

# Interview

## Silver Bullet Talks with Bob Blakley

**GARY MCGRAW**  
*Cigital*

**B**ob Blakley is VP and research director of Burton Group's Identity and Privacy Strategies. Before joining Burton Group, he was chief scientist for security and privacy at IBM. Blakley is active in the security research community, having served as general chair at Oakland and also for the New Security Paradigms Conferences. He's also participated in the US National Academy of Sciences Study Group on Authentication and Privacy.

**Gary McGraw:** Bob, your degree from Michigan was the last in the computer and communications program, and I think John Holland's was the first (John's the father of genetic algorithms for those listeners who don't know that). Are we doing enough these days to teach technology professionals to think?

**Bob Blakley:** I guess. I have strong feelings about that in a broader context than just technologists. There are courses that used to be taught, for thousands of years, that taught people how to think, and they just aren't in the undergraduate curriculum anymore, so my favorite pet peeve is rhetoric.

We don't teach people rhetoric anymore—how to analyze an

argument and determine whether or not the methods used to make the argument are legitimate. There are all sorts of proposed public policy and technological ills that could conceivably be avoided, especially in a democracy, if people understood how to think about arguments.

As far as technologists are concerned, you know, I think we teach technologists more about specialized disciplines and less about general disciplines like mathematics and other fields, and also we tend to have them specialize earlier. I didn't specialize until well into my graduate career. My undergraduate degree is in classics from Princeton University, and that degree in itself came about because I got out of various other curricula under both honorable and slightly less-than-honorable circumstances. I had a very diverse background, as many of my colleagues at Michigan did, but there were 12 graduate students admitted to the program. Only one of them was an undergraduate computer science major. Today, that would be unheard of.

**McGraw:** I have a degree in the lucrative field of philosophy.

**Blakley:** Yes, I think there's much worse preparation for life than that. My classics degree focused a lot on philosophy, and in the National Academy of Sciences work

that I've been doing, one of the things that I always consciously try to do is go back and study and bring into the discussion the philosophical background of what we're talking about.

For example, identity. There's a very rich philosophical background about identity, and identity is a complex topic, so if you aren't reading what Locke said about identity and what Nietzsche said about identity and what the Buddha said about identity, you're just not paying attention.

**McGraw:** I would recommend [Robert] Nozick, too. I don't know if you've ever read his modern philosophy.

**Blakley:** Yes. And it goes right down to the modern day, right? It's not like philosophy just went out of business in the mid 19th century or anything like that. There are lots of people, like Dan Dennett, doing extremely interesting work about the function of the mind and how that feeds on identity.

**McGraw:** Switching gears, in my view, the Java 2 security model and the CORBA security model remained inscrutable to most practitioners, and that kind of rendered their uptake a little more tepid than the ideas probably warranted.

**Blakley:** The Java 2 security model

## About Bob Blakley



**B**ob Blakley is vice president and research director for Burton Group Identity and Privacy Strategies. He covers identity, privacy, security, authentication, and risk management. Prior to joining Burton Group, Blakley was former chief scientist for security and privacy at IBM and served on the US National Academy of Science's study group on Authentication Technologies and Privacy Implications. He's served as general chair of the 2003 IEEE Security and Privacy Conference and as general chair of the New Security Paradigms Workshop. He's the former editor of the OMG CORBA security specification, and authored *CORBA Security: An Introduction to Safe Computing with Objects* (Addison-Wesley, 1999). Blakley is also the editor of Open Group's Authorization API specification and currently holds more than 10 patents on security-related technologies.

has a feature that I don't like very much, which is that its approach to fine-grained authorization doesn't have as many indirections in it as it ought to. This is the JSR115 architecture.

The result is that authorizations have to be expressed very early in the process of designing and deploying an application, and if you change the policy, you have to re-deploy an object.

It is very complicated. These fine-grained models are all very complicated, and it hasn't been adopted very widely yet. Maybe the combination of [xAML] and claims-based authorization will succeed in externalizing authorization from the Java environment.

**McGraw:** I don't know. I'm a little skeptical about that. What about the CORBA idea? How does CORBA differ?

**Blakley:** The CORBA model has a feature that I like even less, which is that I—number one—was instrumental in designing it and—number two—subsequently failed to explain it in any comprehensible way to people who might adopt it. I sometimes go back and think, 'Well, I wonder whether the fact that it wasn't adopted was just a side-effect of the larger failure to adopt CORBA generally,' but I don't think so. I mean, so I designed pieces of the

ECE security model and pieces of the CORBA security model and a variety of other things, including an operating system security model for OS2, all of which had this characteristic that they were way too complicated.

In a sense, all of the things that I was famous for a long time in the security community were failures, and they were failures—it's not in a shallow sense but in a deep sense. Namely, repeated failures to learn the same lesson. Which I eventually did learn. I was also the editor of a spec at CORBA RAD—Resource, Access, Decision—which I think is still my favorite access control interface and model. It's extremely simple. After that, I became the first general editor of the SAML specification. This is an OASIS specification security markup—Security Attribute Markup Language. And that has become very successful, and it became very successful because a bunch of us—Prateek Mishra and Keith Mailer and many others—decided very early in the process that we were going to include absolutely nothing that could not be demonstrated to be essential. As a result, it was very simple. Hal Lockhart was able to draw a very elegant diagram of it that made it very clear to people, and it became successful.

**McGraw:** Do you believe that in

the quest to make security more usable, we should just focus most of our attention on simplicity?

**Blakley:** Yes, I believe that—in what may be the most radical way in the industry. I regularly tell audiences and our customers at Burton Group that no general-purpose device—and I mean this in a technical sense, a general-purpose computing system, meaning a true and complete computational device—can, in principle, ever be made secure. You laugh, but I'm perfectly serious.

**McGraw:** It's like perpetual motion. I laugh because I believe you. I believe you have one easy proposition on your hands.

**Blakley:** The proposition is very simple to state, right? By definition, a true and complete computing system has infinitely much behavior. Well, a secure computing system has a finite amount of desirable behavior and therefore, what's left over is this still infinite amount of undesirable behavior that you have to somehow prevent. And you prevent it not by the design of the system but by constraining the system after it's been designed and built.

**McGraw:** We build it so it can do some stuff, and then we get all mad when it does it.

**Blakley:** Right, we build it so that it can do everything, including everything we don't want, and then we build a set of safety interlocks that don't work, and we deploy it, and it hurts people. Not a surprise.

**McGraw:** I guess you started thinking about some of this in your famous paper "The Emperor's Old Armor" [<http://portal.acm.org/citation.cfm?id=304855>], which was published back in the IBM days?

**Blakley:** Oh, yes, that was published at my point of maximum depression. In 1994, everything was going wrong. Security was self-evidently getting much worse, which it still is. And '94 was not a good time for IBM, and I was at an Open Group meeting, and I was whining to Ellen McDermott that everything was horrible, and, in inimitable style, she said, 'Well, why don't you quit whining about it and do something?' And I thought, 'Well, that's actually good advice.'

**McGraw:** I thought I might read the manifesto from that paper because it's worth quoting. You said, 'No viable secure system design can be based on principles of policy, integrity, and secrecy because in the modern world, integrity and secrecy are not achievable and policy's not manageable.'

I suppose you were a curmudgeon before your time?

**Blakley:** Yes, very much. Well, second-generation curmudgeon, right? My dad's a cryptographer.

I think it's really true, it's demonstrably true. If you look at the systems that we have in operation, everything that we try to build to protect secrecy doesn't; and everything that we try to build that has good integrity, you know, that doesn't need a 'patch Tuesday' or any vulnerability disclosures, turns out not to have good integrity; and, in the huge majority of applications, we just use the default policy, which is inappropriate for the situation, and when we try to manage policy, we then have to buy another suite of tools that tells us what policy we have actually created, and we have to review them periodically to see whether or not they're out of sync.

You know, secure systems should be secure by default. They should be inherently secure. That is, they should be incapable of do-

ing things which are not safe, and their default configuration should be one in which they're secure. We're not close to that these days, and we won't get close to it until we begin building special-purpose devices that have, *as their only task*, the preservation of a security property and learning how to mix those things together with the general purpose elements in ways that produce security.

**McGraw:** I suppose you can draw a clear inference to software security as a likewise doomed enterprise?

**Blakley:** Well, it's hard for me to know what software security means. I know what you mean by it, but the idea that we are going to teach programmers to use a general-purpose programming language to create true and complete systems that are secure is an incorrect idea.

We're not going to do that. Now, I don't mean that the enterprise is useless. Clearly, it's better to have programmers writing good code than bad code, so we should be continuing to teach them to do that. Not only that, but maybe a general-purpose programming environment is the ideal way to tell people who don't know yet how hard security really is. Why, hey guys, work on this for a while and see how you do.

I think the exercise is noble and valuable, but a general-purpose computer system produced using the best methodology we know how to design is still not going to be secure.

**McGraw:** Switching gears to politics. President Obama recently delivered a speech about cybersecurity, I guess based on the 60-day review that Melissa Hathaway performed. Do you hold out any hope that cybersecurity initiatives in the government can move past cyber platitudes into action?

**Blakley:** Oh, absolutely. I firmly believe that in the United States, the government is us. If we wake up and demand that something be done, something will be done. We have to demand that the right thing be done, and I don't think this is rocket science. The right thing is we have to demand that people who produce computing devices that are unsafe and hurt people should be held accountable for those failures.

People are not too stupid to build a safe computing device. They just haven't been focused on it in a way that deeply affects their livelihood and well-being yet. Policymakers have absolutely a role to play in providing security, and the role that they have to play is to construct a playing field in which the incentives drive us toward, rather than away from, production of secure systems.

**McGraw:** It just seems like the market pressures of the "invisible hand" toward the impossible to attain—faster, better, cheaper—outweigh any sorts of policy wants these days.

**Blakley:** What that means is that under the current regime of incentives, we prefer to pay later rather than pay now. Well, paying later is often a lot more expensive than paying now.

**McGraw:** If you're not in office, it's way cheaper from a personal perspective.

**Blakley:** That depends, right? You might've thought that about financial regulation a while back but even the people who put in place the system of regulation that failed in the case of the banking system have now lost 80 percent of their 401K value and a whole bunch of assets elsewhere, and it's a lesson that could conceivably be learned. But psychologically, it's a

very hard lesson. Risk management studies consistently show that we always prioritize small, current gains over the possibility of large future losses.

**McGraw:** Do you think that that liability shifts that calculus?

**Blakley:** Liability regimes can shift that equation if they're designed properly. But designing them properly is subtle business.

**McGraw:** Here's a short question for you. How's privacy related to identity?

**Blakley:** Right. It would not take more than one or two seconds to talk about that. The thing about privacy [is that] people always confuse privacy with secrecy, right? Because the cryptographers got in there and started doing their mischief before we really thought about the problem from a technological point of view. I'll give just two brief examples, to illustrate

the tenuous relationship between secrecy and privacy and identity and privacy. The first example is, let's say that you know something about yourself which you would prefer to stay private. Well, until you tell somebody else, you don't have a privacy problem.

You know something about yourself, and you can just keep your mouth shut, and you're fine. Privacy rights only [come into play] when we interact socially with people who know things about us that are sensitive. Clearly, it's not about keeping secrets, it's about sharing information in a way where its sensitivity is respected by those who we share it with.

The second example that I like to give is, let's imagine that a letter is delivered to your house from a sexually transmitted disease testing clinic. The letter could, in principal, have on it your name and address and the name and return address of the facility. Well, it's going to have your name and address, or you're not going to get it, and you probably want to get it. The only personally identifiable information in the equation, namely your name and address, is going to be on the envelope. Keeping that secret or redacting it or transforming it in some way isn't possible because otherwise, you don't get the letter.

On the other hand, [you have] the return address of the business, which is not personally identifiable information, right? It's not about you, it's about the business, can be taken off, and it can be delivered in a plain brown envelope—and that does go some way to protecting privacy. If you think it's just as simple as secrecy of personally identifiable information, you get it wrong.

**McGraw:** Interesting. The last question for you—you're credited as making possible the film *Perils in Nude Modeling* in IMDb. Do tell.

**Blakley:** Making possible would be too strong a word. I provided some funding for some production. This was a student production by some people I know at the University of Texas. It's a short film that was produced as a senior project, and I know both the director and some of the staff, and also some of the people who starred in the movie. I recently—along with my two sisters and other family members, my two kids—entered the Austin 48 Hour Film Project [[www.48hourfilm.com/austin/](http://www.48hourfilm.com/austin/)].

We produced a film for that project in which we drew the horror genre and the premise of the film is that Schrödinger's cat has nine lives and comes back and does the experiment on him.

**McGraw:** Well, I hope we all get to see it someday on the Net.

**Blakley:** It will undoubtedly be infesting a YouTube near you sometime soon.

You can find additional podcasts in this series, including those featuring Matt Blaze, Kay Connelly, Bill Brenner, and Laurie Williams, at [www.computer.org/security/podcasts/](http://www.computer.org/security/podcasts/) or [www.cigital.com/silverbullet/](http://www.cigital.com/silverbullet/). □

**Gary McGraw** is Cigital's chief technology officer. His real-world experience is grounded in years of consulting with major corporations and software producers. McGraw is the author of *Exploiting Online Games* (Addison-Wesley, 2007), *Software Security: Building Security In* (Addison-Wesley, 2006), *Exploiting Software* (Addison-Wesley, 2004), *Building Secure Software* (Addison-Wesley, 2001), and five other books. McGraw has a BA in philosophy from the University of Virginia and a dual PhD in computer science and cognitive science from Indiana University. Contact him at [gem@cigital.com](mailto:gem@cigital.com).



IEEE Security & Privacy is the premier magazine for security professionals.

Each issue is packed with information about cybercrime, security and policy, privacy and legal issues, and intellectual property protection.

[www.computer.org/services/nonmem/spbnr](http://www.computer.org/services/nonmem/spbnr)

**Subscribe now!**