by exposing this information. I can get classification for it, that's not the problem. Just tell me, is it derivative classification, because most classifications are derivative. And some people said well, yes, it's clearly derived from the Ware Report. I said balderdash, it isn't. Tell what I have to take out and I'll take it out. It clearly wasn't. They said well, okay. So then they said if there isn't derivative classification, which is the usual reason for classified; then there has to be an original classification authority. And I said my boss has original classification authority, that isn't the problem; but I've got to be able to explain to him what the damage is to the national security of disclosing this. And nobody could really articulate that. The couple of folks from NSA had the greatest concern, I think, about it being unclassified.

Yost: Based on culture?

Schell: Yes. But they couldn't explain; I mean, articulate their rationale; the view of the intelligence community at that time, and the acquisition community was fairly different. In the acquisition business, if I classified something, it's going to cost me a lot more money to protect it and take care of it and so I really only want to classify something that makes a difference. And the intelligence community tended, from my perspective—just independent, individual perspective—was that they tend to say well, what harm does it do to classify? If the enemy is even interested in this we ought to classify it so they don't even know it exists. And, of course, you have the usual opsec kind of stories, you know, put together lots of things; yes, we understand that but classification isn't built that way. And yes, we understand we could classify everything we do and they couldn't afford it.

As a project manager, that's how I thought; and so we really came from different worlds in that regard. So I asked, give me the reason I give my classification authority, and they couldn't. And so when it was done; when the last report was reviewed, I just had the uncomfortable feeling that the NSA guys were really unhappy about it being unclassified. So I said I don't want to have second thoughts about this or redo this afterward orhold up my delivery; I'm done. And nobody has been able to articulate any damage at all, and so, I said, go print 300 copies, or 500 copies, or something, because people I had talked to had expressed interest; and send them out to everybody. And so we did. And I thought, when we get it done; I want it out next week. Distributed. It got sent out—300, 500 copies, I don't know—lots of people cleared, and uncleared, and unclearable, and it was sent. And like the day after, I got a call; NSA has determined that this is classified and they are exercising their classification authority. I said, good luck, it's mailed; it will be arriving at all these places.

Yost: On a project where an Air Force officer is the project manager, there's a contractor, where does authority [with classification] lie?

Schell: Well the classification, original classification authority, is a fairly limited set of people. Obviously NSA has those. And they could make a classification determination and everything else is derivative. So by and large, if an original classification authority determines that that's classified, it's classified. I mean, there was nobody that was going to dispute that.

Yost: But it was distributed before that happened?

Schell: It was distributed before I knew that that was their determination. And so I said, what do you want me to do? You want me to declare it classified after the fact? That's pretty ineffective. But, you know, we'll do whatever it is you want to do. They could point to some kind of reasons why it should be classified; the fact that it named NSA participants on it was; this wasn't the original thing. Originally they said it was classified "because", but naming NSA participants got offered as kind of in the excuse category. But at the end of the day, as project officer, it wasn't my decision; I didn't have classification authority; it wasn't mine to say; it was going to be whatever I was told.

Yost: I know that the Defense Science Board, the Ware Report, that was classified. Yet in the report itself there's an argument on the need for openness for computer security research to advance research and for the participation of industry too. Was that at all an influence on your thinking?

Schell: Oh, yes, I thought about that.

Yost: Willis probably wanted that on the Ware report unclassified.

Schell: I was always driven as an acquisition guy and as I said, if this is something we're going to buy, I'm not going to be able to get the industry involved in this significantly if we classify this. Things like doing the mathematical models, where's that going to be?

That's likely to be at a university and yes, I understand universities can do classified research; it's not necessarily the most effective way to do it.

Yost: Toward the end of our morning session, you'd mentioned how the idea of developing a model for the reference monitor came about. At that point had David Bell become involved?

Schell: No, not at all. So the Anderson panel had the job of laying out a road map for the Air Force and said here's what we can do. There was clearly not a program of record, because this had grown out of this requirement for the Pentagon, which was one-off, you know, how do you solve this problem. And the answer is the Ware Commission says you can't easily; we don't know how, I guess. And so (pause)

Yost: And so was the Pentagon holding off on the operating multilevel systems?

Schell: No, they essentially gave it up, at that point. They said well, okay; let us know what we can do but we're sort of stuck. And so they just air gapped with separate solutions to the different parts of their problem; you know, with Secret on one place, Top Secret on the other; transporting tapes between.

Yost: Physically different computers?

Schell: Physically different computers; physically moving tapes one way, and the tape never comes back; that sort of thing. And that's how they had operated, and they'd hoped for better. We essentially said we can't give you a timeline for solving that problem. And the penetration test served to give them a sense for why it wouldn't be wise to do that. We could show that essentially, any user with any access whatsoever could access anything in a machine if you succeeded in penetrating it. And certainly, using commercial machines, it is trivial to subvert it in a way that adversaries would; you'd be a fool to think that something as widely used as commercial operating systems did not have in it subversion by, when you have major nation state adversaries. It's just an unreasonable assumption. Of course, outside the computer industry, and other things that are well known to that community that, as a method of operation, is well known. Why would they not; I mean, you look at the American Embassy in Moscow, right? I mean, take your pick. That happens to be one that has hit the press. But, that's just the way things are done so why would you assume that they would do hands-off of computers. Yes, it doesn't make sense. So that's why the penetration became important to demonstrate to the people that say well, would it matter if they, you know? Well, only if the information matters to you. If all you want is to be able to have access to everything. Well, no; okay. So, that was just sort of laying there. It was still a requirement; and we actually got back to it later. But at that point it was essentially, that's the best we can do.

Yost: How did the penetration studies come about?

Schell: Well, which ones? I mean, there were (pause)

Yost: The first ones on the Honeywell Multics.

Schell: Okay. Well, Multics was actually fairly late in the penetration. So you had the Pentagon running the GE 635, which was the same that was used by the GE network. And since they were offering a sort of a service, I think there was at least some bias among some of the people that said well, this is good for a commercial service; that ought to be good for us. But Jim Anderson demonstrated it just wasn't at all resistant to any professional attack. And then, I mentioned the penetration that was done in the intelligence community, it was one of their computers. And that again, demonstrated that not only could you access what was on that computer, but you could access things that were on the network; and one of the things that it demonstrated, which was not an intended demonstration was that they had told themselves and others, there were certain computers there was absolutely no way to get to; you know, computer X, Y, Z elsewhere on the network. And as a penetration team, we demonstrated that that wasn't true; that we could get to essentially, things that they didn't think anybody could get to; which didn't endear us, particularly. It's also a case where we demonstrated the use of subversion to them. To the small community that was involved in that; General Electric had system calls using their Master Mode Entry (MME) which was the hardware instruction. The penetration team installed something that Jim Anderson used to refer to as the "Roger Schell Memorial MME" because I put in a trap door, which, if you invoked the system call appropriately, you had access to everything that was there. And that was part of the report of that penetration. The World Wide Military Command and Control System

(WWMCCS) was coming about in that era. They selected the Honeywell 635 as their platform, and the follow-on 6000, just a hardware refresh of the same architecture. And they had security requirements from the outset and it was a case of reducing expectations throughout the life of that program. They started out with a strong need, and it really was a need to be effective; the program needed to be multi-level. The choices they made, made it so it wasn't going to be multi-level. They believed vender claims and that sort of thing, and so we had penetration exercises that demonstrated to them that their best thing was that if they weren't going to fix the problem was to reduce their expectations; which they did. So that was done essentially for the Joint Chiefs of Staff.

Yost: This was 1973? 1974?

Schell: Yes, mid-'70s; thereabouts. And then the Anderson Report was issued and said okay, there may be hope. And as I reflected, I think, most of the panel gave up. I don't think they strongly believed that there was hope, most of them, but they couldn't think of anything better. We proceeded to explore the road map that had been laid out, and the first step in the road map was a mathematical model. The thing that was understood then, and I think more recently has been largely forgotten, was to be useful, the model has to actually represent the policy—what I'm trying to do, and not just represent whatever the mechanism is. In my perspective, almost all security models today are tautologies. They start out by saying well, here's what I do; here's my Unix and Sparc mechanism. Now I write a mathematical expression that tells what that is. And it doesn't tell me anything except maybe I accurately wrote down an abstraction of what was there; but it doesn't

have any way to state anything about security properties, really, in a policy sense. And what the Anderson panel clearly spelled out is you have to clearly state, "my policy is this need to know and multi- level security policies." And now the question is, that's what I want to model, and I only succeed if I'm able to do that. And the hypothesis of the security kernel—as yet unproven—was that I can, in fact, implement that model in a verifiable way. So the first step was to do the model, and realizing that most people didn't believe that such a model was possible. What's possible, meaning the model is going to have a theorem at the end that says, for all possible programs that might run in this thing (the security kernel), information cannot flow from say, Top Secret to Secret. And to say it "for all possible programs" is an extremely strong statement because it says you've protected against attacks that you, as a designer, never even thought of. So that's why I say the Stockton Gains reaction to this sort of thing is sort of like the reaction to perpetual motion; it's like well sure, we'd like to have that but we don't think you can do that. We don't think such a thing exists. So what I chose to do; recognizing that was one of the greatest uncertainties; was sponsor two parallel activities. One at the MITRE Corporation—where David E. Bell and Len LaPadula were at; and Steve Lipner. The other one was at Case Western Reserve University, where Ted Glaser was department chairman and had a professor, Ken Walters. And I gave them both that problem. And if you read David Bell's paper a few years ago on the Bell/LaPadula model revisited, you know, you don't have to read very much between the lines that the people working on it at the time felt that they had some guidance from their project officer. I think the people doing the modeling really didn't want to be so tightly constrained; to answer the only question, which to me was the only important question; which was can I satisfy the

policy? Don't tell me about all the other wonderful things you can do; I'm not interested in modeling for modeling's sake. I only want to know whether or not you can say that the policy's preserved for all possible programs. And so there was, you know, some back and forth; particularly on the MITRE side, since they were the system engineering experts for the Air Force. They felt, I think, more autonomy in terms of doing things their way, their direction. On the other hand, the Case Western people, which to them I was just a sponsoring agency for academic research and they were probably more interested in satisfying their sponsor so they were interested in following direction, so to speak. And so those two efforts were launched in parallel. At the same time, as I say, the Pentagon gave up on what they could do with the 635s but Honeywell was making a fairly major press on Multics. And they had a full page ad in the *Wall Street Journal* that said a system so secure—you're talking now mid- to early '70s, full page ad in the *Wall Street Journal*—it says a system so secure you can trust your most sensitive corporate data to it. And so that was their press. And it turned out that in the course of our explorations of people on the industry side, because my acquisition experience always said I wanted to buy rather than build, or whenever I could; that we discovered that both General Motors and Ford had major what you might call multi-level security problems. That keeping their engineering data; was their intellectual property that was highly important; and they wanted to keep it separate both from their competition, you know, each other; as well as elsewhere in the world. And so they were interested in Multics, as well. And so we were encouraged to answer the question, okay, so why hasn't Honeywell solved the problem? I mean, it's clearly claimed to be solved. And so that lead to the Multics penetration exercises, as we did an assessment. Nominally, the assessment was for the DoD, but we

had significant interest from General Motors and Ford—more General Motors than Ford, as it turned out—in terms of what we would find.

Yost: Was the financial sponsorship of it solely from the DoD?

Schell: Solely from the DOD, yes, that's right. The financial sponsorship was always an interesting challenge because nobody had decided this was an important topic to be worked on. The Office of the Secretary of Defense has an organization at that time called DDR&E, which was responsible for research and engineering. And the director of DDR&E came from the electronics industry and his primary advisor in terms of computers came from the nuclear research community and was very open about saying time sharing is a total waste of resources; I mean, you shouldn't waste a computer on serving individuals, and even multi-program use is probably a waste of computers. And so, over a period of several years, that went, after the Anderson Report, began with the DDR&E folks, which had to review the budget, saying well, this is clearly a pipe dream. They'd listen to the people like those that'd come out of the workshop in Princeton; says can't be done; can't possibly be achieved. And over several years they went from can't be done; to we don't believe you; to we haven't seen it running, to well you couldn't really make it work in a real world; to, well, even if—their cry at the end of the saga was "even if successful, would not be useful to have a high assurance multi-level system," which meant, because you shouldn't be running timesharing; you shouldn't be wasting computers that way anyway. So DDR&E was a constant to annually whack-the-budget, even through Air Force Systems Command and other commands said we need this sort of

thing. So getting sponsorship was very much a matter of patching together a little bit here and there from somebody that cared. From ESD programs of record; occasional other things; with Pentagon people sponsoring some things. So these were all relatively small projects. We had two relatively small projects; the MITRE people, there was a discretionary project called "The Line," which just meant that as a FCRC, Federal Contract Research Center, they sponsored ESD out of a very small amount of money that was essentially discretionary; they could have them work on whatever they thought was important to their organization. So Steve Lipner was responsible for the piece that we used to develop the Bell/LaPadula model kind of thing. There was no external sponsorship except that we took it out of the budget for FCRCs, for that purpose. So that's how they did it. For Case Western, since they weren't FCRC, we had to patch together some small amount of money for them.

Yost: Do you recall where that came from?

Schell: I don't remember where their particular pot came from. As a program manager, I sort of didn't care very much. I mean, I had to keep the accountants happy in terms of traceability; but the reality is I said we had a job to do and we find what funding we can to move that along. No, I don't remember which particular color of money might have gone to there. So we did the Multics penetration, which was done by in-house resources. Paul Karger worked for me and a couple of other Air Force lieutenants, and we did the Multics penetration in-house with a little help from MITRE using their line funding in

order to answer this question. They said Multics was the answer, and that of course resulted in the Multics Vulnerability Report.

Yost: Was Honeywell on board with this? Did they want to be evaluated in this way?

Schell: There was an evolution that occurred. They had this commercial product, which they said was secure. And initially, when we said well, we want to do this penetration; the answer was well, have a good day; you're not going to succeed, right? We have a secure system; that's all we can do. They were on board in the sense that they said well, fine, we're aware that you're doing that. They're development activity from Multics was actually co-hosted, so to speak, with Project MAC. Project MAC benefitted by having essentially development team resources available for them. So there's significant benefit to Project MAC, and Honeywell got the advice and consultation of students working on things they were interested in; so it was a good symbiotic relationship but it meant that their resources were there at the university. Well, one of the things that the IT department at M.I.T. would do is they would sell others—if you had a research effort with them—they would sell computer time; you know, sell excess computer time. And so we got an account on the M.I.T. machine as just an ordinary paying user; not quite ordinary because you had to have some sufficient relationship that they considered a business relationship. But basically, an ordinary user just signed up for an account to the Multics. And that's what we used as a primary source of our penetration. It turned out that Rome Air Development Center in Rome, New York also had a Multics machine, and so we actually would do our first experiments on that government-owned machine. And then having

perfected it, would move it to the current activity, because of the two, the one at M.I.T. was always more current and up-to-date than the one that was in the field, the deployed one. And maybe they'd fixed the problems. So that was sort of the penetration test to answer the question, well, isn't it already secure enough? Now, we understood that certainly it wasn't secure enough with respect to any determined adversary, because certainly they'd done nothing about subversion. Well, you could argue that it was better than most; which it was. We'd looked at—by that time—IBM, and GE, and various other kind of computers. There was no doubt, in some subjective sense, that it was dramatically better than any of the extant operating systems of the day; but still wasn't going to be secure from anybody that cared; any professional attacker using subversion. And so that resulted in the report that you saw a volume of it, perhaps, which is the one Paul Karger primarily wrote, with some help from Steve Lipner. That just concluded, not secure. That got picked up and so when you ask about Honeywell on board, well, that Air Force report got picked up by a reporter at *Fortune* magazine and there was an article in *Fortune* magazine that talked about the penetrations of Multics; and there was (pause)

Yost: *The New Yorker* article?

Schell: *The New Yorker*. So that, of course, meant that Honeywell wasn't on board anymore. (Laughs.) This was not what they considered in their best interest. So the original attacks were done against the GE 645, which was the preproduction version of Multics, and they were moving to the 6080, which was to be the proven end product. So the answer to everything was well, that's fixed in the follow-on version. One of our

objectives; my objective as a program manager, was I wanted Honeywell to help move this along because to go to the next step. We required their participation if we were going to have this affect anything that the Pentagon might do because the Pentagon people had already left them hanging—which, we're looking at it—and particularly since it was Honeywell in both cases, for their existing machine and the follow-on, they really wanted that to be the answer. And so we were looking at that for them. Well the obvious thing that it failed, just in terms of features, was that it didn't support the multi-level policy. You had no way to establish the fact that this is Secret and this information is unclassified. Just didn't have those capabilities, but they weren't interested in adding those because they said nobody ever wanted to do that; in spite of the fact that we had both Ford and General Motors telling them they did. That didn't matter to them; their engineers had decided that wasn't what made sense. So, in moving in that direction, we said well, if we can take this model that the Anderson Report had talked about, we should be able to reflect that even in Multics, even though it isn't really high assurance because from the standpoint of the applications it would look the same. It wouldn't matter if it was high assurance or not. The applications would run and look the same; and to put the multi-level security policy into it. Well, they weren't on board in doing that. I talked to the senior people at Honeywell; there were expressions of what it would take; we had estimates that they consider their "careful engineering estimate" was that it would take four times as long to run any job, if they had those multi-level controls; you know, a 400 percent degradation in performance; and they were confident of that. That was just the way it was. No, they weren't interested in doing that. The only way they were interested in doing that was if we were going to fully pay them, and to boot, they weren't even

really interested then because the opportunity cost; they didn't want their engineers wasting their time on doing this stupid mandatory access control (i.e., multilevel secure) stuff. And so, no, at that point when we tried to move in a direction forward, it wasn't going anyplace. And that was the same time we were doing our penetration testing. So I decided that they probably needed a little bit of encouragement and so we made the assertion at our senior management level that their system had been penetrated. Well, since they'd made the full page ad, you understand how that plays out. You play the PR war and, of course, we weren't going public, but they were. And so finally, we said well, if we can show you that it had been penetrated, would be interested in working to improve that situation? And at some level we got some vice president to say yes, sure, right; that's good. He checked with his people, and checked it twice; and they said don't worry, you're not at risk. And so he essentially told our general, yes, if you can show us the penetration, we'll share the cost of making that change for you. So, we invited them to a demonstration; the vice president and his staff all came out to Hanscom, and we had connected their M.I.T. machine because of course it was a remote connection, modem connection; and said just thought maybe you'd want to see this; anyone here got an account? They'd brought their system administrator with them, which was good; anybody here have an account they wouldn't want us into it? So somebody says well, yes, I got an account. What's your name; you want me to find out your user ID? You want to tell? Eh, my user ID is no big deal; it's just my name, right? So here, fine. Come here and watch over here at the console. So Paul Karger sits at the console and types "get password" and print. And spits out a password. He says, is that your password? Well yeah, it is. And the system administrator says, everybody knows you don't have physical security at M.I.T.;

you've been dumpster diving and you found the passwords someplace on some printout someplace, you know, some dump and you got it. That doesn't prove anything. Big deal. Well, that's interesting; tell you what, Mr. System Administrator, you walk over here, you log into your account and change your password. And write down the password that you changed it to, and hand it to your vice president. Okay. And now Paul Karger goes "get password" and print. And then hands it to the vice president and says, is this the one you have. And of course it was. So, obviously, that was harder to make a story about. I got severely criticized because I was accused of running and orchestrating a drama rather than doing science. I said, I don't care; I'm program manager; say what you want to say. You know, there was a bit of orchestrated drama for the effect. But as a result of that, Honeywell had said yes, we'll share half the cost of adding the mandatory controls to Multics; it's going to slow it down by a factor of four, and it would take forever. We still didn't have a budget because it kept being whacked by DDR&E, and so I went to DARPA and said, you folks over here at DARPA, you're always saying you're concerned with military service adoption of your work. I've got the opportunity to give you service adoption, and I said, we think this is so important that I'll take care of getting half this project funded if you'll take care of the other half. And so the DARPA guy said well, yes, we're always getting criticized—this was around one of the periods when they were getting criticized for not being relevant-—he says yes, we'll do that. And so they (pause)

Yost: Is this out of the IPTO office or elsewhere within DARPA?

Schell: Yes, IPTO. And so they provided us, as the agent for DARPA, for half the cost; and of course, Honeywell had agreed to the other half of the cost; and I didn't have to solve my budget problem because I took that, and gave that as a contract to Honeywell to make the change. Half the contract I paid in cash, and half the contract got paid by their cost-sharing; a formal cost-sharing contract. And so now they set about to do that. They had an engineer named Jerry Whitmore, who was going to build that; and so now, what is it you want me to build? What I want you to build is what the model describes. MITRE hasn't finished our model yet; I can't give you that, but Case Western has. They're far enough along that we can interpret that. And so we wrote a document—I think it's historical significance—which are the requirements that we gave Honeywell. But the Honeywell guy wrote it to say here are the changes we want, here's the capabilities we want added to the Multics system in order to make it support the multi-level security policy. And that became the specification for the development that they did. And all the while Honeywell was saying you tricked us into this; you ran this charade; you embarrassed our vice president; yadayada; all that sort of stuff. But we were certainly not interested on this end if it's going to be bad performance and everything else. So they went ahead and they implemented the controls—and they did—to the Ken Walters model, and at the end they had to do some cleanup in order to deal with the security issues, they compared the performance of the two, their standard version and this one. Well, it turned out that the multi-level version had slightly better performance than their standard version because of the cleanups that we'd done along the way. And so they, of course, changed their tune at that point and said oh, well yes, that's what we intended all along. So it became essentially a software switch that the code base was exactly the same

code. That was the code that they shipped as their standard release, and if you bought the feature of their Access Isolation Mechanism (AIM), well they essentially turned on a switch to enable it, but they didn't change the code. Code was there always, for everybody, because it didn't have the four times performance penalty. And so that was where the penetration test fit in. It served as primarily motivation to Honeywell to provide cost-sharing for the contract, which they did; and which resulted in the mandatory controls that were in the Multics for the rest of its lifetime; and were then used by General Motors to support theirs; as well as, I understand, Ford. Less familiarity with Ford than I had with General Motors. And so that was the Ken Walters model and the Bell/LaPadula model is proceeding along, and it had sort of gone in parallel with this. And now the Multics system is already done. We said to the Pentagon, we think this is a significant improvement; still not truly high assurane, but it may meet some of your needs in a controlled environment; and they, in fact, ordered them to install and replace their primary processing power with the Multics capability in the Pentagon. So ESD was off the hook, so to speak, for having delivered—finally—on their original thing that started all this. But we now had a Multics with the multi-level controls in it, and that's what Karger's report talks about, somewhat, is what had to be done. We recognized that wasn't high assurance, so we got a follow-on DARPA sponsorship for a research project to look for what was called an auditable security in the system. And that was a project called "Project Guardian" at M.I.T. headed by Jerry Saltzer. And that went on for several years. There were several difficult engineering problems that were solved by that about how to build high-assurance version of the thing. At a demonstration level, what I had done after the Anderson Report, and as they began to move toward building the

Bell/LaPadula model, I took a Digital Equipment Corporation PDP-11/45. I picked the 11/45 because it had hardware segmentation, and the other DEC equipment didn't; and used the 11/45 to build a security kernel; a guy named Lee Schiller built a first demonstration security kernel. The first running security kernel was on the DEC PDP-11/45, which was a legitimate security kernel, and was tamperproof, small and verifiable, and non-by-passable, and had three protection rings in the hardware. They didn't know it, but they did. And so that was close enough to allow us to build that security kernel. And that served as a demonstration of the engineering feasibility of doing it. And now we'd like to put a security kernel on Multics, which, of course, it didn't have and that was Project Guardian. How can I have a high assurance security kernel underneath Multics? And the Multics security evaluation report, Karger's report, it was advocating the funding for that security kernel-based Multics. M.I.T. then did; we actually had a formal specification and fairly detailed engineering design for a security kernel for Multics, to put a security kernel underneath a Multics operating system. It did not get funded and so it did not get realized, but it was far enough along the design it was clearly possible you say yes, I can build that; we know how to build that. So with what we learned on the 11/45 with the modeling, we did that. And so the question was in the MITRE complex—because Lee Schiller was of MITRE—well, so we got this Bell/LaPadula model; not everybody was happy that the Ken Walters model was the model that everybody was using; was being sold by Honeywell; and so the question is what is the difference? Compare and contrast with the Ken Walters model. So, based on my look at it, I said I think they're equivalent, but just very, very different expressions of it; not all the same modeling techniques at all. And so I suggested to MITRE that the way to address that

image problem was to take the Bell/LaPadula model and do a Multics interpretation; to say, okay, Multics was not built with this model, let's look at how the correspondence; I did want to have the Multics to reflect how the Bell/LaPadula model does it, corresponds. And so the most widely used reference, the Bell/LaPadula model, is the Multics interpretation. If you look in the literature, almost all the references are to the Multics interpretation. That was a post facto document that was done to essentially show the relevance of the Bell/LaPadula model to what was actually deployed operationally. As I'd expected, it did in fact; the correspondence was there.

Yost: The DEC PDP-11/45 that you talked about, what level would that have corresponded to, if you use the assurance criteria in the Orange Book?

Schell: Oh, it would've been an Class A1.

Yost: It would've been an A1?

Schell: Yes, it would've been a Class A1M under the TNI (Red Book) because it only dealt significantly with the MLS; well, it actually had to; yes, it would've been a Class A1. Yes. And, of course, the auditable kernel for Multics was intended; that was designed to be Class A1, as well. But of course, nobody knew what Class A1 was, at that point; but I mean, that was (pause)

Yost: Right.

Schell: But in terms of the technology, yes, that would've been a Class A1 system.

Yost: Was there any criticism in the industry that you were working with certain companies and that if successful there would be potential commercial advantages?

Schell: Oh, yes. You always have criticism. But again, my acquisition background had just made me jaded about any kind; any time you ultimately select a contractor for some activity, you're always going to get; the losers are always going to complain. And so I'm just largely immune to that. Obviously, you do in fact have to be fair at that level, or else you get in big trouble and the corrective measures are all there. But just the rabble rousing and complaining, I, just as a project manager, I just threw that away. So yes, there were complaints about that. But the complaints were sort of, at that point, non-serious on the face of it. I mean, clearly the only thing that made this work, as I had mentioned earlier, is processor hardware segmentations supporting access to the hard drive. Well, nobody else had hardware segmentation supporting access to the hard drive. IBM for a while supported something called Future System that would have been Multics-like, and would've supported segmentation. And they didn't pursue it for whatever set of reasons.

Yost: IBM had allocated a significant budget towards computer security.

Schell: Yes.

Yost: So was IBM achieving anything with that?

Schell: (Laughs.) Well, 1974, as I recall, the ACM conference in Boston; IBM was at the height; it was just coming out that IBM had put $40 million toward that computer security project. And in 1974 dollars, that was a fair bit of investment. I was on a panel with other people at the ACM conference about computer security and somebody; a reporter from the audience asked me that question and said well, hasn't IBM solved this problem? They asked me what's your budget? I said small, hardly six figures. And he said well, IBM's putting $40 million into this, don't you think that what you're doing's pretty insignificant, or something of that sort? And I said well, not really. From what I can see, the IBM budget allocation of the $40 million is roughly $39 million for marketing and a million for travel; which some Computer World market reporter picked up on and that hit the press, and so created a little bit of consternation. I'd had my panel participation appropriately cleared by the Air Force public release people, but that wasn't in what I'd proposed to say. (Laughs.) But yes, there were always those sort of efforts to penetrate and patch. IBM showed no serious interest in high assurance security or security kernels, just none. Their interest was entirely in terms of what you would today call hardening their existing base. They didn't want to disrupt what they were doing. You may recall, or at least have heard of, after IBM went through a major trauma to go into the 360. And their management declared never again would they introduce a new operating system. And when there was any talk about a security kernel, they would just

immediately go into their corporate statement, “Never again. Go away.” So no, they weren’t a serious contender.

Yost: Can you talk about DEC’s interest a little bit more?

Schell: Yes. DEC, of course, was always a very different kind of company. They did not try and sell to the government in the same way that IBM would. They had a version called TENEX, which had DARPA tentacles in it, and they were wont to declare—this is on the PDP-10—to declare that as a secure system. And there were some penetration-related exercises that demonstrated that it was not. There were some fairly widely published attacks against the system. And it was no better or worse than anybody else who (without validation) claimed to build a high assurance system. It’s just that it’s not secure and the attacks merely demonstrated that. So they never, at that juncture, in that period of time; they did not have that serious an interest in security. They did find it interesting that we had singled out the DEC-11/45 to do the research demonstrations of the security kernel, which, you know, we defended in papers and publications as such; you know, here it is; it works; it’s basically bulletproof. What else do you want from the standpoint of concepts. But there was no interest, if you thought about what they offered on the PDP-11, it was a mini-computer and they didn’t offer the general services that would have an obvious demand for high security. And of course, when the Bell Labs people left the M.I.T. and went their way and pursued Unix, they used the PDP-11 as their base, but not the 45, in environments which exactly reflected that. They didn’t need the high security for Unix since they were just a collaborative laboratory.

Yost: If I understood you correctly, you mentioned the achievements at CWRU with Ken Walters and the AIM matched achievements (pause).

Schell: We built the AIM to the Walters model; that was the model that drove the specification. The Walters model, for the Bell/LaPadula model was an after-the-fact matching up. But the Walters model was actually the design-to model for the Multics.

Yost: You hear about CWRU and Ken Walters far less than Bell/LaPadula.

Schell: Yes.

Yost: Why is that? And is that justified?

Schell: Well, justified is a sociological judgment that I can't make. But the realities are that for me, MITRE was much easier to work with; the fact that I had the line budget that I could get them to do things, and such; and so there was a natural affinity between the Air Force and MITRE. And also, they had engineering resources; and probably more importantly, MITRE had engineering resources to go and build something like the Schiller PDP-11/45 Security Kernel, and Case Western really didn't have those resources. So as a model, no, I suspect there's no justification for one being fundamentally better than the other. But, as a program manager, it was much easier for

me to work with MITRE because they were a one-stop shopping kind of thing in that area.

Yost: I ran across a report from 1976 on Multics Security Evaluation, and came across a number of names and was wondering if you could comment on their level of participation.

Schell: Yes.

Yost: N. Adelman, J.R. Gilson, R.J. Sestek, and R. Ziller?

Schell: Okay. Adelman did significantly participate. Is this a report; do you know what the report was? Was this written by MITRE, or by the Air Force?

Yost: I believe it was by MITRE, I think; published in ‘76.

Schell: Do you know what the title was?

Yost: I think it was just Security Kernel Evaluation for Multics.

Schell: Okay, then there would’ve been a subtitle. Okay. Yes. So Adelman significantly participated; the others I don’t have recollection; I recognize the others. But I don’t have a recollection of their activities very much.

Yost: Okay. It really seems to be that a lot was being achieved in the mid-70s. And in 1976, the Air Force's computer security program essentially got shut down, is that correct?

Schell: At least by 1976, 1977. Yeah.

Yost: Who made that decision and what was the context?

Schell: Well, it was largely driven by—in terms of the actual decision—the DDR&E, because the OSD has to approve anything that goes into the budget; into the five-year plan. And as I say, every year we got from DDR&E, a whack; and the final one that essentially when they said no, that said "even if successful, would not be useful." And that was the one that proposed to do a security kernel for Multics. We'd done bits and pieces of it along the way and stringing together things. In fact, one of the objections from the OSD people is they said, every year we zero out this program and every year it doubles. That was roughly true. We found other funding for the program. It was also the case—I have no idea what the genesis of this was—but the General Accounting Office did an audit of our program because somebody had led them to believe that there was something going on financially that was not right. So they did an audit and they published an audit report; very unusual audit report that reported that indeed they were not able to determine the source of all the funding that had been given because what they determined was that the Air Force—my program—had spent more money and achieved more results

than what the government had ever paid for, and they couldn't explain how that was possible.

Yost: You found other sources for funding this for this research and development activity?

Schell: But they couldn't keep; you know, put that together. Of course, we gave them all the information. I suppose; as program manager, that was my business. Other people were researchers; I was a program manager. When I left that job they gave me a beer mug; computer mug that has engraved on it; that says "Master of the Schell Game," which was sort of the reflection of local accountants and everybody else in the bean counter area.

Schell/Yost: (Laughs)

Schell: The mission comes first, from my perspective and my job was to do this. I will do that in whatever is legal and ethical. And that didn't necessarily mean it was the way that they expected it to get done. So yes, and they were roughly right. So yes, OSD was largely the one responsible; it got executed through Systems Command. You know, if somebody in the commands below had wanted to take issue with it, they could have. But it was not big enough that somebody wanted to challenge. And so they just; okay, they don't like it, zero it out. And that's just the reality. It didn't surprise me. I understand how that worked very well, actually; (laughs) how that world works.

Yost: Do you have a rough estimate of the total of what expenditure was from the Air Force and what from other sources that was made for your computer security work that you lead between 1971 and 1976?

Schell: Yes, roughly $9 million. It's a pretty all-encompassing number that got every little bit and piece we did.

Yost: A tremendous amount of achievement for only $9 million especially compared to stated expenditures from IBM?

Schell: That's why the GAO had trouble writing their report; which was supposed to be critical of us. There were things that happened along the way; I talked about that particular Multics thread. One of the things that was going on in that same time, when we put together the plan that the Anderson Report had put together in terms of the road map, included building out this sort of thing. We were just following that road map. But if you had a Multics, you had to have a front end processor; you had to have a communications processor. So part of what we did was we had the opportunity to take some money from the Pentagon that through what some people consider creative accounting on my part, we took money that supposed to be spent on paper, and printer ribbons, and stuff like that, and used it to build the secure communications processor, this SCOMP. And the reason it's called a communications processor is it was; it was a special-built security kernel communications processor to replace the standard Honeywell communications processor

in a multi-level high assurance environment. And so I was the principle architect on that document; that was a separate contract we gave to a separate group at Honeywell. That was a Honeywell Level 6 computer. And the Honeywell Level 6 didn't have in it, segmentation, and protection rings. And one of the things that I felt was possible and we demonstrated that was indeed, was to provide essentially a hardware plug-in and add onto the hardware to add segmentation and protection rings to their standard, just ordinary, garden variety minicomputer. And part of my goal was to say to the industry, even if you; you don't have to completely tear your machines apart and start over again, you could make this as kind of an add-on to your existing, standing computer architectures and make it into one that you could support high assurance on. And so we built this SCOMP as, again, a Class A1 Security Kernel, running as it was intended, to be a front end, a processor for Multics. Because of the availability of money, we built the front end processor before we got the kernel built for Multics; and that didn't get funded. And so this SCOMP was essentially on its own as a, you know, special-built kind of kernel.

Yost: And when was this; when did you get to complete the SCOMP?

Schell: Well it was in that same 1976; 1975/1976 sort of time frame because that's when we were still trying to get the Multics one done, and we'd made good success with the SCOMP. At the same time—and that was part of the $9 million, was building that SCOMP—which was, as you know, was later spun off separately as a separate product. At the same time that the people at the Strategic Air Command (SAC) had a requirement for a system called SATIN IV, which then became SACDIN , which provided missile

controls. And I had mentioned that Dr. Nelson from TRW had been heavily involved in the missile targeting—well, his company was; I don't know his personal involvement—in that area. And one of their challenges is that as the Russians were moving to put the missiles on railroad trains going around Russia, rather than our approach, putting them in silos. They became targets; you had moving targets. And so you had to rapidly retarget Minuteman missiles. And the question was, how do you do that? Well, if you want to do it, you have to do it electronically. But if you're going to send targeting data electronically, you want to be very assured of the integrity of that targeting data because you don't want it to change from Moscow to New York, or something. So this became a major issue. And I had the opportunity to talk to some of the senior people at SAC, and had said, you know, this is a serious problem and faced with subversion, you know, just the same story I had had years before in terms of the SAGE BOMARK Nuclear Safety; same problem. You still have no protection; except now the things have moved along so you've got adversaries with a lot more people that are knowledgeable. So it wasn't much of a threat then, but now it is a serious threat, and the Russians certainly had the ability to do that. How are you going to protect that because now you have programs that are operating in these environments that are not the actual operational programs were still built by classified people, cleared people on classified machines, and etcetera. But a lot of the support programs weren't and how can you keep from contaminating? How do you preserve the integrity of what's going on? I'd given that story to General Myers when he was CINC SAC, and I said in my view, your program is at serious risk. You could do something about it, but the people in charge of that have not seen fit to do anything about it. And that, said in front of my general, so it couldn't be completely dismissed, you

know, created some bit of stir. He went back to SAC and the people there that I'd been talking to and they said no problem, security is taken care of; our contractors all tell us all this is resolved; fine; cool; no problem. And I gave the story again, and talked to some of the staff; and I called up one Friday and was talking to that office; and next Monday I called up and said I would like to talk to the Colonel's office and they said he's no longer with us, in terms of the Director of Data Automation. And so apparently, the general felt that they did have a problem; and they then surfaced their requirement that they wanted high assurance for their SACDIN Missile Control System. Well, the ESD was the acquisition agency for the computers and stuff; and were doing that; so my office, as one of my additional duties as assigned, was to provide support for that program office and we wrote the RFP that went out on the street. And the RFP essentially said you need to have a security kernel; you need to have a reference monitor; you need to have a formal security model; etcetera; and here's how it ties; and oh by the way, we're also looking at the question of integrity and you perhaps have encountered the Biba interpretation of the Bell/LaPadula model for integrity. Biba's interpretation was done specifically for the SAC program because they required integrity for missile data, and that's when my mission was to do that, for that program, specifically. And that was what we wrote into the RFP, is that they would have to meet the Biba interpretation. Well, when that RFP hit the street, the contractor community just went nuclear and the protests were all over the place for putting an unachievable nondeliverable kind of requirement on the contract. And of course, what I understand as a program manager is the difference between a need and a requirement is whether or not I can buy it. I may have lots of needs, but I don't have a requirement for anything I can't go buy. So they were telling me this isn't a

requirement because it couldn't be bought. And IBM was one of the most vociferous protesters, and one of the biggest contenders; and obviously had many millions of dollars kind of procurement. And they withdrew the procurement in the face of protest. And said well, okay; the industry says this is not a good thing. And we said well, you know, we thought you should at least hear our story. No, don't want to hear your story; don't want to hear about your demonstrations; because we had the DEC 11/45 demonstration; we said that 11/45 would do what we're asking. Everything that's required in this RFP is already running in the 11/45; all you have to do is put it on your computers. Didn't want to hear it; wouldn't look; nothing; just nope. And some of the people in the Pentagon, which were (pause)

Yost: And the industry was uniformly opposed?

Schell: The industry was uniformly against it, yes. There was nobody that wanted to sign up to bid on it.

Yost: If the system was; if a PDP-11/45 was proven, wouldn't that be as an opportunity or potential solution?

Schell: Well, of course, what you needed for this, this has to be essentially a military deployable kind of thing. So the literal DEC 11/45 is probably not a suitable computer. So you had to put it into a different computer; essentially a militarized computer. DEC didn't have one of those and so it would take work on the part of others.

Yost: Honeywell wasn't [pause]

Schell: Honeywell wasn't in that business. So, but after having done that, ESD was still the acquisitioning agency, and so I had taken to heart some of my earlier advice about management information system and put all the RFP's that we were issuing on a text editor system, so they were all electronic, which generally wasn't the case but was in our case. And so they came back and they said; you will remove every reference to security kernel out of the RFP. Yes sir. Global substitute; "computer security kernel" replaced with "access isolation mechanism" or something like that; I don't know. All the definitions, everything else remained the same; just gave a new name to the thing. They said well, we don't really trust you. Well you shouldn't. So we're going to have our guys read it carefully to be sure that you did that. See if you find any reference to security kernel in there. Nope. Not a single reference to security kernel. None. So the RFP got reissued. Still the same RFP, just gave a different name to the security kernel.

Yost: And industry responded?

Schell: Well, industry; now they've got a problem. They protested it before; they had it withdrawn. The government says I've responded to your protest. The RFP is issued. And, you know, there's a very structured dance that exists for procurements. You can't go back and ask the question that way, right? So the question is either we're going to protest again for the same thing; are you saying we didn't do our job? I mean, you know, you don't tell

the sponsor, you idiot, right? So, yes, they had no choice but to respond. Ironically, IBM, who was the main protester, won the bid.

Yost: And did they build the requisite system?

Schell: They went out and built a what would have been a Class A1SACDIN; as we knew they could. And they hired Ken Biba to lead that portion of the effort. And as of a decade or so ago, when I was down in a Minuteman missile hole, they were still running the SACDIN system. And as of three or four years ago, when I was talking to the Director of Communications at SAC, who was bemoaning the fact that he ***had*** to replace the system because they couldn't buy parts for it anymore; because you know, you're talking a long time. Trouble was, they don't have a Class A1 system they can replace it with.

Yost: What IBM system was it?

Schell: I don't even remember the designation. That's why people like DEC and Honeywell weren't contenders for that system. So that was probably going on at the same time; and again demonstrated that it didn't actually get done until it was a firm purchase requirement; I think I had departed by the time SACDIN actually got delivered.

Yost: In the 1971 through ’76 period when all this work was going on within the Air Force, what was going on within the Army and the Navy and was there cooperation between researchers in the different branches of the military?

Schell: The intelligence community had their own thing. As somebody from the NSA said this past October; they said, we’ve lived in an organization in which we have been monopolist. We have always built for and by ourself. That’s true. And so in true monopolist fashion, they had a competing program which they called BLACKER, which is essentially a VPN; a high assurance multi-level VPN. And they were sponsoring work at UCLA, and other places, to produce the high assurance kind of solutions. None of those, in my view, were particularly successful. They didn’t have the same underpinnings of the Air Force work, but they largely did not want it; they weren’t interested in taking somebody else’s work. So they were building it themselves; doing their own thing; and that was going on.

Yost: Based on adherence to a mathematical model of security?

Schell: No. It was not. It did not have the same scientific foundations, from my perspective, as what the Air Force program did. It was much more based on, you know, best engineering practice; which was certainly better than anything else that people were commonly building; just run of the mill stuff. But it wasn’t one that you could have verifiably secure. So that was the intelligence community, and they sort of; they were well aware of what we were doing. Well, we would share what we were doing; they

wouldn't share what they were doing; but they were making use hardly at all, of what we were doing. The Navy was trying to patch up things; they sponsored an effort like TENEX to do out in Hawaii; that had multi-level controls to TENEX, the way we had done to Multics, and really did not seem to concern themselves with high assurance at all. And there's sort of the pattern that I think exists, it is if you didn't have an operational incentive and hadn't seen what penetration teams do, security didn't mean the same thing to you. So that adding on the features of the multi-level controls, you know—which was what the Navy was trying to do—made it a secure machine. And then we're going to do the best engineering; and well, yes, you're always going to have holes; and you just; the same kind of attitude that you would see today about penetrate and patch; or whatever. That was largely the view and I think it's fair to say that probably the Air Force was the only place that was really saying hey, we're putting pilots lives on the line here; that's not what we mean when we say "secure". And I'm not interested when you killed my fellows just because you made this bug and you've got a buffer overflow. That's just not an acceptable answer. And that notion of dealing with a determined adversary is just a completely different point of view. All the Navy work, in terms of the TENEX; the Army didn't largely have very much. They were very much occupied with their common instruction set. They were trying to have one instruction set for all military computers and they would not entertain the notion of including segmentation in that instruction set. We said this ought to be one that could support a high assurance system. They did a vote, of course, among their computer manufacturers and very few people were voting for segmentation; they said nope, not going for it. And so that was their preoccupation; they were doing things about security for the common instruction set; wasn't really going

anyplace. And part of the reason when the Air Force terminated the program, part of the nominal excuse—I mean, it was really driven by the DDR&E comment that I reflected—but in terms of backing up that decision, one of the comments they made was this is not an Air Force problem and if OSD isn't going to support this, that means that we, the Air Force, will end up have to pay for solving the problem for the Army and the Navy, the Coast Guard, everybody else; and that doesn't make sense. And so this is a DoD problem; it should be funded by DoD, supported by DoD; and so we don't think we, the Air Force, ought to support this because it's not uniquely our problem. And I think that reflected their perception probably accurately; that the Air Force is really the only one that was significantly working on the problem of determined adversary.

Yost: In engaging in the tiger penetration in the first half of the '70s, those were done to better understand vulnerabilities defensively; but developing techniques that could be used offensively as well. Was there any kind of offensive mindset as well?

Schell: Well, without trying to identify any particular specifics, certainly if you consider that we're sitting there using classified programmers to do our development and we recognize that it would not be unexpected if an adversary were to take an offensive thing, and we didn't consider ourselves stupider than the adversary, you know, you can pretty well connect those dots.

Yost: But any discussions or communications within NSA about using penetration techniques for intelligence gathering?

Schell: I'd say that there's nothing in the historical literature that I would point you to that would provide much insight in that area.

Yost: So when the program got shut down in 1976, was that when you went to Air University?

Schell: It happened to be; yes, I was slated to go to the Air War College at the Air University. Yes, I was actually on that list before that termination occurred; I mean I already had my assignment notification when that decision was made. That was made, sort of, the summer time frame.

Yost: And what were you working on at that time?

Schell: Well, at that time, you know, before we shut down; we certainly had done the Project Guardian, which was a major effort at M.I.T.; that was wrapping up. We'd solved some of the hardest engineering problems. One of the difficult engineering problems, just for an example, if you're dealing with interprocess communication and you've two processes that want to share a set of information, and you've got a writer at one level say the low level is writing it and the higher level wants to read that information, but the low level is making multiple writes so the high level has to get a consistent picture of what it says, right? Well, the usual way you solve that kind of problem is something called P&V, in computer parlance, right? It's some sort of semaphore; some sort of synchronization

mechanism. All the synchronization mechanisms—I think ***all*** the synchronization mechanisms that existed up to the mid 1970s—were all bilateral; in other words, you had communication flow both ways. And so if you wanted to have the Top Secret read something, it would essentially put a read lock on the information and then the writing wouldn't write while it was reading, right? The trouble is that if I can put a read lock on it, then the low side can know that I have a read lock and so consequently, I can have communications by means of that lock itself. That lock just becomes a bit of information. I can make as many locks as I want, and I can communicate as high as bandwidth as I want, right? And this is not a covert channel, as covert channels are defined. It's just plain and simple I've got a piece of information; something that I can both read and write from both the high and low side. And that's just a flaw. If you look at the Bell-LaPadula model or any model, it's going to say no, you can't do that; that's not a valid interpretation. But you do need to provide synchronization. So one of the problems which M.I.T. had was how do you do that? Well, David Reed and his associate Raj Kanodia, part of their research results was to come up with a mechanism called "Event Counts," which in fact allowed me to do secure synchronization; and they published a paper in Communications of the ACM to do that. And that was, from a practical point of view for building public systems, a major step forward in engineering. Wasn't science, really. But it was engineering. I said oh, I can now do synchronization between the different levels. And any secure system that's been built, I think, since is either flawed or it uses Event Counts for synchronization. So those are the kinds of things we were working out of Project Guardian.

They did other things in terms of the problems of engineering; a layering approach, so that I could in fact do it. If I have a high assurance system, I essentially have to build the code as a proof sketch, in which I have; I create a lemma and that lemma becomes the basis for the next theorem that's on top of it, etcetera, and it builds a mathematical proof sketch. To do that, I have to have strict layering of my information. Nothing at a lower level can depend on anything at the top layer. Well, nobody had ever built an operating system that way, including THE. So people like Phil Jansen at M.I.T. developed some very powerful techniques for doing that and splitting apart some of the hard problems, like how do I do page management in an operating system in a layered fashion? They had previously all depended on the fact that I wasn't layered, that I was using higher level information. So those were all things that made it possible to engineer a high assurance system that had broader functionality. The PDP-11/45 couldn't do that because it just had a limited functionality; it was a demonstration.

But some of these hard engineering problems weren't yet solved, and so that's part of what was being worked on was wrapping up the Project Guardian, which made some major steps in terms of making it practical to build high assurance systems. The Multics system, of course, was being finished off as a commercial product, with their military controls, and mandatory controls in it. The SACDIN system; we were advising them. I was also serving as an advisor at that time for some acquisition efforts at ESD. You know, although most of what you're interested in is computer security, I probably only spent half my time doing computer security during that time. I served on acquisition reviews of programs that were in trouble from various sort of things, at the command

level. Things like that; giving advice on security to other programs; just a range of things that had nothing much to do with the things we're talking about. So those are all things I was doing when I left there.

Yost: Okay. And you wrote the Achilles Heel paper . . .

Schell: Yes.

Yost: . . . That was quite controversial.

Schell: Yes.

Yost: Can you talk about the context of that?

Schell: Well, at the Air University, you have a set of choices as to how you want to meet requirements. One of the things that you can choose to do is to prepare a research paper for the Air Force Symposium. It was an annual symposium that was held at the Air University and I had submitted to it like you would submit to any symposium; and either it gets accepted or not. And so I decided to use that; maybe one of my Air War College graduation requirements; was to write that paper. So I set about to do that. It gave me an opportunity; I would never have the time or energy to write a paper like that; but it obviously was a collection of the experiences of several years before that. And I created the paper; given the experience I'd had with the Anderson Report—somebody decided to

classify it at the last minute, after everyone had agreed it wasn't classified—I had some concerns that somebody might see this issue differently. I did all the proper reviews and such, and made sure there was only unclassified material in it. But, you know, people do find some things embarrassing and some people respond to embarrassing things by classifying them, even though we say we don't. And I recognized what I was writing there could prove embarrassing from some peoples' points of view, and so I actually prepared the paper using the Rome Air Development Center in Rome, New York, their time sharing system, on their Multics system, used their text editor to do the preparation so that it was clearly an unclassified system accessed over an open line. I used that just for mechanically preparing the paper. And then I had it reviewed. I had good fortune there at the Air War College of having people with operational experience. I had a CIA case officer and an NSA engineer that were attending there; and they gave me very good feedback in writing it there; suggestions even in terms of some of the major structure. And because of where they were from, they wouldn't let me provide an acknowledgment in the paper, but some of the techniques and such that I used for the presentation were ones that I benefitted from. So I did that; that was very valuable input they had. It came time that I had to get it to be cleared for public release since it was going to an open symposium. And it got submitted, and it was just was a normal batch of papers to the public release. And the people that were submitting it, of course, had no idea; it was just another paper. They got back this response; as far as I know, one of the few that got back NO! REJECTED. Not authorized for public release.

Okay. So it was just a flat no. Well, it turns out that the colonel at the Pentagon that had been responsible for getting me money for the SCOMP was, at that time, at Maxwell, where the Air University is located. And I collaborated with him; or at least he had connections there and I collaborated with him; and I said, you know, don't you think this is a story that needs to be told; and he very much was supportive of what we were doing; and said yes, he thought so. So he went to bat for me in terms of the public release, and said no, that's not a satisfactory answer; you can't just say no. They have to say what is it that is objectionable. And so they gave something like saying eliminate the security kernel out of the SACDIN; it was sort of well, remove all references to x, and y, and z. And so I removed all references to x, and y, and z, and the paper remained basically unchanged. But the local Air University public release people had exactly a script as to what were all the things I had to do; and I said go through and check; I said don't trust me. I warn you; don't trust me. You go through and read it and see whether or not I've done all the things there. And they went through and said yes, you've done all the things that are there. We think it's done. So they sent it back and said here, we've responded to your comments. And, at that point, the people who were disagreeing, who probably felt embarrassed by some part of it decided well, you're just stupid; you don't get the message. We said no. You don't know that no means no? It's classified. And so, I said oh, that's most unfortunate because it's all been kept for the last six months on this unclassified computer in New York and they've got backups, along with all their mission data, and everything else. And that says you'll have to just destroy all those backup tapes and everything else that are there because it's in every daily backup, and it's all mixed together, and you can't separate it, etcetera. That's just most unfortunate. I of course

immediately called the people out of New York; this resulted in some amount of turbulence; and at the end of the day, my advice to the people who were being threatened to having their machine shut down was, I said, I suggest you ask one question. What is the information in this report that is detrimental to the national interest of the United States and who is the original classification authority who's decided that; who's classified what's in there? There was never an answer; the time frame passed; and went to the University guys and said, you know, I'm going to assume that we've answered the question and if they don't come back with it; they wrote them, you know, gave them a deadline for publication. Got published. Slightly edited. There were things that were removed. Might have made it easier to follow; but you know.

Yost: It certainly came across as a warning that more needed to be done with computer security. Who did you see as the primary audience for that message?

Schell: Well, the primary audience for me was the operational Air Force; the people who were deploying systems, which are putting lives at risk by having weak security. And so that's my audience because I think that what had begun to develop then and has proceeded since then, is this sort of "good enough;" it is good enough; it doesn't really matter; this is just another engineering discipline; and yes, nobody does perfect engineering; we have to have practical security. It's that kind of rationale, without any substance to it to say what is the difference between grave damage to national interest and operational inconvenience, because you have to do something for security. And they're weighing those two on like even scales. So that was my audience; was to say to

people hey, there's a problem here and so you would deploy systems; you know, use air gap, if you don't have a better way.

Yost: What type of response did you get from the Air Force community?

Schell: Well, I think that operationally, at the time, people generally found the report readable and understandable. That people that weren't computer people—which was my audience—would understand that there was a problem here; that they at least ought to ask the question. And that was my goal was to get them to ask a question that normally they wouldn't have been asked. I mean, I thought of the colonel at SAC that kept telling the general that there's no problem; that everybody says we're doing the best we can. And the best we can is simply irrelevant when you're talking about the grave damage to the national interest. And that was my message.

Yost: Can you discuss how you went from the Air College to; was it next the Naval Postgraduate School?

Schell: The Naval Postgraduate School, right. Since I had the Ph.D., the Air Force wanted to get their money's worth out of that, so they wanted to have an assignment which required; they categorized, at that time, slots as requiring a Ph.D. or benefitting from a Ph.D. So out of the Air War College, they decided I should go to one of those kind of assignments. And the assignment that they thought I should go to—they had a lot of different candidates and such—and they had; I have no idea what they do in detail. But

they essentially; having been on the receiving end, I understand—they give you a list of candidates and you say what ones you're interested in. You know, they make a selection. At the end of the day, I was targeted at a position in DARPA, in the IPTO, to be one of their project leaders, whatever they call them in DARPA. It was probably a reasonable characterization. And it was up fairly near the final determination of that; and I went for; they wanted to have an interview, of course, at the end; and they had all my resume and everything; and so went to the interview. The person conducting the interview was a very strong advocate of Ada. And essentially DARPA had previously put forth the proposition that said well, the solution to the computer security problem is a proper language, and if I had an adequate language I can solve the computer security problem. In fact, one of the propositions of Ada of that era was that you don't need an operating system; that Ada ***is*** the operating system. And if you have Ada, you can run Ada on the bare hardware. And of course, underneath you've a run time package and whatever, but you don't need an operating system. And all these problems are operating system problems and so clearly, if you have a proper language, you don't have a security problem. You can solve the security problem with a proper language. So the interview question was essentially, don't you agree that we could design Ada to provide the solution to the computer security problem? I said no, I don't agree. It doesn't address the noncomputability problem; it doesn't address any tie to the policy; it's simply irrelevant to that question. I need a strongly typed language for implementing a security kernel, and Ada could be that language if it didn't have a huge run time package to it; but no, it is not the solution. We went around with that question for most of the interview and I basically wasn't going to say I thought—because I didn't believe—that Ada was the solution to the

problem, and I understood that if I didn't give that answer that probably wasn't going to be satisfactory. But, you know, that's the way it is. So after the interview, the Air Force got a notice that I was not technically qualified for the position and I said to the Air Force personnel guy, I said well, you want to explain this to me? I take it the problem is probably I went to not a good-enough college, right? Well he said, that could be it; let me look. Oh, M.I.T. Umm. Well, I said, maybe the problem is I didn't get enough of an advanced degree. That doesn't seem to be the problem. Well, let's see; maybe the problem is I didn't have an adequate academic record and just barely got by. No, what I see here is a straight A. Well, I said, as my personnel manager, in what way am I not technically qualified because I'd like to know so I can improve on this and be technically qualified. (Laughs.) The guy says, nah, he says, give it up. He says, they don't like you. (Laughing.)

Yost: Was Ada widely used as a possible tool to address security?

Schell: It was being widely promulgated as the thing. It was the answer to many unanswered questions. And so yes, it was the language du jour, and the solution du jour. These things come and go as quick fixes; and yes, that was Ada. It happened that I had experience before in a government-only language because JOVIAL was a government-only language and I'd experienced Ada before and I'd been involved in that before; and I'd heard that argument before I went to the War College. And I'd just drawn the parallels to the JOVIAL language and I'd just observed that to have the government be the primary sponsor of a computer language is a tough row to hoe. And there really has to

be a very disruptive advantage to doing that, and I didn't find that kind of disruptive advantage in the case of Ada. Yes, there are advantages, and yes there are nice things about it. But no, I found it was; again, from an acquisition point of view, as a program manager, it wasn't going to solve my problem because it wasn't to get adoption.

Yost: Was JOVIAL looked at, also, as having security potential at the language level?

Schell: No. JOVIAL had nothing to do with security. JOVIAL was simply a way to come out of the machine language, assembly language, into higher order language. And FORTRAN was just emerging at the time that JOVIAL was there, and FORTRAN was not very friendly for command and control kind of environments. And JOVIAL was definitely friendlier for that.

Yost: When you were at Naval Postgraduate, did you have time for research or was it largely a heavy teaching load?

Schell: I did, but it was a very different assignment. I had "work withdrawal" symptoms because I kept waiting for the "real job" because there's no crises in the university. (Laughs.) My whole military career has kind of been managing crises. There's no crisis to manage.

Yost: Or at least what they consider a crisis in the academic world really isn't.

Schell: No, I couldn't find a crisis. So I didn't quite know how to deal with that. But I figured it out. You go to work at seven o'clock and come home at five. The rest of the time you grade papers and do whatever. So I did that, but I also had interest from the CIA in some of the work I'd been doing before, in the Air Force program, and they had some research issues. At that time the CIA had a research organization, ORD. They don't currently. They actually were interested in real research and had somebody interested in operating system research. And —although I'd never been a researcher—I said oh, okay, I'm here at the university; I'll play professor. So I undertook a research program on looking at hardware implications for security, and looking at structuring a security kernel; and I thought primarily as a teaching aid and building a tutorial operating system—I taught operating systems—that I could use. And I could use, the way they worked it at the postgraduate school, I could use the CIA sponsorship money to pay somebody else to teach my classes. And so I could, in fact; I taught half time and did research half time; which a military faculty member doesn't get research time, normally, but I bought my way out to do research. And so yes, there was a whole project that was called the Secure Archival Storage System, the SASS, was the paper from that. You may have run into it. That represented the work I did at the postgraduate school for building, essentially what you could think of as a high assurance Class A1 level of network file service. And I looked at it as a multi-level archival storage system. Actually built prototypes of that in anticipation of the Intel x86 architecture. During that time I also consulted with Intel on the x86 architecture. Ted Glaser had been with them as a significant consultant during development, since he was an architect at Burroughs, and then it was naturally he would be an architect and consultant. And he had recommended that I consult with him on some

of the security issues, which I did, and had some of what I think the architect for the x86 called small but significant impacts on the x86 architecture—a reasonable characterization—so that it would support a high assurance security. And it did. I mean, the architecture; what they had originally did have flaws, and problems, and those we believe were wrung out so that the x86 architecture was one that could support that. And so the papers you saw then later with GEMSOS from Gemini Computers and such, you know, leveraged that. But the x86 was just evolving at that time and since I knew enough about what it was, and there was enough published, my research results at the postgraduate school looked forward to that. We took a z2000 microprocessor and actually laid out how we could add hardware, much as we did with the SCOMP in order to add segmentation and protection rings to a commodity microprocessor, knowing that Intel was actually going to build those into its chips.

Yost: Did you have faculty colleagues at Naval Postgraduate who were also interested in computer security?

Schell: Not in a deep way. I had supportive colleagues, who in terms of lab access and things like that; the lab manager and his staff were very supportive of what I was doing. But in terms of the other colleagues, you know, universities tend to have whatever their interests are. Everyone else, virtually, was civilian faculty that had been there for years; and planned to be there for years; were just doing their thing. No, there was not a lot of close; support, but not participation.

Yost: And did you teach at all during this period?

Schell: I taught about half time.

Yost: Any on the topic of computer security?

Schell: I taught a special graduate course, special topics on operating systems. I advertised that half the course was going to be about security and the other half about other advanced operating system topics. And then I taught the usual graduate level operating systems; you know, numbered courses.

Yost: Phil Meyers was a student of yours….? . . .

Schell: Yes.

Yost: . . . that wrote very usefully on computer security.

Schell: When I was there, I was probably one of the most productive faculty in terms of thesis students, I suppose because they were military, or whatever. But I had a good response from the students, in terms of thesis activities. Phil Meyers was one of my thesis students and so as you would appreciate, you know, in the academic community, young students come to you, they don't really know what they want to do their research on and so you sort of put them together with their skills. And so Phil Meyers was not going to be

a Nobel Laureate level mathematician, and so you found one that fit where his skills were. He had good operational background so I suggested; what I tended to do was I had a list of 20-odd topics of things that might be suitable thesis topics. I'd start out when a student came to me; I said well, there's 20 topics, any of those that interest you? We can talk about those or go make up your own; do whatever. And this was one of the topics that Phil was interested in so he did the thesis on it.

Yost: And you were at Naval Postgraduate until 1981?

Schell: Yes.

Yost: As I understand it, you—I forget what position you were going to—but that changed and then you ended up at NSA?

Schell: Yes, I was supposed to go in my acquisition career, to the Space & Missile activities in the Air Force, and I would've been Deputy Commander for the space safety for the missile shots, including the Air Force's portion of the responsibility for the space shuttle. And, you know, the last couple of people that had had that position had gotten their star out of that position. So it was viewed as an upward mobile kind of position. And that's where I was going, down in the LA area. I'd gone down and gone house hunting; and was pretty excited about going there. My wife had gone down and we were interested in going there. And given my previous experience with last minute sort of things, I called my personnel guy and I said, "just want to make sure. You've guys have

changed every assignment I've ever had. Is this for real?" He said, I've got signed notification; it's all here. I don't care of the Chief of Staff wanted to change it, he couldn't change it. Believe me. I said, so I can go buy a house? Yes, go buy your house. That's fine. And, within a week, he called me up and he said, "I lied." He said, I have on my desk a letter from the Chief of Staff of the U.S. Air Force, and it says that one Col. Schell will be assigned to the National Security Agency. And he says, I was wrong; the Chief of Staff can change it. So yes, that was very last minute.

Yost: And that was to be Deputy Director of NCSC at NSA?

Schell: That was to be Deputy Director. I knew about the forming up of the center. I'd been approached several times about various positions in the center. And actually, when I was in the Air Force, in the last six months or a year of the Air Force program, people had begun to talk about well, shouldn't we have a national or DoD Center for computer security. I reflected that the Air Force said we shouldn't be paying for this by ourselves. And so I'd been in actually a couple of meetings with various services and agencies. And there was very little that they agreed on. There was only one thing that I can recall that they strenuously agreed upon, and that is it could be anywhere except NSA. And the rationale was very understandable. It's that NSA has no urgent need for multi-level systems. Everything they did was in a closed system high environment, air gap. They simply did not understand our problem. Various representatives in those meetings would give their war stories; I said these people simply don't understand that we really need those systems. And consequently, we just couldn't depend on them to provide multi-level

solutions because they said they'll set the standard so high that nobody can meet it, and we'll never get a multi-level system, so anybody but NSA. And so when they came, three years later, and said wouldn't I like to join this center [interrupted]

Yost: How did it end up at NSA with that?

Schell: Well, it ended up at NSA. I'm told, and the person, who may not be on your list of people, but Steve Walker was in OSD at the time; and he was previously at DARPA in the area of security. And he actually had set the directions for many of the things that had gone on in security in DoD. And he had; he'd been the one to advise various people as to what was and what was not reasonable to do in terms of a center. The basic competition boiled down to there was a general recognition security providence was not getting better; we need to do something about this problem. And the viable competitors, after the politics have played out, in terms of who's got the biggest bear and a big gorilla; is it's NSA or Commerce, you know, NIST. And for various set of reasons, the NSA people—Admiral [Bobby Ray] Inman was at NSA—and he did not believe that NIST was going to do an adequate job. This was second-hand; I was not a part of any of this but from what I understand is he won the competition over NIST and he said no, it doesn't make sense to make it there, and if it's going to be in DoD it should be at NSA. And he volunteered NSA to do that job. And so that was how it ended up there, I'm told. Steve Walker could speak much more authoritatively than I can to that.

Yost: When you go there, was first task at hand was to hire a staff?

Schell: Well, I hadn't quite gotten the full story, I think, before I got there. Steve Walker had called me up after, because he was the one who gave my name to Admiral Inman. And Inman actually had come to the Naval Postgraduate School; he'd given the graduation speech as I was wrapping up here. And he'd come here to check me out; that I didn't know about; he met with my department chairman, who was, "So why does this admiral want to meet with me?" (Laughs.) The admiral doesn't want to meet with a computer science department chairman. Well, what they talked about was me, he told me afterwards. And so Walker had suggested that I go there and Admiral Inman had come and apparently he checked me out; and so that's how I ended up coming there. Walker called me up afterwards and said, well, he said, yes—'cause I called him; "how come you did this"—he says, "well, you didn't want to be a general anyway," was his comment, because he understood that was a career termination assignment. But I thought that they had formed a computer security center; that this was just sort of an ongoing thing; you know, keep it moving; keep it growing. I was told there were 35 people in the Computer Security Center and that's what I understood. When I got there, the director of the Computer Security Center had another full time job; is responsible for IT at NSA; and the Computer Security Center consisted of 35 names. He handed a list of 35 names and told me where they were. They weren't in any one place; they were all over the Washington area; and these were people that can say computer security. And they agreed that these were the Computer Security Center, and my job was to now make a Computer Security Center out of that. And he said, oh well, I'll meet once a week and we'll go over this. And obviously more than that, as needed, but he had a full time job.

Yost: And were these 35 people coming on to relocate there at NSA

Schell: Well, that was my job to figure out; with his help, of course, because he knew the agency and I didn't. Very capable manager; I learned an awful lot from him about the agency. I mean, I never could've gone in there without George Cotter, he was just tremendous, even though he left a lot of the hard work for me to do. So we got spaces, you know; we got an area; we got them together; we said this is the Computer Security Center. We located there and brought those people together, and started growing the center. And that was my job. So after a while, because he was essentially the acting director, they brought on Mel Klein as a real full time director; and that was for the rest of the time that I was there.

Yost: And from the start, was it your idea or was it understood that developing computer security evaluation criteria was to be a principle aim?

Schell: It was the aim. The official name in the DoD directory was the Computer Security Evaluation Center. That was the name. And the DoD directive that set up the center said that its goal was to encourage the easy availability of secure products. And so it was understood that it would work with industry to get secure products; encourage secure products; evaluate them; and make them available for DoD use. You know, sort of the informal characterization that my boss gave me was; he essentially said, look, we've always built all of our own hardware and software here at NSA; which was true. But, he

said, we can no longer; our systems are too complex. We cannot build from scratch everything that we need to field that needs security. We're going to have to use products that are commercial products. And if a commercial product is built out there anywhere, as far as we're concerned, it could be built by the KGB. And so, he said, since we're going to have to use those commercial products, your job is to be able to take a commercial product and evaluate it and know that I can use that commercial product to manage my most sensitive data without fear of it being compromised because that's essentially the definition of multi-level security. And having said that, my sense was that in his heart of hearts, his body language said to me that I don't really think you can; I don't think that's possible but that's your job.

Yost: And can you discuss how work went forward to create The Orange Book?

Schell: There had been parallel work that had been sponsored with MITRE because at that time I'd been out of that business for four years. MITRE had been working on an evaluation criteria; and there were lots of things. And it was sort of a Rorschach Test, in terms of evaluation criteria. And so there'd been a lot of ideas put forth and good work done by good people. Not particularly, in my view, informed by the end operational sort of goals; but, you know, that's not their job. They were technical professionals. So there was that stuff. It was not a blank slate. Steve Walker had been heavily involved so he knew what was going in on those things. So that was the starting point, in terms of the criteria. I looked at that; the people that I'd worked with in the past that were there available to me in the agency. Dan Edwards was one of the people on the Anderson

Panel, was extremely bright, very talented, capable person; and he was in the evaluation. I said okay, I want him to be in charge of that sort of evaluation stuff; he understands this; he has the operational view to understand this. So he has the job of really putting this together. I worked with him; I provided input to that; but the creation of much of the framework and such that was really his creation. My insistence was that it needed to be, at least at the highest level, a scientifically sound sort of basis. And the question of well, do you have levels? How many levels? Whatever. Well, everybody; all the political advice was look, you've got all the industry out here and the things that they are doing security. You've got to do this. You have a problem here that essentially everything you care about could only be protected; it was going to have a determined adversary. It has to deal with subversion but that isn't what anybody else was thinking when they talk about computer security, you know, IBM and their $40 million, or whatever. They're just not thinking that. And so what I suggested with Dan was you start with what we need to deal with subversion with the highest level. And then essentially chop off requirements with that, to get them to where they match some reasonable kind of things in the commercial world; and no particular misgivings that those things had significant value if you had a determined adversary. But, sort of the political trappings; you had to have that. I mean, if people were going to buy; wouldn't you rather have something that was evaluated Class C2 rather than something that was not evaluated. I suppose you could make an argument for that. But there was no misunderstanding—it wasn't protecting information you care about. It didn't matter, government or commercial. Just if you had information that that was worth a million dollars to somebody else, then none of those things that you were going to do at the lower levels were going to matter. We started out with the hope that we

would be able to have multiple high assurance levels, and so what we did early on was we said we're going to divide the criteria into divisions. And the basis for a division is unspoken; was the political context; it corresponded to threat levels. And Division A was what was dealing with the problem of a reasonably determined adversary. Division C was dealing with the pure rank amateur; you know, the bored college student; no real resources at all. And Division B was one where I lived in a protected environment so that I had a lot of classification clearance protection in a controlled environment, as it was called. But I still needed to be able to manage the different levels of classification, so I needed the multi-level controls, but I didn't rely on it to protect me against the full thrust of the determined adversary. But for Division A, I did; I assumed that they had access. And that boiled down to roughly that Division B, I assumed an adversary might produce applications that were malicious; Trojan Horses in the application. But that the operating system, I was essentially going to declare it as providing adequate controls; I do these things even though I knew it wasn't dealing with a determined adversary. It's going to say okay, that allows me to control the environment and the applications I don't trust. So it was a matter of providing protection against malicious applications. The Division A was obviously dealing with the problem of providing protection against a malicious adversary that could access you through subversion at whatever level, operating system or otherwise. So that was the structure, that guidance at about that level that I gave to Dan Edwards and we obviously discussed that. So I said Division A is, in terms of George Connor saying something I can buy from the KGB and use to control my most sensitive information, which would be crypto keys for NSA. We all agree that it's only Division A that should ever be used by NSA or anybody else that cared. And the rest, you do

whatever seems appropriate; because it says you don't have a very significant threat; this data isn't that important to you, if you decide that. And so, very much of bifurcation; that only Division A tended to deal with subversion. Anything below Division A was known to not deal with subversion, and that was the difference.

Yost: And so, in a sense, B and C were more political, fostering [interrupted]

Schell: Well, the division between the two were threat-based. As I say, Division C, you assume was best commercial practice. You didn't have a serious threat against it. Division B, you assume the threat would attack the applications because of the mandatory assess controls, wasn't able to get through the operating system in a controlled environment. And the exception; the one sort of exception thing that was done was that B1 was a transition phase. Jim Anderson called it training wheels. It was one which didn't significantly protect even against malicious applications but it provided some of the nuance. And the rationale was Jim Anderson's rationale. He says, if users get an application built, if it's used to build something where you have mandatory controls, then they can move that to a higher assurance level later on, and you have no idea what those are about in Division C. So Class B1 has virtually no assurance requirements beyond Class C2. It only has the appliqué of the appearance of mandatory controls; and that was a transition thing. It could just as well have been called Class C3, but it seemed to fit better as Class B1.

Yost: Did Jim Anderson have any formal role at the center? Did he consult for the center?

Schell: He did consult for the center; not extensively. Most of his consulting was informal. As I mentioned, he had had his long term consulting career with the agency. And, you know, when he was in town he would come and he and I and others at the center—such as Dan Edwards—spent a fair bit of time with him. But he didn't have a lot of consulting contracts with us.

Yost: Were there major efforts to try and get industry buy-in with this evaluation area, and were industry advisors brought in?

Schell: There were. You know that was a major thrust. Probably in terms of the resource efforts in the formulation; that effort to engage industry was probably the biggest effort. And we had just a lot of collaborative elements; things were sent out; publication in the federal register on down; lot of involvement. What became clear was that industry—which was not a surprise to me, but a surprise to some of the government people—that there was only industry interest where there was a market. And that the principal impression that was left, and I think we believed it at the time—turned out not to be right—was that the government cared about security. And the formation of the center was a reflection that there would be a market. And the intention was that procurements would reflect requirements; that it'd say okay, you've got to have a Class A1 for this, a Class B2 for this, as a procurement requirement. And industry saw that as something that the

government, although the government is not their dominant customer, [for] many of them it's their biggest customer; so they were interested and they figured they'd be better to cooperate and participate, and provide their input. And so we had very good participation and input from industry, frankly, mostly at the lower levels. There was virtually no significant industry contribution to Class A1. It just didn't.

Yost: See the market opportunity?

Schell: They didn't believe the market opportunity. I mean, the people who were; had the; when it came to the point of decision makers, they understood that Class A1 was dealing with the problem of determined adversary, and that was not what they believed they were dealing with. And they didn't see that; and also in terms of skills, terms of comments on the criteria, they didn't have staff that had a lot to say about Class A1.

Yost: And how was the evaluation and designation; of needing a classification to take place; and who paid for that?

Schell: So, the actual conduct of evaluation was part of the center's budget. That's how the center was formed. One of the things which was, in my point of view, that did not prevail, I suggested that the government should only sponsor the higher levels of evaluation. If I'd had my way; I had said resources are always secure, I would've said sponsor Class A1 and maybe cost share Class B3, and below that, you know, (pause)

Yost: C is less for the government.

Schell: For the government. And so, yes we will do that as a service but it's a service where you have some reimbursement method. Of course, from the bureaucrats' point of view; the U.S. government and the difference in the U.K. government is it's very difficult for the government to charge an evaluation service. I didn't have an answer to the question. How can I charge IBM for doing a Class C2 evaluation? You know, what's the nature of this instrument? How does it fit in our government procurement system? It doesn't. And so there were some parallels one could draw but they were tenuous and people said no. It ended up being all evaluations were paid for by the government.

Yost: Were you the principal author or who were the principal authors who wrote the criteria?

Schell: Well, I'd say, it really was sort of a committee effort. I was the final technical authority. In other words, they would obviously; if you get two engineers together, you get three opinions, right? There were always discussions. And so I was significantly involved in terms of addressing things and deciding no, we're going to do this because you ultimately have to have somebody decide if you've a committee that has to produce a result. That alternative is the kind of thing that NIST does do, which are sort of consensus documents. And I just said I do not believe in the center we are here to do consensus documents. We're here to do this for the benefit of the government, plain and simple, right up front, no question about it. And so it was not a consensus document, but it did

benefit from the input from industry. But I was the final authority for technical decisions; what went in, what went out, that sort of thing. And I participated in the discussions. Dan Edwards was probably technically the most knowledgeable in terms of writing it. In terms of the structuring and putting it out, Sheila Brand, who worked for Dan Edwards, is the one who, she was branch chief and it was her branch that actually produced the document. She sent them out for review; she conducted the review meetings; she did the work. She was not in terms of content, the primary technical author; but she was the author in terms of actually putting the thing together; so Sheila Brand would be the author.

Yost: Through your time as deputy director, COMSEC and COMPUSEC were still separate?

Schell: Yes, they were. There was an ongoing discussion. When the DoD directive was formed, and Steve Walker was responsible for what was going in the DoD directive; he had to sign off. And it said that NSA shall have a "separate and distinct organization" concerned with computer security, meaning separate and distinct from COMSEC and SIGINT. And the reason was that the COMSEC people clearly had a long history, a long dominance in NSA, and if you had COMSEC in charge, the computer security was not going to have the same perspective as relates to the computer. The people at COMSEC are very good at COMSEC; many of them are not good at computer security; they simply don't understand.

Yost: And they had a history of internal development rather than outside collaboration?

Schell: Absolutely. Total history of internal development; monopolistic viewpoint. And our viewpoint was exactly the opposite. And Steve Walker understood all this. He'd actually earlier in his career had worked for NSA, so he also had that experience. And so that's why the directive said it will be a separate and distinct entity. Of course, that was also part of the reason why all the services said it could be anybody except NSA; because they understood that if it went to NSA it would be dominated by COMSEC.

Yost: But within a year or two after you retired, that changed.

Schell: Yes.

Yost: From talking with people there after you retired, do you have a sense of how that impacted computer security work at the agency?

Schell: Well, from my perspective, all I have to do is look at the personnel list. When it was a separate center, we had the people that were the leading experts in the world, beyond question, in terms of computer security. I mean, there was just nobody that had any questions. I, several years later met with the Russian computer security leader; they knew which was the best security center in the world, just bar none; our organization was the NSA Computer Security Center; there's no question about it. The people that were there were international leaders, you know; from David Bell, to Marv Schaefer. Those

were the people that drove the center. After the change was made to integrate this, none of those people were there. I mean, none of those people were there. Within two years there was; you looked on their personnel list, none of those people were there. The sort of observation I made from a person from the vender's side is when I was there, I had a number of young engineers that the industry would've wanted to hire. And I had to fight with industry to keep them from hiring my engineers. If I looked at what happened, say, two or three years or four years after the integration, there was almost nobody that I would've wanted to hire in Gemini or any other vender, in a business situation. Just no, they weren't there.

Yost: Do you have a sense of the rationale, or what was the force lead to the integration?

Schell: You know, it's really speculative. If I look at correlation, which we all know is not cause; if we look at the end of the Cold War, it correlates with the end of the Cold War and a period when a lot of people didn't think we had a threat and you don't need this anymore. And so that's the thing. I think that unless you have a career path for people in an area like this, that you cannot have a sustainable organization. And I fought unsuccessfully to have a viable career path for computer security people in the agency, and that was not going to happen. The agency had adopted; they had both intelligence and COMSEC in the agency, and they had managed to have career paths for those people; they still lived in the same general career path; people could move around. That, I did not believe was going to work for computer security. And so I think there was no future in the long term for a viable organization, without doing something about the

career path. Almost nobody was going to be senior person at NSA and have a viable career in computer security without addressing that; and it was not addressed.

Yost: It was in 1984 that you retired?

Schell: Yes.

Yost: And had you thought about becoming an entrepreneur and starting a company at that time or before that?

Schell: I'd thought about it before that time because before I left the Postgraduate School, I had not really expected to get promoted to colonel. I had thought okay, I'll retire from the Postgraduate School and live in Monterey because I knew that the Postgraduate School kind of assignment was not good from a career point of view. Having gone to the Air War College, and then to the Postgraduate School; having spent four years without a real job, from the Air Force point of view, I said this is not a good sign. I said I'm probably in trouble. As it turned out, for whatever set of reasons, I did get promoted, but by that time I had already had discussions with another professor about his interest, and it was actually his idea, not mine, of some sort of an entrepreneurship venture in the area of operating systems for microcomputers. He was in the electrical engineering department. He had students that were working on microcomputer controls on things like for real time systems, surveillance, and such. And I'd helped him out with some of his students because they were doing very ad hoc executive operating systems.

And I said, it's really much simpler than that; here, look what we know about operating systems. We can in fact structure it this way. So I had helped him with some of his students to do that, and he saw that; and he saw that yes, this structure of operating systems and such, in a real time environment, was something that he didn't have a deep familiarity with was very valuable in things that he understood in terms of signal processing. And he saw that he needed multiple microcomputers and I said yes, we know how to do multiprocessing, and such. And we had begun to put that together before I knew I was going anyplace else; we'd had those discussions.

Yost: And was the goal to develop Class A1 systems?

Schell: Not so very much, at that point. It was more in the area of an operating system. Certainly I would be influenced by looking at the applications that could be highly secure, but I think the issue of a multiprocessor minicomputer, microcomputer kind of environment, was one that we said, you know, technology is going to keep; Moore's Law is going to continue at work; we're going to see multiprocessing; we're going to see parallelism; we understood those things. He said he had worked in signal processing; I was on the computer side; we said we think there's a business opportunity to do that as essentially real time controllers.

Yost: And who were some early clients or customers for you?

Schell: So that never really got to the point of being an independent thing because, I mean, we started that talk about that independent activity and then, you know, I went off. And although Gemini Computers was formed as such, and I was one of the founders, I was not actively involved in the day to day activity of the business, clearly. So in the early stages the work; not a lot of customers; there was some research kind of work, you know, fairly small businesses. So, that was an occasion where there really wasn't significant business during that; say, when I was at NSA for three years, Gemini was there; they were forming, you know, they were putting together things. They'd taken the work that I had done at the Postgraduate School on the secure archival storage system. Since that was all in the public domain and a published report, the engineers used that as a guideline that they could be informed by in doing the Gemini Multiprocessing Secure Operating Systems (GEMSOS), Gemini's flagship product**.** And that was what was done; pretty ad hoc, early on.

Yost: And when you retired from the Air Force and came on full time, what directions did the company take?

Schell: By that time, of course, I believed that security was an important part of the market opportunity. And so, at that time, we were targeting it to the Class B3 level; and that was mostly because we didn't have the resources and couldn't see the market for the Class A1 level. But particularly the formal method was a matter of resources; we didn't have anything like that, you know, available to us on staff. And so, we targeted the typical kind of real time systems, embedded systems, those kind of things were really

what we were looking at, and of course, did a lot of looking at what kind of opportunities were out there. And we were actually approached; our first biggest contract, was the NSA's BLACKER Program that approached us about serving as the primary operating system for their access controller component because they needed Class A1 and the story was that they had NSA research that was targeted for BLACKER, this was now coming to the first deliverable system. That what they were doing for their Class A1 controller wasn't adequate to meet their contract that they were billing; you know, couldn't give them performance. And if they had something that met the performance requirements, it didn't meet the security requirements. And so the report was—don't know if this is accurate—that the contractor essentially said to the government, which do you want? The one that meets the performance or the one that meets the security? Well, you know, the answer is neither of those meet the requirement. And since GEMSOS from Gemini was in NSA evaluation, the government said well, we know we have an example of one that can meet the security. Yours, Gemini. And you guys are in the business of measuring performance. And so we don't know anything about performance, you determine whether or not it meets the performance. And so UNISYS, which had the contract, approached us about this sort of contract, and when we got their proposed specification it had far more about performance than it did about security. One could say they didn't want to be embarrassed by having this startup beating them in their space. We didn't know any better; and we just went ahead and met the performance requirements. We passed all the performance requirements. The A1 requirements we passed; and so, met their needs.

Yost: In the '90s, I understand, you were a featured speaker at a trip to Russia.

Schell: Yes.

Yost: Can you tell me a little bit about that, and what you learned about their perspectives on U.S. computer security, and maybe what you learned about Russian computer security?

Schell: Yes. This was, you know, after the center was already basically disintegrating and becoming an information assurance directive of NSA.

Yost: Do you recall what year this was?

Schell: Not exactly. I'd say 1990-ish; thereabouts. And I worked for Novell at the time and Novell was in evaluation for Class C2. I was responsible for the Class C2 evaluation at Novell, along with other things; and I was in charge of their security products. And one of the things that Russia had was an evaluation regimen to determine whether or not products would be allowed to go into Russia. And so that was the basis of why I went there was to try and see whether or not Novell could get their products evaluated. And so I met with the Russian evaluation agency, and then I met with the people at the University in St. Petersburg that were providing some of their staff, as well as their computer security center. And the one thing became clear early on in the evaluation, talking to the admiral in the Kremlin about their evaluation and what they would do; their primary concern was what the translator called the "undisclosed functionality." And they

really didn't care a lot about the stuff that we in the United States would think of as security, in other words, Division B and Division C systems were just simply not of interest in terms of the issues they concerned themselves about. They were really primarily concerned with undisclosed functionality. They didn't know how to deal with that problem effectively, but they knew that that was ***the*** problem.

Yost: Trap doors, Trojan Horses.

Schell: Yes. They knew that there really wasn't any other problem that made any difference except that one. And the U.S. interest in the hackers and the things that make the press were just not of major interest to them. They considered it largely as stupid. If you put yourself in a position that allowed that to happen to you, then why were you using a computer when you; believing you could trust the one; you know, you'd be stupid not to know that. So that was the discussion in the Kremlin; and then when I went to St. Petersburg, I was asked to give a presentation. And I kept having trouble getting information about the presentation. And who exactly am I going to present it to? I'm going there corporately with Novell, and the country manager's saying, I don't know, I got a potential customer, he wants you, he asked for you by name. Just show up. (Laughs.) Do your job; get me customers. I mean, that's what a marketing manager's for, right? And so I walked into the place almost cold. And I kept getting promised tomorrow, tomorrow, tomorrow. And then on the day, I'm showing at 10 o'clock for my presentation to I don't even know who; and I don't even know what. Literally, half a hour; they had a half an hour break before my presentation; they hand me a Russian

language program, and in it is a synopsis of my speech, in the program. And I learned for the first time, from my translator, that this is essentially the annual conference of their computer security center, and that the theme of the conference is "the problem of Western software". And what I'm giving is, in fact, the final keynote; and what I learned is that although the conference had been classified, they had made unclassified the final keynote so that these university students could come. So I'm in this pretty large hall, in which university students, along with the normal participants of the conference are there; and I'm talking about; to a conference whose theme is the problem of Western software; and I was given 20 minutes. I said okay, I can wing 20 minutes anytime. And I looked at the agenda, and I was an hour and a half away; my time was an hour and a half away from the next activity, you know, lunch was at 12:00. I showed up at 10:00, and I went on at 10:30; and the lunch break was at 12:00. And I said what goes on after me? Don't you worry about it; we'll have questions and such for you. Well, of course they did. That was interesting. And following that I was invited to a sequence of activities with them; I spent the whole day there while my wife was in the hotel waiting for me to come back from my speech. I didn't get back 'til midnight with various people of the Russian security activity. They talked about the things that were of interest to them and what was going on. I went to the university; they had a department of computer security, at the time. The department chairman says, I've read all your papers. Yes, right, I thought. No, he had. I mean when someone makes that sort of claim, you know how it is, you'd say yes, right, okay, sure. Well, what did you think about what I said about (pause); well, he knew. I mean he had read, obviously quite thoroughly, all the papers that I could think of that I went through. And he was asking very informed questions; and yet, everyone I talked to

about it was clear the issue that they were addressing was undisclosed functionality; trap doors and Trojan Horses. You know, there was just little interest in the other sort of nominal controls; didn't matter to anything they cared about. 'Course, they were the government; not surprised, having come out of a controlled society. But they didn't care about stuff that didn't matter and I observed that as I would try to probe they'd pick up a Russian officer, junior officer or something; and through my translator, I would ask him about security questions. To him, computer security meant subversion. And I just observed that in contrast, if I went to a random U.S. officer and I talked to him about computer security, he would not only not mention subversion; but if I asked him what he thought about this as a problem, very few U.S. officers would've thought that this was an interesting problem to them. This is something maybe academics are concerned about but no, not a real problem. And just directly the opposite view of the Russians, as to what's important; so I found it an interesting visit.

Yost: I understand that you were involved with the Black Forest Group? Is that correct?

Schell: Yes.

Yost: Can you describe the work of that organization?

Schell: The Black Forest Group is a consortium of international Fortune 50 kind of companies. And they started couple decades ago; they're focused on IT—their principle members are CIOs or CIO designates that come—and they discuss common IT problems.

The industry understands they're common. I mean, they're selling the same IT products; which is to say we haven't got our act together because we've got petroleum, and banking, and automobile; and we never talk to each other. And we're not able to have any kind of a uniform front because the venders have divided and conquered us. And so this is a non-vender kind of consortium, whose idea is to share their perceptions of the need and requirements, and to share those with the venders. And they have a representative from each market sector, in order to keep away from the problems of anti-trust, and such. They have one from automobile, one from petroleum, one from banking, and never more than one. So they have one representative from each sector. Those are chosen in a fairly ad hoc way. But then they get together and they have people come in and give presentations, and then they give feedback to the industry. In the information flow kind of process management, they had great success by essentially forming a cabal of saying, okay, we all agree we're only going to buy products that follow this framework that we're interested in. And that worked; and they thought that was good. And they said what's our next big problem? And so I came late; since I was of the vender Novell, I was not a member of what they would call the user members, I was a technology advisor, and I was there. And they said what are the things that keep you awake at night? You're talking 1996, 1997; and they're saying the biggest problem we have that's going to do us in in IT, is malicious software. It's 1996, 1997. They said none of our people can tell us we're going to solve this problem, and the damage to our companies is absolutely devastating. And that's the biggest problem we have that's unsolved. And we don't know how we're going to get this problem solved. An instance of where that came up is electronic commerce; and they put together an electronic commerce committee to try and

look at the problem for electronic commerce. They came to this kind of obvious kind of solution, well, PKI is the kind of thing that we are going to use; let's use PKI. And they came to understand what they called the VeriSign problem. As one of the automotive representatives said, the transactions we have in a day in our electronic commerce exceed the total market worth of VeriSign. Now how in the world can VeriSign be accountable for losses we suffer as a result of what they do; there is no recourse. And Europeans called that the VeriSign problem. The PKI industry is made up by people who are essentially selling certificates and have no responsibility for accountability; there's no recourse. So they put together a framework to do that and the key to their framework was—surprise—essentially a high assurance, call it Class A1, platform, for things like the certificate authorities (CAs) to run on. They had one of the world's largest banking organizations to do a prototype of this kind of direction. They didn't have the high assurance CA, they got a commercial CA vender to go in and say we'll cooperate with you. They did a prototype; they showed how they could deal with the financial accountability problem. And no sooner did the prototype proof of concept move along, one of the other major venders in the area bought the one that they'd collaborated with and declared their contracts null and void. And that was the end of that. And at the end of that the Black Forest Group said you know, we can't do what we did in this other area because there's just too much money in the computer security business; people that don't want this problem solved. Then when they'll actually go out and buy a company because apparently there's just no other obvious reason why they bought it except to stop them from moving forward in this direction. It would've been a disadvantage to the PKI industry for sure, because it would've been a whole different set of criteria, which would

have been a business criteria of liability allocation; that you're only liable for actions you take and for things where you can provide recourse for. So yes, that was the Black Forest Group, and I continued to serve with them for a number of years; even after I left Novell I continued to serve with them. But at the end of the day, they concluded that the vested interests against high assurance just made it impractical; they were not going to fight that battle; they had other things to do; that they could not overcome the vested interests against high assurance.

Yost: Did you have any interaction with I4, that group? Do you know if, are they similar in some ways?

Schell: They're not similar. Right. And only some, you know; one or two times I had interaction. But no, they're not similar. They don't have the same rigid business criteria against IT.

Yost: Finally, are there any topics I didn't address that I didn't speak on? This has been tremendously helpful, I really appreciate all the time you've spent.

Schell: I'm glad you found it useful. I think historically, what I see is that; I was too right in the anticipating where it would be today. As one of the senior researchers at IBM labs wrote a decade ago, he said, Roger Schell predicted exactly where we would be today. I don't know if you saw that quote out of IBM labs, and he quoted one of the 1973-era papers in which I said as long as we continue to penetrate and patch, it's just going to

keep getting worse; and we trust things that are untrustworthy. And he said that a decade ago. And it's still true, unfortunately. The history has, I think, shown me that we have good solutions. If you read the papers today; you look at what's going on in Congress. Various bills that essentially want to give a free pass to people in the critical infrastructure, or anybody else who will apply the best commercial practice, that's exactly the wrong thing to do. What you're doing is you're institutionalizing the mistakes of the past. Yes, it would make sense to set a standard. Well the deal is, the only way we're going to get better infrastructure is we're going to have to require people to do something. Good. Yes, but if you said I require verifiable protection in order to be part of the critical infrastructure, that would make a difference; would make a tremendous difference. Would not be significantly more expensive; simply not significantly more expensive than what they're proposing to do; might even be cheaper. It certainly would not be a major expense. And yet it absolutely gets no consideration at all. And so I think historically people understood more; there were a smaller set of people but the percentage of people who understood the problem was much higher than it is today. Today, it is a very diminishingly small set of people who even understand the problem they're facing. They can see the consequences, and they can wring their hands about the consequences; and you see people like Admiral McConnell talking about doing more surveillance in order to have everybody share information; the answer's all surveillance. Well, we understand the first principles of non-computability, there is nothing major to be gained by massive surveillance. You can have complete full text of every message that's sent; that simply is not going to significantly impact the determined adversary, who's going to use subversion to get in and none of the things you propose to do deter me in the least, as

a member of a penetration team; as a tiger team. So why are we spending our time on things that are absolutely useless in deterring a modestly determined adversary? And so I think the unfortunate thing about history is that we've not learned from it and therefore, we're destined to repeat it.

Yost: Over lunch we talked about different factions and how they see the computer security problem. What other factions do you see besides the high assurance model that you favor, and who are the principal adherents of other perspectives, the most articulate spokespeople of such other positions?

Schell: Well, I think if you look at factions, if you look at business interests, there are really just two. There's high assurance and non-high assurance. And the ACM recently published, or had an online webcast that dealt with some of the issues, and they put it out. That security has to [be] approached fundamentally differently because it has an adversary. Almost nothing that we're currently doing has to do with dealing with it as an adversary. We deal with it as an engineering problem, or we deal with it the same way as you would deal with reliability or safety, which has nothing to do with an adversary. They assume that everybody who is in the process of creating infrastructure are equally bought into having it succeed. That isn't true when you have an adversary. And so that's why I think the security world is really bifurcated. The reality is that the money in the world, almost all the security money, is now to those vested interests. If you look at the last Bush administration had a major effort at cybersecurity. That money went entirely to two places. A major part of it went to surveillance. Second major part of it went to

research of new research. I have this constant discussion with the research people, when are you going to go out and propose all these wondrous new ideas in research? Why don't you talk about it in terms of incremental impact? Talk about what we could do with Class A1, and then talk about how much better your research is than Class A1 and what problems you'll solve that Class A1 doesn't? I find it very disquieting in the research community that researchers won't talk about that. They want to go out; they can sell a research proposal; act as if we didn't know anything more than what we knew 40 years ago; that's what they're selling. Reinventing the wheel. I had a full professor at an Ivy League college complaining because when their stuff got reviewed carefully, people were complaining well, this is something we've already; nothing new; we've already done it. Well, that's true. So instead, she said this was just vested; this was just people being, you know, in their own perspective. No, it's fact. I asked about some of her students; if they knew about some of these papers. And she said are they on the web, and I said no. And she said I can't be competitive as a professor getting students if I make my students go out and do major research on things that aren't on the web; I'm not going to do that. And she doesn't. So what if her students only know about what they can get on the web; and they're reinventing ideas poorly that have been done in the past. And so one of the vested interests; you know, rather than answer your question of enclaves, I'm talking about vested interests. One of the vested interests, I think, is the research community. One of the things that occurred with The Orange Book that people reported; people in the research community fought strongly against having The Orange Book be a standard because it dampened the interest in research; and successfully. Part of the question—this was very much a part of the question when the center was combined with the

cryptographers—and the attackers on the center included a major part of the research community, who said no, we don't want The Orange Book declared as the national standard because it'll say to people, I can do Class A1 without any major research needed; that's not a good thing. And people actually took that as a position in the DoD organization. You know, IEEE symposium in Oakland actually had a panel session which discussed that, and about how The Orange Book had discouraged things. And they claimed also—ridiculous claims and I said; they were talking about the Center's research program; I said, you know, I find it interesting you knew I was going to be at the conference. You didn't invite me to participate on your panel, yet as a former deputy director I am probably one of the most knowledgeable people in the world. Why don't you want me on your panel to talk about this? Well, the answer was clear. No, we know what you're going to say and we don't want your questions being asked. Oh, that's right. So, that's a vested interest; the financial interests. I heard people say that one of the major aerospace companies, a researcher, applying some of the things we've talked about; was told by his VP to shut up and sit down because every time a customer had a solution, had a problem, [if] it was treated as a computer services problem, they'd make hundreds of thousands or millions of dollars, for each organization. If they gave, in this case it's a question of a multi-level secure client, then I could access anything; if they delivered that, that service contract would essentially not be needed because they would've solved much of the problem. They did not want there to be on the market a high assurance multi-level client and the researcher who was interested in that actually knew some of our equipment well; was doing some research and was told essentially, sit down and shut up, because we don't want that. That's not what we're going to invest in. Another major aerospace

company, when I talked about the problem of subversion and some work they were doing for the Air Force, and said why aren't you applying high assurance? He said, after he had talked to his boss, he said, you know, I understand. I understand we can be taken, what we're doing; and we probably ***are*** being taken, but the reality is that my bosses are not interested in what you have to offer because that's not what they're getting paid for and the customer isn't insisting that we do Class A1, and therefore, they can make more money on what they're doing. So you've got that financial interest.

The other thing, if you observe David Bell's paper—which you said you read—one of the comments he made there was that he reported that the head of the Boeing Class A1 effort said that NSA put the Class A1 venders out of business. And I understand why he said that; and it's still true today. This is the turf war sort of problem. And yes indeed, NSA has had a long history of saying wait for what we are doing rather than buying vendor products; he talked about MISSI as a program that was going to be a multi-level program; going to solve one of your problems. I know at Gemini, we had 30 good candidates; and every one of those 30 good candidates said we're not interested because MISSI's coming down the line and it's going to solve the problem. And furthermore, it's being done by NSA, and they're the certifiers, and so they're going to pass it. And if we buy from you, well we don't know whether the certifier's going to; you can't give us a guaranteed pass. And so, they wait for that, but it never came. And that was followed by a similar sort of story in terms of NetTop; in terms of SE Linux; in terms of NSA's HAP; you know the line goes on. And that's vested interest, in terms of turf war; of saying this is my turf, I run this monopoly; I've always run the monopoly in the security area; I can't speak to

that personally; I'd say this is what the Boeing guy says. I understand what he's saying. And so I can't say why any of those; these are things that people have reported as reasons why. And, you know, they're plausible.

Yost: Well, thank you so much; this is great—extremely helpful.

Schell: Very welcome. I wish you well.