Interview with Rain Forest Puppy

From: InfoSec News <alerts () infosecnews org>
Date: Tue, 8 May 2007 00:19:23 -0500 (CDT)

http://www.ush.it/2007/05/01/interview-with-rain-forest-puppy/

May 1, 2007

Antonio `s4tan` Parata, software security researcher and member of the ush team interviews Rain Forest Puppy, famous bug hunter, specialized in web application assessment. It's a pleasure for us to publish the full interview, in this case talk is not cheap.

Antonio "s4tan" Parata (ap): Hi Rain Forest Puppy, many thanks for this interview. You are considered one of the fathers of web security and the inventor of the SQL injection attack. Anyway in the year 2003 you decided to publicly retire from the security field (to get more infos http://www.wiretrip.net/rfp/txt/evolution.txt). Can you briefly sum your decision?

Rain Forest Puppy (rfp): My decision to retire from the public eye was based on a lot of reasons; overall, the amount of resources & energy required to release and maintain advisories and tools was just getting to be too large. It wasn't fun anymore-and why pursue a hobby if you're not enjoying it?

Plus, the security industry was becoming commercialized. Advisories and exploits are now bought and sold; performing security research in the first place can land you in legal waters. The intellectual value of the security research performed has been reduced to a single severity rating, which…if not high enough…causes the entire research to be dismissed. I really enjoy security from the intellectual angle; to me, it's all just a big mental challenge…a puzzle, if you will. So when the creativity and intellectual aspect of it started to fade away, I decided to go with it.

As for being the "father of web security", there were many people working on web security prior to me (for example, see Lincoln Stein's classic WWW Security FAQ). And I didn't invent SQL injection. I may have been one of the first to publicly explain it in tutorial fashion, but it existed for as long as SQL itself existed; it was just that few people saw the security implications of it. But that may be because SQL wasn't ubiquitous like it is today, so it had limited impact in limited circles.

ap: 4 years elapsed and the web changed radically. Phrack is dead, Owasp testing guide raised, the web is filled with blogs and the web 2.0 buzzword is on everybody lips. How did your thought change in these years and what do you think about nowadays security world, who works in it and researchers?

rfp: Well, the good news is that there is an increased awareness for the need for security. That's a good thing. Even consumers are starting to understand the need for personal firewalls and the need to be vigilant when online.

The flip side of that awareness is that people now care when they have security–or more importantly, when they don't. Combined with the litigious society we've become, and now you have the very real threat of someone pursuing legal action against you for informing them they have a security problem. Now that security can be linked to tangible dollar losses, and security regulation violations can have drastic impacts, I've witnessed first-hand companies who felt it better to be in the dark and cover up any signs of security issues rather than having those security problems disclosed and thus being forced to deal with it. It's the Enron approach to security.

But, like I said in my Evolution essay (a.k.a. rant), security is now a big-time commercial business. There's money to be made in having it, improving it, breaking it, exploiting it, etc. That's probably the biggest change. Although, I suppose I'm part of the problem, having a security-related day job. :)

ap: At the moment are you working for a security company or are you an independent consultant?

rfp: I work for a security company. In fact, at the beginning of this year, I started working for a security software vendor. Prior to that, I worked at the same small security services company for 7 years, performing pen-tests, web app assessments, source code reviews, etc.

ap: What do you think about companies like gleg (http://www.gleg.net/) or iDefence (http://labs.idefense.com/), parties that make part of their profits from the selling of 0day exploits?

rfp: Well, I have mixed feelings. Part of it is how you frame it too… saying iDefense and 3Com sell 0day is only half right. Sure, they inform people of those 0day problems. But, they also handle the overhead of dealing with the vendor, coordinating advisories, etc. All that stuff takes time and resources, and can be particularly frustrating if you happen to deal with a vendor who doesn't understand the security disclosure process (see my previous answer about Enron-style security silliness). So, being someone who likes to find bugs, and wants to do the right thing (i.e. inform the vendor) but doesn't necessarily like the hassle of dealing with the vendor, iDefense & 3Com seem to be a win-win situation: they deal with the vendor, and you get paid for your research time (and the dwindling of low-hanging fruit and increased complexity means more research/time is required for each bug).

Part of my answer to this question ties into the next question…

ap: You are the creator of rfpolicy (http://www.wiretrip.net/rfp/policy.html), globally recognized as the policy to follow for the vulnerability disclosure. What do you think about mailing lists that practice full disclosure like FD (http://lists.grok.org.uk/full-disclosure-charter.html)?

rfp: In the end, it all comes down to the motive of the researcher:

* Trying to make the world a more secure place
* Trying to make a buck
* Trying to impress their friends/peers

Each of those has it's own response. If you're truly trying to make the world a safer place, then the only way to do that is to pursue a fix (and that typically means dealing with the vendor/author); if, for some reason, the discussions with the vendor are going horrible and you've exhausted all other options, then full disclosure to the public is a last-ditch effort to at least get the warning out.

If you're trying to make a buck, well, sell it to the highest bidder. There's been a lot of media reporting in the last 6 months about 0day black markets, and iDefense/3Com occasionally hold specials where you get paid extra for certain types of vulns (remote Vista bugs in particular).

If you're trying to impress your friends/peers, then just run straight to the disclosure lists/venues. You'll have your five minutes of fame until the next bug comes out. Hopefully though, you won't pursue a security job down the road with a company who has negative feelings towards full disclosure…your efforts to build your 'cred and impress your friends now may backfire later when you look to start doing it professionally. Remember, the Internet archives everything these days…

What probably bugs me the most is that a lot of people have the "trying to make the world a more secure place" facade, even though that's not really their true intention. I call it the "MS. America 'World Peace'" phenomenon, after all the pageant contestants who say they want world peace because that's what they're supposed to want in this age of political correctness. If a researcher truly wants to make the world a more security place, then they need to attempt to get a solution to their problem, and that usually means making some attempt to work with the vendor.

The moral to my long-winded answer: full disclosure is a tool, not a solution. Use it wisely, and where appropriate. If you truly want to be part of the 'security solution', then offer a (realistic) solution when you have a problem to disclose. Be responsible. We control our own fate: if we run around like Internet Anarchists, then laws and regulations are going to tighten and make things more difficult. If we act responsibly, we may be able to continue with what we're doing as-is.

But you can't have it both ways.

ap: What policy to apply in the case of public site vulnerability research? Should the researcher avoid it completely, apply the rfpolicy or the full-disclosure way is viable too?

rfp: Funny, because I was just mulling this over recently. It's one thing to have a security problem in something you control, such as a device or a piece of software installed locally. There's the potential for you to enact a workaround or introduce another mitigating control.

Public websites are another matter. The only one who can fix the problem is typically the web site. There's no mitigating strategy users can usually do other than forego use of the site. You think everyone is going to cease to use MySpace because they have an XSS hole? No way.

So thinking that it's better to tell the world about a security problem in a public site than to tell the site owners is being part of the problem, and not the solution. Again, full disclosure is a tool, and is a worst-case/last-ditch scenario after all else fails.

ap: You are the author of the libwhisker library (http://www.wiretrip.net/rfp/lw.asp), widely used to create assessment perl scripts. What do you think about nowadays products related to web application assessment? What about some open source software (like parosproxy or nessus) changed to closed-source?

rfp: I have to choose my words carefully, because I very recently started working for a security software vendor. :)

Having had open source projects, I will say this: it is very hard to bootstrap a development community, and achieve the same level of polish, quality (as in QA), and implementation thoroughness as a commercial product. This isn't necessarily because commercial software vendors are better coders; the dynamics are just different.

Open source coders are usually working on their own donated time. That means contributions are often catch-can and best-effort. Open source (when not sponsored by a commercial entity) are typically limited in resources (with time being the critical one).

Commercial companies, on the other hand, don't necessarily have a constraint on resources and time, because they can be bought. And they are bought with the money used to purchase the software. However, because the software is purchased, they have the additional obligation of making sure it satisfies the user and the user's experience. That usually means better UIs and usability, full feature sets, and thoroughly implemented features with all the bells and whistles a normal user would expect for that type of product.

If anything, I would say the bar is set higher for commercial products, because purchased software has certain additional expectations and obligations to live up to. If you grab a free suite of open source software, and something in it is broken or it doesn't implement some basic functionality which you deem fundamentally necessary… well, your only recourse is to submit a bug report or feature request. It's free, and because of that, there's not necessarily an obligation to satisfy you as a user. But if a commercial software package is broken, or it's missing something fundamental, you can ask for your money back, or make a request to the vendor to fix it with a reasonable expectation that they will. If they don't, you have recourse with entities such as the Better Business Bureau (in the US).

Given all of that, I have made a few observations on how open source relates to commercial products:

* Commercial vendors don't draw from a different, exclusive pool of uber-developers. Good, smart developers can exist on both sides of the fence; in fact, often times they play both sides. So the concept that commercial vendors magically have better coders that are more capable of solving a problem or being innovative is a fallacy. An open source project can be just as innovative as anything a commercial company pushes out; the difference is that the commercial company can usually push it out farther and wider.

* The really good/innovative open source projects often go on to either form a commercial entity, or gain commercial sponsorship. This almost makes open source a research incubator and proving ground for new ideas (which, IMHO, is great). The good ones take off and develop into large entities (Apache, Samba, MySQL, etc.) and the rest live out the remainder of their lives on SourceForge. :) But once an open source company gets commercial backing, there then becomes the requirement to satisfy the conditions of that commercial backing…so the sponsorship usually provides resources in exchange for better meeting the obligations/expectations that come with traditional commercial software.

  In that sense, sponsored open source sits on the fence between normal open source and commercial software, probably getting the best (and worst) of both worlds.

* I made indication of it in my previous answers, but despite open source being free and best-effort, many users still hold it to a commercial product expectation of quality, implementation thoroughness, etc. This is where I think a lot of problems arise. Yes, open source software should be as good (or better) than commercial software, even though it is constrained by resources. But we all know that's usually not the case…something as simple as a clean UI and better documentation is all it takes to give something a

commercial-level appeal/feel. My personal experience with open source is that these are the areas where they most often tend to lack.

So, going back to your original question about security tools: the security industry is such a hot topic, that everything is in such a state of flux, that it's hard to say. Established open source tools have migrated to commercial backing (nmap, Nessus, ParosProxy, etc.).

There's a lot of tools which are the byproducts of commercial research, and/or being used for marketing purposes (all the great Foundstone tools, HTTPrint, etc.) Some of these have no identical/suitable commercial counterpart. And yet there are many commercial tools which don't have effective open source counterparts (I haven't seen a good open source static source code analysis tool yet on par with Coverity, Fortify, or Klocwork). There's no open-source equivalent for what AppScan and WebInspect fully do.

In the end, I've developed my own personal approach. All I care about is whether the tool works and/or gets the job done. I've spent so much wasted time trying to get a screwdriver to do a hammer's job, and vice versa. I really don't care if a tool is open source or commercial; I let the job dictate the tool, and not the other way around. Of course, there are certain artificial restrictions on this (like price limitations), but in general, I think there are some things that currently only exist in free & open source tools, and there are some things that currently only exist in commercial tools.

So use both wisely and get the best of both worlds. :)

ap: What's your method to keep yourself updated on security news?

rfp: There's just too many sources of information these days to digest. I have a very large RSS feed list I try to keep on top of, and I keep tabs on a few traditional mailing lists. I find that, if something is big enough, it will usually trickle down onto the security mail lists or one of the popular security blogs, which tips me off and I do further research on it from there.

So I suppose a good analogy is: rather than waiting to hear about stuff from the horse's mouth (especially when there are many horses), I wait to see what interesting things the manure handlers heard or found after it passed through the horse. :) (note: I can neither confirm nor deny the intentional comparing of manure to the information content on some of today's blogs…)

ap: Which books have you read lately? Is there any book that has to be recommended anyway?

rfp: I currently like "Developing More-Secure Microsoft ASP.NET 2.0 Applications" by Dominick Baier. Rather than being a 'security 101' approach filled with lots of overhead most seasoned security professional already know, this book is almost like a collection of technical tips and insights into little topics, all with security relevance. I like to think it fills in the remaining small gaps that the seasoned pros might have.

Nowadays though I really don't read books in the traditional manner…there's just too many coming out. And to make matters worse, they're expensive and often don't contain material that satisfy me. So I use O'Reilly's Safari, which lets me search for specific topics across a whole library, and just download PDFs of the chapters I need. It's more efficient and cost-effective. Occasionally I'll check out the bookstore's selection for books that aren't hosted by Safari, but Safari has a good selection overall.

ap: Is your life style Infosec related even in your spare time or do you have extra IT&C hobbies?

rfp: A lot of things have changed since I faded out of the public eye in

2003. At the height of my 'RFP days', I was a bachelor spending all day doing security work, and then all night doing security research…sometimes not even sleeping. Now I have a family, and I give all my spare time to them; so my security-related pursuits tend to be limited to just work-hours, with the occasional evening or weekend for a special security project.

ap: Will the Infosec community have a chance to see you back to the scenes like in the past?

rfp: Well, there's two ways to look at that question. When you consider the qualifier "like in the past", then no. Don't expect wiretrip.net to start spewing out new advisories or tools. But will the Infosec community see me involved in it? Sure. Actually, I never left. I still post to the security venues, I still publish, I still work with vendors to get things fixed, etc. I would say I'm still very active in the security community–but in a way that has nothing to do with the name RFP.

ap: Thanks rfp for the interview!

rfp: Thanks for the thought-provoking questions!

⬅ By Date ➡ ⬅ By Thread ➡

Current thread:

Interview with Rain Forest Puppy *InfoSec News (May 07)*