Interview with Marc Maiffret

RIK FARROW

After being raided by the FBI at age 17, Marc Maiffret started his first security software company, eEye Digital Security, pioneering early research into critical Microsoft vulnerabilities. As an entrepreneur, Maiffret created one of the first vulnerability management products as well as one of the first Web application firewall products—both of which have been deployed worldwide and won numerous awards. Maiffret has also been a leader in next-generation malware prevention while serving as Chief Security Architect for FireEye. Maiffret served as Chief Technology Officer of the privilege and vulnerability management firm BeyondTrust after its acquisition of eEye Digital Security. In 2015, Maiffret left BeyondTrust to pursue a new but unannounced venture. Maiffret has testified before Congress, published an op-ed in the *New York Times,* and is an avid speaker and advocate for improving security.

Quite appropriately, I first "met" Marc Maiffret online. We were both participants in a security mailing list, and I was struck by Marc's youthful exuberance. I could tell that Marc was on a mission, and that mission appeared to be to embarrass Microsoft into improving its security practices.

What I didn't know at first was Marc started his first business at 17. I'd certainly noticed the rough edges in his online postings but had little idea just how young he was or how he had become an expert in Windows security through self-education and experimentation.

Over 17 years later, I decided to ask Marc more about what he had been doing before we met, his part in some security drama (Code Red), and his various business adventures. I also wanted to get Marc's impression of the state of Windows security today.

Rik: When did you start learning about computers?

Marc: My path to learning about computers really started first with an interest in phone phreaking. I had a friend who introduced me to the world of phone phreaking in 6th or 7th grade. I always had a curiosity about how different things worked, and the telephone system seemed like this infinite world to explore and learn from. Wanting to learn more about phone phreaking led to needing to get on BBS systems, and that was my gateway to eventually getting more into computers, hacking, Internet, etc.

We didn't have enough money for a computer at home, so in the beginning I learned what I could from computers at the school library or the office where my mom worked; the owner was nice enough to let me use a system sometimes after school. That same business owner eventually gave me an old computer from their office, and that is when things really started to move quickly for me in learning about early hacking and related topics.

I had a turbulent home life growing up that can be summarized by my deciding to run away from home for almost a year when I was 16, moving entirely across the U.S. to live with friends in the hacking and research group Rhino9, my stepdad eventually dying from a drug overdose, and so on. Not to understate it all, but it was a variety of things, plus my natural curiosity about how things worked, that drove me deeply into learning as much as I could about computers as a means of escaping my then reality.

Rik: You started a business, eEye, back in 1998. What led you to develop software to help secure Windows systems, back when Windows was really awful?

Marc: When I got back home after running away from California to Florida, and a few places in between, I was 17. I did not want to go back to high school and wanted to start working in computers. My mother gave me three months to find a job and support myself or it was back to school. Within a few weeks I had my first job, and then a couple of months later all the hacking I had been doing over the previous few years caught up to me when the FBI raided my family's home. This was a great wakeup call for me to try to figure out what I was going to do with my life. At the time, I had been writing a lot of free security tools for Windows and researching various software vulnerabilities. This led to the creation of my first company, eEye Digital Security, and the vulnerability assessment product Retina. This was all around the 1998–1999 time frame.

Within a few short years, eEye and Retina had a lot of business success. More importantly, though, we were a part of pioneering a lot of the early research into Microsoft-related vulnerabilities. We also were pushing aggressively for companies like Microsoft to treat security as a technical problem instead of a marketing one.

People coming into IT security today would not recognize the Microsoft of the early 2000s. At eEye, we did not just find numerous critical vulnerabilities within Microsoft software, but rather exerted as much pressure as possible to get them to change their culture and behavior, making it as painful as possible for them while we helped to get vulnerabilities fixed to protect customers. This was a sometimes difficult balancing act which led to fun encounters, like the then head of Microsoft security response calling on the phone to curse me out. There were many people doing great security research back then, and all of this led to Microsoft evolving in positive ways. eEye had a very special role in that process, not just through research but also by having customers we could help leverage to put pressure on Microsoft.

For example, a large reason why Patch Tuesday was created was because of customers being outraged and exhausted by having to deploy critical patches for remote system vulnerabilities one after another on a random basis. A lot of people do not know that behind-the-scenes we were doing things like accumulating critical vulnerabilities that we would report to Microsoft one at a time. As soon as they patched one, within hours we would send them another, and another, to keep pressure on until something broke their poor software development behaviors.

That something eventually came in the form of Bill Gates' Trustworthy Computing memo, in large part spawned by the efforts of eEye and many others and, of course, the fallout from things like Code Red and other widespread malware/worm attacks at the time.

Rik: Tell us about Code Red.

Marc: Code Red was a Microsoft computer worm discovered by Ryan Permeh and me while we were at eEye. Code Red leveraged a vulnerability within Microsoft's IIS Web server that Riley Hassell, also at eEye, had discovered. Ryan and I were actually hanging out on a Friday after work drinking beers at his place when an IT guy emailed mentioning that their IIS Web server was acting weird, connecting to other systems. Now this is in 2001, a very different world in IT and security. You can actually find archives on IT mailing lists where people were experiencing IIS Web server crashes for a good week or two before Ryan and I made this discovery. After getting a packet capture of some IIS traffic, we started looking to see what might be going on and determined, in fact, that someone had developed a worm to automatically propagate to IIS Web servers via the vulnerability Riley had discovered.

Ryan and I worked over the weekend to write up a technical analysis of the worm and eventually posted our analysis online late Sunday or early Monday morning. We didn't think much of it at the time as Code Red was one of the first of its kind. By Monday afternoon, the whole thing had taken on a life of its own, and by the end of the week we had done everything from talked to folks in the White House situation room to the head of marketing for Pepsi, the company behind Code Red Mountain Dew, which we had named the worm after. While the worm was actually easy to manage in the end, due to its propagation method, it affected a lot of systems and was certainly a wakeup call for Microsoft.

Rik: What did you do after you left eEye?

Marc: eEye was always more than simply a business to me and to a lot of the employees there, particularly those working in research: guys like Ryan Permeh, Barnaby Jack, Yuji Ukai, Derek Soeder, Riley Hassell and too many others to list. We wanted to make a great product in Retina, but also we were a part of the early days of the security industry and were hackers trying to find our place in this world. Living in Southern California, I find conversations with old skateboarders who rode the wave of evolution from their hobby to a business to be more relatable in understanding just how special what we were all a part of was, as opposed to some person who is new to IT security these days.

When people have been in this industry all of five minutes, it is easy to think security is terrible and hasn't made much progress, when in reality a great deal of progress has been made. When I catch up with my old colleagues and we reflect on the wild ride we were a part of, it is not so much about what place eEye holds in that history but rather about hoping people understand that the evolution in security and improvements in companies like Microsoft has not happened naturally; instead they've happened because a research community was willing to fight and hold technology companies accountable. This is something often forgotten today as we focus as an industry solely on hackers and adversaries, on countries and cybercriminals but rarely on the vulnerable technology that allows such attackers to break into systems in the first place. This is something I expanded on further in a *New York Times* op-ed a couple of years ago (http://www.nytimes.com/2013/04/05/opinion/closing-the-door-on-hackers.html).

After leaving eEye, I took some time off to take a break and hang out. I had been hacking and working in security since I was a teenager, and eEye was the only job I had ever had. After a short time helping run a managed security company, I went to work for a less well known company, at the time, that also liked eye-related company names—FireEye. At FireEye, I reported to the then CEO and Founder Ashar Aziz as Chief Security Architect. They were not then the behemoth that they are now, and I was lucky to be a part of helping innovate their product in its malware detection capabilities and amplify the great stuff they were doing in those early days. Bringing my background in vulnerabilities and exploits to the malware world helped increase their systems' ability to generically detect malware and compromises within corporate networks. It was an amazing team and experience to have been a part of.

Rik: How has the Windows security landscape changed from your perspective?

Marc: The Windows security landscape has changed dramatically as it pertains to Microsoft software. The reality is that Microsoft has made amazing strides to improve the security of their code and systems and continues to do so. Clearly, many vulnerabilities remain, but Microsoft has consistently done things to raise the bar on attackers and researchers alike. There are too many examples of positive changes they have done to list them all.

Probably the biggest area of improvement is not just in their internal security efforts to eradicate bugs but in their efforts to continue to make the exploitation of vulnerabilities that much harder. This even goes to the point of their offering $100,000 bounties on novel ways to bypass their various mitigation technologies. This is a wildly different Microsoft than the one I knew many years ago. There are, of course, a lot of technical examples of how they have improved security and their architectures over the years, but more than hoping for one individual safeguard, I think the biggest improvement is yet to come in Windows 10 because of changes to the overall ecosystem.

Microsoft has realized that no matter how secure they make their own code they will still get a bad rap so long as their ecosystem of third-party developers and software remains insecure. In a lot of ways, most security products have existed as bolt-ons to harden operating systems and to more tightly control application behavior in ways operating systems should be doing by default: the age-old problems of separating code and data, users and access, and so on. Where Microsoft and even Apple seem to be moving in terms of the desktop OS ecosystem is to mirror what has happened in the mobile OS space with much tighter control of what applications can do, how they inter-operate, how they are sandboxed, and so on.

Microsoft already started down this path with Windows 8's app store but failed to get developers to adopt their new model because it would require whole code rewrites in a lot of cases, not to mention generally failing to get companies to even migrate from Windows 7 to 8. Microsoft seems better positioned to successfully get people to adopt Windows 10, and it seems to be doing everything possible to get developers on board with getting their apps moved to the app store model, including going to great lengths to allow for classic Win32 applications to be packaged up as store applications; this is done through leveraging some level of virtualization and sandboxing so as not to violate the overall benefit of store-based applications. This has interesting implications for the desktop security landscape because store/mobile OS models more granularly control and sandbox applications in ways that can be very beneficial for security and even IT management.

You can think in terms of whether you would trust a family member to be safer online via an iPad or Windows 7; which are they most likely to get hacked on? Now this is not some religious debate about what is the better OS or technology company, or which has more or fewer vulnerabilities; rather, it's a question of OS and application models that are very different in mobile OSes vs. traditional desktops. While exploits can and do exist for both models, there is a dramatic difference in attack surface and how tightly controlled applications and code are. I could expand on this a lot more, but hopefully the implications of and differences in these models are obvious as to the benefits to security if Microsoft can successfully win over developers to this new model.

To be clear, I'm not suggesting that such an app store model will magically make Windows 10 secure out of the gate. It is not that Windows 10 only allows a mobile OS app store type model but rather that it is a hybrid, as Windows 8 was, of both a traditional desktop OS app model and an app store model. If Microsoft can successfully bring developers and their apps over to the store model, then it moves us closer to being able to hit the kill switch on the traditional desktop OS app model and all the attack surface that comes with it. And, of course, there will be plenty of problems with the store model from a security perspective; expect to see someone talking about win32 apps escaping the Windows 10 Store app sandbox at a future security conference. But these problems will be far better than the current state of the Windows desktop security model, where companies struggle with a bunch of bolt-on security software simply to make sure their users are not running malicious code from Web browsers, email, and so on.

Rik: What do you think of open disclosure currently?

Marc: When discussing vulnerability disclosure, full disclosure, and related topics, it is important to understand security research in the larger context of where we currently find ourselves in the continuum of such research. Vulnerability research in the early 2000s was being done more out in the open where everyone could benefit from it.

Things have changed over time where the value of such research has increased well beyond the primary value in the 2000s of simply making a name for yourself or for a security company in order to get some press. As such the trend has been that more critical vulnerability research is happening much more often behind closed doors to the benefit of a few. It is also important to think about the increased impact a vulnerability can have now vs. years ago as society grows more dependent on technology.

So with that context in mind, I can see validity in the arguments from all sides. I understand why a researcher would rather sell a vulnerability to a defense contractor or private party than deal with the sometimes truly painful process of trying to report a vulnerability "responsibly" to a technology company, all for the reward of a thank you in a security bulletin or possibly a payment that is a fraction of what they would have made by selling it privately. I can also understand a researcher who thinks selling a vulnerability to a defense contractor is morally wrong but equally hates dealing with vendors and simply wants to drop the information online for the community to sort out.

And I can see why plenty of people would be upset at researchers who seemingly claim to do their work for the benefit of everyone but are inflexible in their own views and timelines of what a vendor might require to fix a flaw. I think this debate has persisted the last 17+ years I have been in security because there truly is no right answer or magical governing principle applicable to just vulnerability research. I think the only thing that can be said for certain is that regardless of your opinion on such debates, the debates would not be happening if the information were not public in some form. Wait, was this question about Snowden? :-)