

# Interview

## Silver Bullet Talks with Ed Amoroso

GARY MCGRAW  
*Cigital*

**E**d Amoroso is AT&T's chief information security officer, a position he's held for almost a decade. Amoroso started working at Bell Labs right out of school and rose through the ranks before moving to the business side. His responsibilities now include security strategy, incidence response and monitoring, and customer interaction. Amoroso has written several security books, including *Cybersecurity* (Silicon Press, 2006), *The Fundamentals of Computer Security Technology* (Prentice Hall, 1994), and *Intrusion Detection* (Intrusion.Net Books, 1999).

Featured here is an excerpt adapted from a full interview between Amoroso and Silver Bullet host Gary McGraw. Their conversation ranged widely, from architecture and system design to privacy in general. You can listen to the podcast in its entirety at [www.computer.org/security/podcasts/](http://www.computer.org/security/podcasts/) or [www.cigital.com/silverbullet/](http://www.cigital.com/silverbullet/); you can also subscribe to the full series on iTunes.

**McGraw:** Software security in practice today seems to be a little bit more about bugs than flaws—that is, we over-emphasize implementation problems instead of architectural problems. Do you think too much attention is placed on things such as cross-site script-

ing bugs to the detriment of architectural progress? How does that impact your work at AT&T?

**Amoroso:** It's like the difference between wellness and good health and a guy bleeding on the street. When there are problems and bugs in software that you rely on, you have no choice, right? You have to be aware of these defects, and you have to come up with fixes. One dimension certainly is triage, and you and your gang [software security experts] are really good at that. We depend on software security experts to help keep us aware. It's tough because that can very easily devolve into a trivial pursuit game where you're just keeping track of all this crazy, arcane stuff.

My primary issue from an architecture and engineering system design perspective has always been that security should not be an overlay. When you try to overlay anything or try to retrofit something, it's never a clean fit. I like to draw an analogy to the reliability overlays in telecom in the 1980s. We'd make a big deal of the fact that we could overlay these fast rerouting algorithms onto an existing telecom network. You'd see these charts where people would say, "Oh, look, instead of making your call directly from New York to Boston, we can avoid congestion and route you from New York to Chicago to Boston," and

everybody would go, "Oooh." It looked like such an interesting approach.

Now we laugh at that. That's all embedded into the underlying infrastructure. You would never overlap reliability onto a network now; it has to be an intrinsic component. Similarly, in the '90s, we did that with quality: we hired consultants to come in and tell us how to add quality, and we would sing quality songs and do TQM [total quality management] and try to do all these things that would take the systems that we were building and the business processes that we were enforcing and designing, and then overlay some sort of a quality onto it. We have since realized how silly that is. You can't do quality separately; you have to do it as an embedded component.

We live in a decade right now when network engineers and system engineers and computer scientists and security people think absolutely nothing of designing a system, writing a piece of software, designing a network, or building an infrastructure as step A; step B is, okay; now let's do security, and do it as this add-on or overlay. That never works, particularly in software, right? I don't even know how you'd do that in a program without starting with a basic understanding of what you're trying to do.

## About Ed Amoroso



**E**d Amoroso currently serves as AT&T's chief information security officer. During his 22-year career at AT&T, he has focused exclusively on information and network security, particularly on security programs for AT&T's federal government clients and helped build the first secure Unix operating system at Bell Labs. He was the lead for trusted software security development on the Strategic Defense Initiative. Additionally, he led the initiative to address real-time security protection for the White House Y2K Information Coordination Center.

Amoroso is the author of three textbooks and several articles on information security. He has served as an adjunct professor of computer science at the Stevens Institute of Technology in New Jersey. Amoroso has an MS and PhD in computer science from the Stevens Institute of Technology, as well as a BS in physics from Dickinson College.

**McGraw:** You might become confused about it. If you think, "Hey, security must be that cryptography stuff or that authentication stuff or that password system," then you think of security as a thing instead of as a property. One pithy way of putting it is that software security isn't really security software. Do you think that message has caught on, or are people still fundamentally confused about that?

**Amoroso:** It's hard to say. You see evidence occasionally; sometimes you'll see a well-crafted piece of code and just sit back and admire it. But then you still see, even from very large corporations, these grand initiatives where the word "security" is in there somewhere, but it's this add-on, like, "We've got this initiative to make things secure." If that's necessary to get from point A to point B, then fine. Ultimately, for the same reason that you wouldn't say, "we need to do X, Y, Z to make our software dependable," you wouldn't just embed it in. I don't see the difference for security.

I've always had a hard time differentiating software correctness from software security. I know that's something you and I have had some private discussions about in the past. You really could kind of come up with different views, but to me the way software is exploited is that somebody goofs. Sometimes you're lucky and the goof is exploitable in a way that's not all that important. Other

times, they'll goof in a way where the exploitation is significant and can allow people to take control of an operating system or make an application go berserk or something. In all cases, it just strikes me that the software is too complicated. Simplifying is the most important thing.

**McGraw:** I actually agree with you on that. Maybe the difference between software quality and software security is that while you're doing software quality types of things—best practices such as code review and whatnot—if you think about your potential attacker, you might change the way you think about quality. If you know you're up against Russian organized crime versus just some stupid bug, then hopefully you're going to up your level of quality.

**Amoroso:** I guess that's true. I remember being up on my big high horse, standing on my soapbox, making a big deal about software correctness. When I was a graduate student I was a disciple of Edsger Dijkstra and just thought that everything he ever said was gospel. I still continue to maintain that much of what he says is so good.

Somebody pointed out to me that in countries (particularly in Western Europe) where Dijkstra was taken very seriously, there's certainly a weaker software industry than [there is] in the US, where we're very happy to throw code out in the wind and hope that it works.

I mean there really is, to your point, this context sensitivity—knowing your audience, knowing what you're building to. It's one thing to build something that will be some SATA controller in a power plan or something, and it's another to build a little script to share a printer in an office. Those are two fundamentally different pieces of code, and they require different attentiveness, and the vulnerability associated with each is very different.

**McGraw:** Now that we have so many home PCs on the edge, I think you've said hilariously that many people have no clue that their computer is leading a secret life of crime. That's really the only hope we have in trying to battle something like malware.

**Amoroso:** I don't know; some people think that a lighter client would make sense. I know in your book on gaming [*Exploiting Online Games*, Addison-Wesley, 2007] that you point to the fact that gaming providers are starting to rely on the end user to provide and enhance the computing environment. That's sort of an argument away from a lighter client. Take my mother as an example. She needs to get on the Internet, get her email, and occasionally open an Office application. That's pretty much it, so what the heck is she doing with a big, giant Windows machine with 99.9 percent of the software

running on it? She not only has no use for it, she doesn't even know what it is. It's sitting there in some sense for the convenience

the volume metrics across the IP infrastructure go crazy.

The trick is trying to figure out how to correlate things that you

get how many we're up to, but it's some 15 million DSL customers. If we decided, hey, we're going to police all those DSL customers' inbound/outbound, it's a little tough. If you decide that you want to, for example, cut off all UDP [User Datagram Protocol] traffic inbound/outbound, you're going to break stuff. I mean as it is, it just got so bad with spam and with relays of outbound port 25 TCP that we do filter by default. Does that cause some problems for customers? Well, if somebody's trying to run a mail server on an SMTP server through their home, then they've got to call us up and we've got to go designate an exception for them.

It's a tricky sort of thing, and that's the essence of most of the research and development going on in my own team, the security R&D that we're doing. Our dream scenario is to be able to write signatures and create these anomaly patterns that we can look for in the network, and when we see them, take action to keep things running. I mean that's the ultimate.

**McGraw:** Right, because you guys are in a great position really to monitor the cloud. There's a danger that comes with the power to look and filter—that's customer privacy and how you trade it off against security.

**Amoroso:** It's always been there. If you go back to the '70s and '80s, you had a situation where toll fraud was pretty rampant, people finding ways to use telephony service without paying. A typical case would be a little business with an 800 number, and you had all your employees call the 800 number, they get dial tone, they dial out, you get one bill. This is sort of '70s-'80s technology; you wouldn't do it that way now. In those days you did it that way. Then some nasty kid finds out about it, tells all his friends, and suddenly,

## You can't do quality separately; you have to do it as an embedded component.

of the computing industry, where it's just easier to sell everybody a PC than for her to have something that would be much more appropriate—namely, a lighter client with less software running on it, less opportunity for attack, and much easier to administer. Network providers can potentially come along and help, but there's also a very sane argument for there being considerably less software running on the desktop. You know what's cool, Gary? I think if you did a pie chart or a histogram of the origin of the software that my kids use on a daily basis—they're 14, 12, and 7—the vast majority of it does not live on their computer.

**McGraw:** On a different note, Led Zeppelin just did their first concert in 19 years, and I hear that 20 million people registered for tickets. Did you see it on the Internet?

**Amoroso:** We saw it, absolutely. For each of the different types of services that any provider offers, you've got to have the ability in the network operations center to correlate external events with what you see in the network. Like Super Bowl Sunday, boom, you see a big blip downward. *American Idol*, you see a big blip upward in TDM [time-division multiplexing] and mobile telephony, and certainly, the Internet as well. Some kooky event on the Internet where everybody wants to go in and watch—some fashion show or something that everybody's all excited about—boom, you watch

see with potential indicators of future attack because you and I both know it doesn't do you much good to notice that the infrastructure or some enterprise or your network is under attack. You already know it. A lot of times people say, "Hey, I've got this great tool that will tell you when your LAN is down," and I say, "When my LAN's down, I know it's down."

**McGraw:** I know. I keep clicking the mouse and nothing happens.

**Amoroso:** It's down, I can't get on. It doesn't make sense to observe something that's happened or is happening; you want to somehow come up with indicators that some event is about to occur or is pending or is getting started. That's the real R&D activity around this. It's complicated. You start with the basic stuff you learned when you were reading Doug Comer's book on TCP/IP. You look at that TCP/IP full association, the IP addresses of where it's coming from and where it's going to, everything. It's tricky because of botnets, particularly a botnet running fast flux, where servers in some sense are hidden by this round-robin approach to putting PCs in front of a server. God, you look at the sources, and it's almost worthless.

Destination is always very interesting because you know where an attack is aimed. As a carrier, instead of trying to go out and police all the potential sources, what you end up doing is just trying to protect customers. Let's say AT&T is a purveyor of DSL, I for-

you have 80 bazillion phone calls to Elbonia overnight.

The way that was solved is that the phone companies, credit-card companies, and anybody else doing that type of transactional analysis will watch and profile your behavior. If you haven't called Elbonia in the last six months or a year and then suddenly you call Elbonia, boom, you get popped into a database where we're going to be watching you. Now, wow, you called Elbonia again and again. Pass some threshold, a ticket is opened, an operator then calls you up and says, "Oh, Mr. McGraw, do you mean to be calling Elbonia all night?" You go, "Ahh, no, I don't." You put your clothes on, run in, go to your PBX or something, change the passwords, and then call up Sprint or Verizon or AT&T or whoever you've got, and you thank them. I would say that in 20 years, we never had anybody say, "Wait a minute, you mean you're watching when we're calling Elbonia?"

**McGraw:** You said something that was pretty funny the other day about how kids are surprised that each telephone call is actually on the bill.

**Amoroso:** I know. Isn't that funny? The kids, they're right. What's the difference between telephony and HTTP? To them, it's all the same. It's texting, it's IM, it's MySpace, it's Facebook, it's talking on the phone—they're all the same. Their question is why would we keep track of and bill you for phone services and not keep track and bill you for Web services. The answer is that they grew up in different places and in different technologies. Ultimately, I don't think there's anybody in the industry who wouldn't say that at some point, the blending of services will become very natural. You see it now with VoIP.

The privacy thing is some-

thing we're well aware of. As a service provider, the word "service" means that we're there to serve, and we're there to serve society and our customers. It's by no means anybody's intention in the telecom industry to be crosswise with your customers. You do that and you're going to be out of business real quick. What we're always trying to do is balance the needs of protecting our infrastructure, doing R&D, doing the right thing, following services like toll fraud, and then if there are things that are uncomfortable—for example, me sitting and monitoring your DSL connection looking for inbound/outbound attacks—you probably would say, "I prefer you don't do that," and we don't.

We create these bounds and constraints based on a model of what we think is the right thing for customers. It isn't easy because, like I said, the toll fraud has been a very successful run, but you and I both know that you can't apply the same model that worked for telephony to Internet service. It doesn't work, and you'd be run out of town if you tried it.

**McGraw:** I want to switch gears. You're pretty deeply involved with *The Hugh Thompson Show*, on which I've been a guest myself. What do you hope to accomplish with that show?

**Amoroso:** We started streaming little videos, short clips on att.com about security events as they were unfolding. I mentioned a little while ago that we were trying to find a way to identify patterns of attack as they're happening. Well, we've had some success, like that old Slammer worm from 2003 and Nachi and Blaster and all these kooky worms that were bouncing around, and now DDoS [distributed denial-of-service] attacks from botnets. We see them building up speed. Rather than stand on the hill

with a megaphone and scream out to everybody—we can't do that—but we realized that we could send alerts and page people and make these little videos. I built a little studio and we've literally gone around powdering the noses of our analysts and put them in front of a camera.

The next thing I know, I've got this infrastructure for doing that, and they said, "Why don't we start doing some news?" So we started doing a daily news show that you can get at <http://att.com/tech> channel. We hired some anchors and then we started doing some other shows.

We're going to keep doing them. I think we probably have done anywhere between 60 and 80 episodes, and we're full throttle. We love the show. In fact, at the RSA show in spring 2008, it will be the second keynote. We're actually bringing the whole set, the band, the whole thing, and do a live *Hugh Thompson Show* for the RSA crowd.

You can find additional podcasts in the series, including those featuring Chris Wysopal and Mikko Hyppönen, 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. He is a member of the IEEE Computer Society Board of Governors. Contact him at [gem@cigital.com](mailto:gem@cigital.com).