Interview with Elias Levy (Bugtraq)

Kurt Seifried, kurt@seifried.org

Bugtraq is probably the best security mailing list around. However while the quasi-founder (technically Aleph1 didn't start Bugtraq as I was surprised to find out) is quite prominent online I wasn't able to find any detailed information about him or Bugtraq (except for one old interview). So here for you to enjoy is an interview with Aleph1.

Kurt: Where does the name Aleph1 come from?

Elias: Its comes from transfinite mathematics. There exists many "infinite" numbers or sets. The first infinite number is small omega or alef null. It is also called countable infinity. Many infinite sets can be mapped one-to-one with each other. For example, the set of all natural numbers can be mapped one-to-one with the set of odd natural numbers. Yet one is a subset of the other. Both these sets are said to have a cardinality of alef null. Alef One is the first cardinal number after alef null (i.e. the first set that cannot be mapped one-to-one to a set of cardinality alef null).

Alef one is a funny number. One of the reasons its difficult to grasp is because it is a regular number. A number 'n' is regular if it cannot be represented by the sum of less than 'n' ordinals less than 'n'. Of all the ordinals between 0 and alef one only 0, 1, 2 and alef null are regular. You can't write 0 as the sum of less than 0 terms less than 0 (doesn't make sense). Nor can you write 1 as the sum of less than 1 terms less than 1. Nor can you write 2 as the sum of less than 2 terms less than 2. If you live in an universe were all you known is the number 1 how can you grasp the number 2? Alef One is also regular. Its difficult to reach alef one from bellow.

As I mentioned alef null is the cardinality of the natural numbers. The question of whether the cardinality of the real numbers is alef one is called the Continuum Hypothesis (since the real numbers are used to represent a continuum). Interestingly is has been proven that whatever the cardinality of the real numbers is its also the same cardinality as the set of points in a continuous line, continuous plane, and continuous volume. The Continuum Hypothesis was the first problem in Hilbert's list of 23 important unsolved problems in mathematics. It has been proven that the Continuum Hypothesis and its negation are both consistent with standard set theory and logic, and are thus independent of it. Yet the Continuum Hypothesis has not been adopted as an axiom of set theory.

I picked up that handle many years ago while reading the book "Infinity and the Mind" by Rudy Rucker. I also recommend the book "The Mystery of the Aleph" by Amir D. Aczel and "White Light" by Rudy Rucker. The latter is a science fiction book written by Rucker that deals with infinity. Rucker is a very interesting man himself. He is a professor of mathematics at San Jose university and author of such science fiction mind bending books like Software, Freeware, Wetware and The Hacker and the Ants.

Kurt: What prompted you to start Bugtraq? When you started Bugtraq what the public reaction, if any?

Elias: Bugtraq was started by Scott Chasin, not me. Scott started the list as a reaction to a lack of useful information about security vulnerabilities. At the time CERT was almost useless (read some of the earlier advisories and try to figure out what the problem is) and vendors did little to help. Systems were ridiculously easy to break into. System administrator started to depend on each other to stay informed. The firewall mailing list was used some times to discuss vulnerabilities but it was outside it charter and the list owned did not want exploits or detailed information on his list. This environment prompted Scott to start BUGTRAQ in 1993.

At first BUGTRAQ was no more that a few hundred users. I took over the list in 1996 after Scott decided he was no longer interested in managing it. Since then the last has grown up to a peak of over forty thousand subscribers.

Kurt: Long term, do you see Bugtraq continuing in the same vein as it does now? Do you have any long term goals for Bugtraq?

Elias: As BUGTRAQ and the Internet have grown so have the number of topics and messages send to the list. This means that many people receive information that are interested in. Subscribing to BUGTRAQ can sometimes seem like sucking water from a fire hose. My main goal for BUGTRAQ at the moment is to figure out a way to either implement filtering by topic server side or facility such filtering on the client side. We are investigating possible ways to implement this feature (suggestions are welcomed). I hope that this will make BUGTRAQ more manageable for the subscribers and more accessible to the many people that wish to subscribe but don't because of the message volume.

Kurt: For system security, do you think the future is in auditing and producing secure code, such as OpenBSD, or in things like MAC and type enforcement, such as the NSA's SELinux or Argus Pitbull for Solaris?

Elias: Actually, I see the future being almost as bleak as it is today. Auditing efforts like the one from the OpenBSD project are a must. They are a baseline. Systems that have not gone through a source code security audit should include a mandatory tag that says "Lasciate ogne speranza, voi ch'intrate" (Abandon hope all ye who enter here).

But even so I find UNIX's security model as being to simplistic. The all or nothing approach does not lend itself well to the principle of least privilege or to defense in depth. In such simple model running an insecure application in a well audited operating system will defeat the benefits of the audit. The folks from OpenBSD and similar projects can only audit some many lines of code in a given amount of time. The trusted computing base will never offer all the feature users want.

On the other hand I find most of the implementations (and many times the ideas) of mandatory access controls, type enforcement, privileges, etc as being to complex. They provide more flexibility and fine grained control but at the expense of ease of use. A complex system is seldom implemented or configured correctly resulting in vulnerabilities. I find this is exacerbated when you try to bolt these security models to an existing one (as is the case with many of the implementations now available for Linux). They result in difficult to predict interactions that many result on vulnerabilities (recall the sendmail vulnerability that was the result of privileges being introduced into the Linux kernel).

In a perfect world we would be able to find a middle model that provided ease of use, transparency and flexibility without being simplistic or overly complex. The model would be implemented as part of a new operating system so that there would be no conflicts with some previous model, and which would allow applications to make full use of the model's capabilities.

Of curse this is a pipe dream as an operating system is only as useful as the applications available for it, and the resources required to write all the applications users expect to be available is astronomical.

Kurt: Do you think the majority of commercial vendors currently take security seriously, or are they simply paying lip service to it? Do you think the will improve, stay the same, or get worse in the future (as features and connectivity increases)?

Elias: I don't think this is a meaningful question. Corporations only purpose is to generate money. Software vendors attempt to make money by selling software. They write software based on what they believe are their customers priorities. If they match those priorities well they will succeed and make money. If they don't they will fail and loose money. Software vendors will only "take security seriously" when their customers do. Until then they have no incentive to write secure software.

This is an oversimplification but its the basic truth. Things are somewhat complicated by the fact that some operating system vendors have monopolies and can sometimes ignore the priority of their customers. Also corporations not only want to make money, they also don't want to loose any. So the fact that software vendors are not liable for writing insecure software means they have one less reason to write secure software. And the fact that the are third-party products that help alleviate the security concerns (albeit not very successfully) gives software companies one less reason to write secure software (let the people that want security buy an add-on package).

But in the end its the publics fault.

Things will continue as they are until vendors become liable for security problems in their products (an unlikely event given the lobbying done by the industry unless the fear mongers are correct and we have an "electronic Perl Harbor" and the government steps in), or until insurance companies start offering computer security insurance and charge astronomic rates unless you are using products from vendors that make security a priority (this is a more likely event).

Kurt: As computers become increasingly connected (high speed ADSL, cablemodems, etc.) and software like ShareSniffer becomes increasingly prevalent do you think things will get much worse, or will users become more savvy and protect themselves?

Elias: Things will be worse for the foreseeable future. For each person that learns a lesson in the school of hard knocks two new people join the Internet. This pattern will go on for a while. The number of likely victims will continue to grow. Maybe many years from now when everyone that lives is born after the Internet and is familiar with it people will be computer security savvy by nature. But that is a long ways off. Today you need no "drivers license" to join the Internet. You are given no advice before you are thrown into the deep end of the pool. Heck, most people only known enough about their computers to dial-up to the ISP, read their mail and browser the web. How can we expect these people to securely manage a general purpose computer?

Kurt: If you had one wish to improve computer security, ignoring backwards compatibility as so on, what would it be?

Elias: Mandatory computer security class for CS majors. rm -Rf * all software (and some hardware) and start from scratch.

Generally speaking I agree with Aleph1's views, especially the fact that users have not been demanding security as an important factor in products. While it is unlikely that we will be able to start from scratch completely there are some efforts such as OpenBSD, NSA SELinux and Eros to build a better foundation.

Reference links:

http://www.securityfocus.com/ - Bugtraq

http://www.eros-os.org/ - EROS

Back

Last updated 4/10/2001