Bruce Schneier

Schneier on Security

A blog covering security and security technology.

« Bush's Watch Stolen? | Main | Real Life/SecondLife IT Security Video »

June 12, 2007

Teaching Viruses and Worms

Over two years ago, George Ledin wrote an essay in *Communications of the ACM*, where he advocated teaching worms and viruses to computer science majors:

> Computer science students should learn to recognize, analyze, disable, and remove malware. To do so, they must study currently circulating viruses and worms, and program their own. Programming is to computer science what field training is to police work and clinical experience is to surgery. Reading a book is not enough. Why does industry hire convicted hackers as security consultants? Because we have failed to educate our majors.

This spring semester, he taught the course at Sonoma State University. It got a lot of press coverage.

No one wrote a virus for a class project. No new malware got into the wild. No new breed of supervillian graduated.

Teaching this stuff is just plain smart.

Posted on June 12, 2007 at 2:30 PM • 26 Comments • View Blog Reactions

To receive these entries once a month by e-mail, sign up for the Crypto-Gram Newsletter.

Comments

Of course Bruce,

I totally agree. Teach the technics don't make malware makers. And if so, the trade off to build a generation of trained scientists that know the anatomy of malware is a very good one to research on protection.

my 0.02c

Posted by: Jacson Querubin at June 12, 2007 3:13 PM

There have already been several other schools that have taught similar courses, so this is nothing new. BCIT in Vancouver and University of Calgary taught similar courses well before this.

I wouldn't be opposed to such courses if the professors were actually teaching the material to help improve coding practices and solve security issues. But the ones I've encountered are only grandstanding and seeking to draw publicity so they can justify keeping their teaching position.

At an open presentation, I asked the instructor of one of these courses if he had taken any of the "lessons learned" from his courses and used them to teach better coding methods in the other programming classes. The "deer in the headlights" look followed by the comment "gee, I hadn;t thought

of that" told me all I needed to know...

Posted by: They're Only Seeking Publicity at June 12, 2007 3:55 PM

I agree as well, Bruce. As a cyber security student at a small college (which has limited funding for IT to begin with) I would much prefer studying the code and behavior of viruses/malware rather than reading about them in a book and not knowing how to apply that to security solutions when I get into the "real world". A lot of colleges/universities suffer a bad experience from one student who abuses their opportunity to learn and instead of recovering from it, they (the school) shut the entire program down. Our schools need to be more trusting (as well as "one up-ing" the students by better isolating environments for testing) in order to properly educate their students to protect the future's infrastructure.

Posted by: Sean at June 12, 2007 4:01 PM

So three "security software companies" responded with hostile and threatening letters. It leads me to wonder what those companies considered such a threat to them.

Posted by: Eric Norman at June 12, 2007 4:05 PM

I totally agree with this statment.

"But if we are going to develop better security, we need to know how these programs work."

We can't be expected to know something that we have not been taught.

Posted by: Linda at June 12, 2007 4:25 PM

"Teaching this stuff is just plain smart."

The most natural counter-argument is an obvious strawman: 'If we hide our heads in the sand like ostriches then the problem won't exist.'

More sophisticated counter-arguments often sound like no more than sophisticated versions of that strawman, wrapped up in a few layers of obfuscation and misdirection.

If someone has a non-strawman counter-argument, then could they please sketch it clearly, plainly and concisely? Twenty-five words or less?

Posted by: nedu at June 12, 2007 4:29 PM

I agree and do the same.

In what sport do the defense never learn the offense?

Posted by: Rich at June 12, 2007 4:51 PM

In my 400-level security class (2 years ago or so) our prof tried to explain some of the techniques and practices of malware writers, but as an academic his information was about 5 years behind the wild. That prof is fairly well-known in academia for the security field, but he was telling us about the threats from 1999 like they were the up-and-coming threats of 2005.

I think this underscores a big problem with the idea of teaching malware in academia: it is ahead of the game on some areas, and way behind on others. Especially in more theory-oriented computer science curricula, the focus isn't on today's big thing but on the underlying models -- the implications of knowing what PSPACE is isn't going to be obsolete in 6 months, or 6 years. The understanding students will get studying the mechanics of today's malware won't really help them fight tomorrow's, any more than knowing COBOL will help you with J2EE.

Not saying there's no value in it, mind you. Just that the value to be extracted from those studies is the abstracted understanding, not the implementation.

Posted by: complich8 at June 12, 2007 5:25 PM

You have to know your enemy, if you are going to be able to win over him.

Posted by: Kai at June 12, 2007 5:26 PM

Consider a simple model:

There are two networks: White and Black. The White network competes with the Black network. Internally, both the White network and the Black network cooperate. But, neither network is perfectly efficient at communicating--there is a definite speed of communication among the nodes. Further, both networks are leaky--that is, messages that are widely communicated among nodes in the White network propagate to the Black network, and vice-versa--again, at a definite speed of propagation.

Nodes in both networks "discover" messages to be passed to other nodes in their network. The network that manages to communicate most effectively among its own nodes, while perhaps degrading communication among nodes in the competing network, will have a strategic advantage.

The White network is better organized and has more efficient communications than the Black network.

QUESTION:

Should the White network attempt to degrade its own message-passing, in order to gain a strategic advantage?

Posted by: nedu at June 12, 2007 5:30 PM

I agree fully, Bruce.

If you're a convenience store chain, your best bet at making your stores robbery-resistant is to find out what robbers look for in a target. If you're a detective, you want to understand how and why serial killers do what they do.

It isn't so much the mechanics--buffer overflows and asking for change for $100 bills and extension cords slipped around someone's neck in a dark parking lot are not in themselves terribly helpful in preventing such occurrences--the way that the perpetrators look at things and the reasons they choose one method over another and one victim over another are the things we want to become aware of.

Posted by: W^L+ at June 12, 2007 8:02 PM

"What I cannot create, I do not understand." -- Richard Feynman

Posted by: Martin Rodgers at June 13, 2007 2:43 AM

Lets see here. I can compare two universities in Europe where I have studied:

University 1: During a course covering network protocols I got my accounts suspended for "breach of ToS". I had installed nmap and ethereal during class and used them to examine how computers on a to various packets.

University 2: "You want to build a DoS client to test how our experimental routers handle packet storms? Sure, go ahead. Just make sure to disconnect from the lan first"

In computer security there is far too little academic research into viruses and malware... It's a strange contrast to biology for example.

Posted by: Student at June 13, 2007 3:25 AM

I allude to the A-life qualities of computer viruses and worms in my introductory computational biology class. Seems fair game to me.

Posted by: Ian Holmes at June 13, 2007 5:19 AM

When every other country adopts this method Germany will suffer from incompetense since they are creating legislation to criminalize people who has, maintains or creates "hackingtools".

Posted by: SixDays at June 13, 2007 6:20 AM

Well, if there is a counter-argument, then I believe it stems from how much you have to trust the pupils to keep their hats "white" - both in the school, and in later life.

Worst I did at school was write a logon simulator, phishing for econet login details on the BBC B network, just to see how feasible it was - learn where the machine's network address was in RAM, pick up a sense for the things that could be logged about me - and I never abused the results enough to get caught, while I did spot the telltale signs of other kids attempting to do the same thing in the process, too.

Now I'm a sysadmin, hopefully with a suitably permissive/exploratory attitude, as well.

Posted by: Someone at June 13, 2007 7:01 AM

It is essential to study malware to write secure code. It is also essential to learn to do it yourself so that you better understand the techniques. There's no reason you can't do these things.

Posted by: C Gomez at June 13, 2007 7:31 AM

@Eric Norman, those companies were grandstanding, just like the profs cited @They're only seeking publicity

@Jacson Querubin (and @Bruce), technology is value neutral. But neither technology practitioners nor bystanders are likewise...

Posted by: guvn'r at June 13, 2007 7:57 AM

Of course they're upset. You can't have universities teaching amateurs how to write better viruses than

Symantec's and McAfee's sweat shop "professionals" crank out. How could they possibly protect against unknowns?

Posted by: sarcasm at June 13, 2007 11:28 AM

Most of the programmers we hire fresh out of college (and even more experienced ones) write plenty of viruses and worms: They just don't realize that's what they're doing. At some point, they have to see the difference between well-written code and errant code.

Well, the larger problem is that we get so many with I.T. or C.S. degrees who can hardly program at all, but I suppose that's another show.

Posted by: Bachus Naur at June 13, 2007 1:00 PM

In Cooking 101 we learned how to grill a squid. Every cook should know how to properly prepare squid.

Posted by: Stan at June 13, 2007 1:26 PM

Hey Bruce, have YOU ever written a virus?

Posted by: Jaguar at June 13, 2007 1:57 PM

"a non-strawman counter-argument"

I was hoping Marcus would comment, but I'll take a stab at it.

Writing/designing viruses is too narrowly focused and not directly useful in itself and it is not useful for learning to defend against viruses, etc. For that you need to understand the vulnerabilities, which studying viruses will teach you not as well as directly studying the vulnerabilities themselves. Basically, it's a waste of a student's valuable class time.

Posted by: Anonymous at June 13, 2007 2:28 PM

Before teaching kids to write viruses, we should teach cops how to rob a bank, rape women, kidnap kids for ransom and get away with it, doctors how to develop a new virus with no known antidotes at the moment, architects how to blow up bridges, etc.
Of course this teaching has to be in real live situation and not in highly restricted labs.
Don't you see that protecting something and destroying the same things need different skills.
Of course an airplane designer will know how to destroy a plane with minimal effort, but a terrorist is not necessary able to design airplanes.

Posted by: Somebody from the security industry at June 15, 2007 1:08 AM

I begged my college professors to teach me how to be a hacker. Instead, they gave me a good grounding in classical computer science and modern software engineering, and I think that this background is good enough for most undergrads who want to delve into information security topics. If there are changes to be made, maybe they should add elective classes on linkers and loaders (a topic my professors mentioned only briefly in their operating systems and compiler construction courses) or reverse engineering (though this could be on shaky ethical ground). These topics are of general interest, in keeping with my professors' desire to provide students with a broad understanding (instead of teaching individual tools or techniques). A sufficiently motivated undergrad who wanted to study

malicious code construction in detail could already do so as part of the mandatory senior project, and the existing software engineering classes focused a great deal on failure modes and the attendant risks (not just security risks, but also risks of injury, etc.).

Posted by: Matthew X. Economou at June 15, 2007 10:42 AM

I find this talk about teaching virus writing in class amusing. Amusing, because at the University of Florida they were teaching a class were an assignment was to actually write a computer virus. We also had to write the detection and cleaning algorithm and we would receive and instant F (retro actively if necessary) if it was ever found in the wild. It's amusing that people are thinking this is something new, and no one has bothered doing it before. I took this class back in 1995 I believe (it's been a long time, I'd have to look at my transcripts to know the exact date and who the professor was). The class was called "Computer and Network Security", but the major project was writing a computer virus.

Just to make it more interesting I took as an elective called "Legal and Social Issues in Computing". In this class, which was a mix of English, journalism, medical, and engineering students, the professor actually posed the question to the class "Should we teach computer science students how to write computer viruses". It was a good discussion, which I think surprised many people. At the end of the class the teacher mentioned, to many students surprise, that a class at UF did teach students this. Even more to the surprise of the other students was when the teacher asked if anyone was taking this class, and three of us raised our hands. Needless to say several students had a slightly stunned look on their faces. It was a great class and taught me a lot about the human computer relationship.

Anyways, just think it is all amusing. Seems old news to me, but I guess it all depends on what an individuals experiences have been.

Posted by: David Cafaro at June 16, 2007 5:40 PM

Post a comment

Name:

Real names aren't required, but please give us something to call you. Conversations among several people called "Anonymous" get too confusing.

Email Address:

E-mail is optional and will not be displayed on the site.

URL:

Remember Me? Yes No

Comments:

Preview **Post**

Powered by Movable Type. Photo at top by Steve Woit.
