RACK911 Labs

Year in Review

May 6, 2014

The security of the hosting industry has always been a concern of RACK911 and in May of 2013 we decided to take a more proactive role by creating a new brand called RACK911 Labs that specializes in finding security vulnerabilities before the bad guys do.

It's been exactly a year since the inception of RACK911 Labs and never in our wildest dreams did we expect to find out just how inherently dangerous the entire hosting industry is due to the use of insecure software, lazy software developers and pure incompetence.

When people think of the hosting industry in general, they tend to only focus on the amount of hosting providers out there which would easily be measured in the thousands. What often gets overlooked is how many people are *indirectly* affected when a website gets compromised or when an entire server is compromised.

There are over 950 million active websites in the world and it’s safe to assume that a large percentage of them are hosted on shared, reseller and VPS hosting environments. It would also be safe to assume that the majority of the websites in a hosting environment are using a non-proprietary control panel that we are all familiar with.

That's a significant amount of people who are part of the hosting “eco system” and that’s why we decided that something had to be done. Until 2013, we privately worked with companies to report security vulnerabilities and they would often take advantage of us by taking an unreasonably long time to issue patches and sometimes they wouldn’t even bother. There would be no pressure for them to do anything and even worse, no accountability for their incompetence.

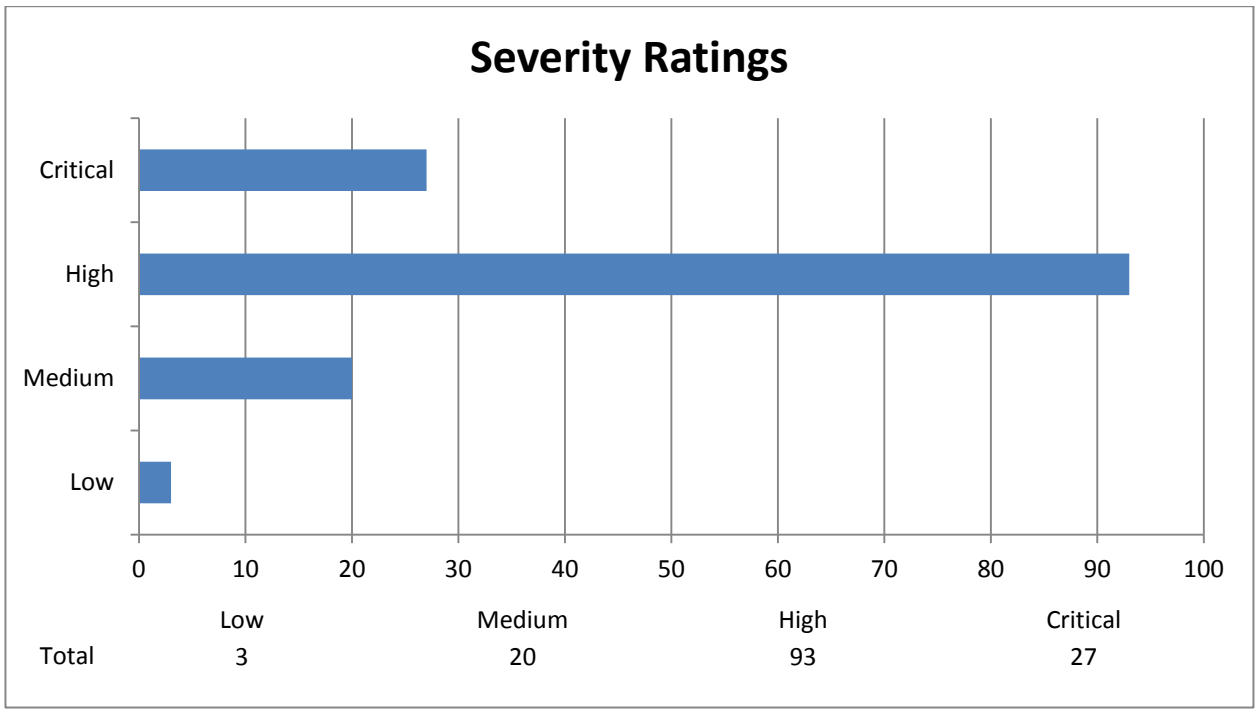
We drew the line and said enough is enough! Too many of our friends, our colleagues and the general population are being needlessly put at risk by software developers who clearly don’t care about their security and more importantly, the livelihoods of their customers. With all of the news in recent years about security breaches, customer’s data being exposed and sold on the black market it is truly unacceptable for any software developer to not make security their top priority.

Now that you understand how RACK911 Labs came about and why we are so passionate about security, we’re going to go over some of our findings from the last twelve months.

Statistics

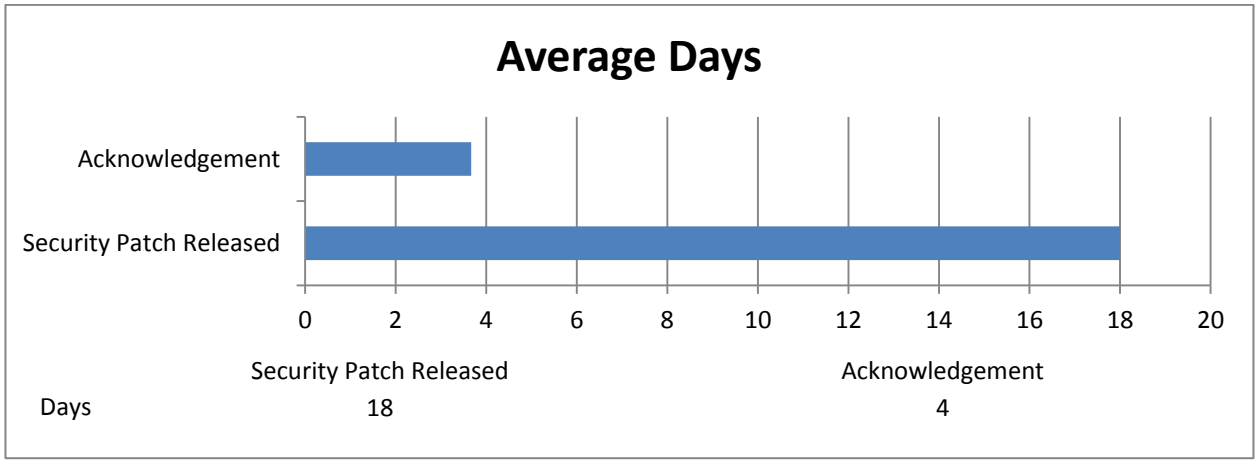
Over the last twelve months we have released 143 public security advisories with another 35 pending release any day now. We also have over 100 other advisories that were fixed privately through special arrangements with the software developer or were simply not released for various reasons. In total, we have been responsible for well over 250 security vulnerabilities most of which have been in commonly used hosting applications.

When we release a security advisory we give each a severity rating based on the threat potential for the vulnerability. For example, if interactive root or admin access can be obtained then we will give it a high to critical rating. Whereas if the vulnerability is more of a nuisance and no data can be compromised, we will give it a low rating. Sometimes our ratings are in disagreement with the software developer, but for the most part we rate them accordingly based on our assessment of what can be done in reality and what can be done under theoretical situations.



Out of the 143 public security advisories that we have already released, 27 of them were rated critical to indicate that a grave threat exists. There were 93 vulnerabilities that we classified as high meaning the potential for root or admin level attacks exist and then 20 were classified as medium to indicate that some level of data can be compromised. Only 3 vulnerabilities did we classify as low, all of which were nothing more than a nuisance attack with no threat to other users.

When it came time to reporting the security vulnerabilities to the software developers, for the most part they responded in a timely manner to acknowledge the report. The average was 4 days from our first point of contact to them getting back to us. (We would prefer acknowledgement within 24 hours as that should really be the longest that it takes a software developer to respond to a report of a security vulnerability.)

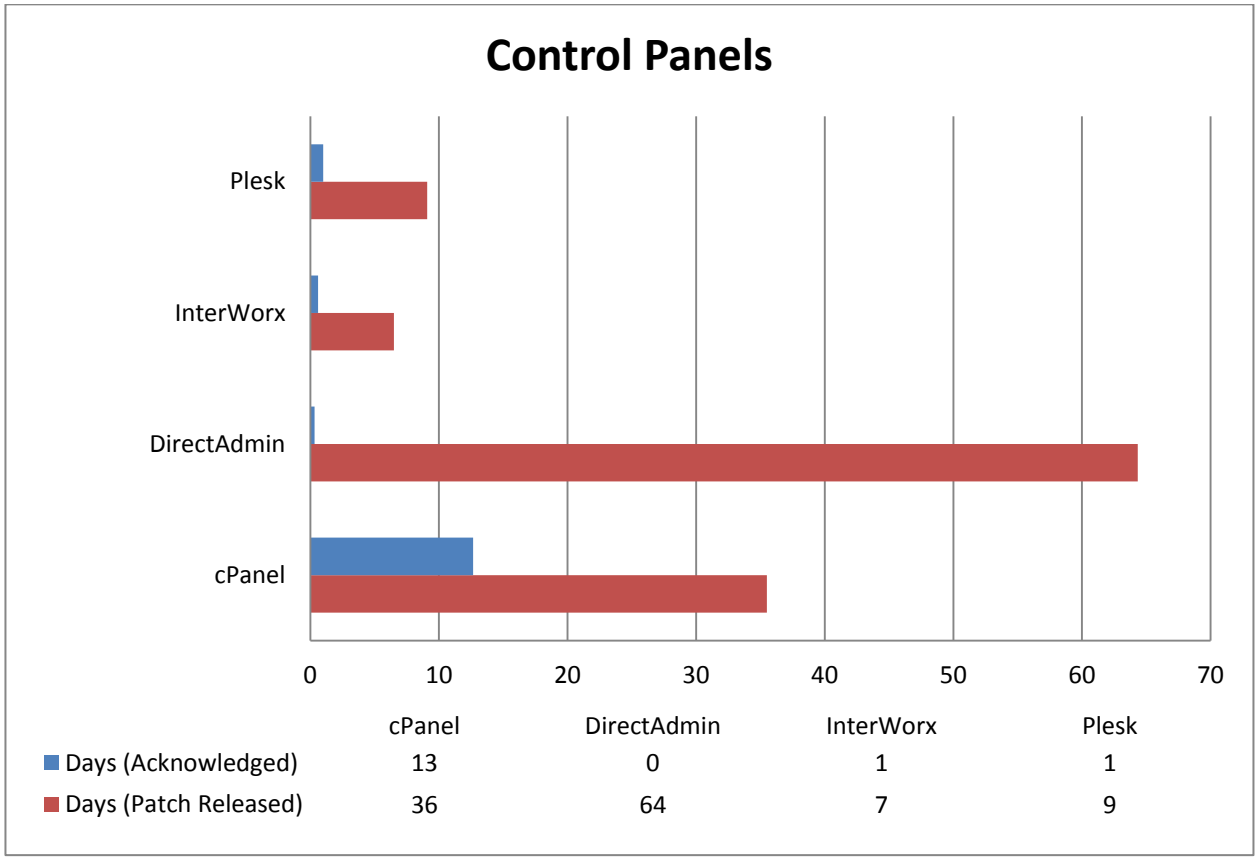


As for when it came to the software developer releasing a patch, the average was 18 days from our first point of contact. A lot can happen in 18 days and we find that number to be extremely concerning. To be fair, that number is higher than it should be based on a few companies who follow monthly release cycles which is a practice that we are viciously against when it comes to security patches.

We are a strong believer that security patches need to be pushed out regardless of any release cycle and they need to be done so with the highest priority. If a software developer is made aware of an exploit that could cause serious harm to their customers, they need to work as fast as possible to get a patch released and if it's going to take a considerable amount of time, provide a temporary workaround if possible.

Given that the hosting industry is 24/7, there is no excuse for software developers who don't have a security contact that is monitored around the clock. Unfortunately, we have encountered several companies who only maintain a security contact during normal business hours and we shudder to imagine what damage could be done if a zero day exploit were to be released on a weekend.

The last statistic that we feel is important to talk about are the four most popular control panels in the hosting industry. When it came to who acknowledged our security vulnerabilities the fastest, almost all of them did so within 48 hours. It's worth noting that cPanel technically acknowledges all security vulnerabilities using an auto responder, but we measured when they manually got back to us to confirm our vulnerability not when they received it.



As you'll see with the chart above, the numbers are a bit all over the place but InterWorx was the fastest when it came to releasing security patches based on our reported vulnerabilities. On average, it took them 7 days from first point of contact to releasing a security patch which is very respectable. Along with their 24 hour average to acknowledge our security reports, we feel that InterWorx were the most effective of all the control panels when it came to handling security.

Parallels Plesk was a bit slower, however, it's worth noting that we have a special arrangement with them and there were a few vulnerabilities that took considerably longer that we were not able to get an exact time frame on. Realistically, the 9 days from point of contact to releasing a security patch would have been higher had we been given the exact dates from them.

cPanel came in third with a 36 day average to address security vulnerabilities. That number is high due to their monthly release schedule which we absolutely do not agree with. As mentioned earlier in this review, we don't feel that security patches should ever be done on a specific release schedule. We are hoping that in the future cPanel will forgo the monthly release schedule and consider something more efficient.

DirectAdmin came in dead last with a 64 day average when it came to addressing our security vulnerabilities from first point of contact. There were a batch of security vulnerabilities present in their backup system that took a lengthy period of time to resolve as it required an extensive rewrite of the software. (The vulnerabilities were pretty serious and no doubt did require a long period of time to resolve, but we do believe that it could have been done a bit faster.)

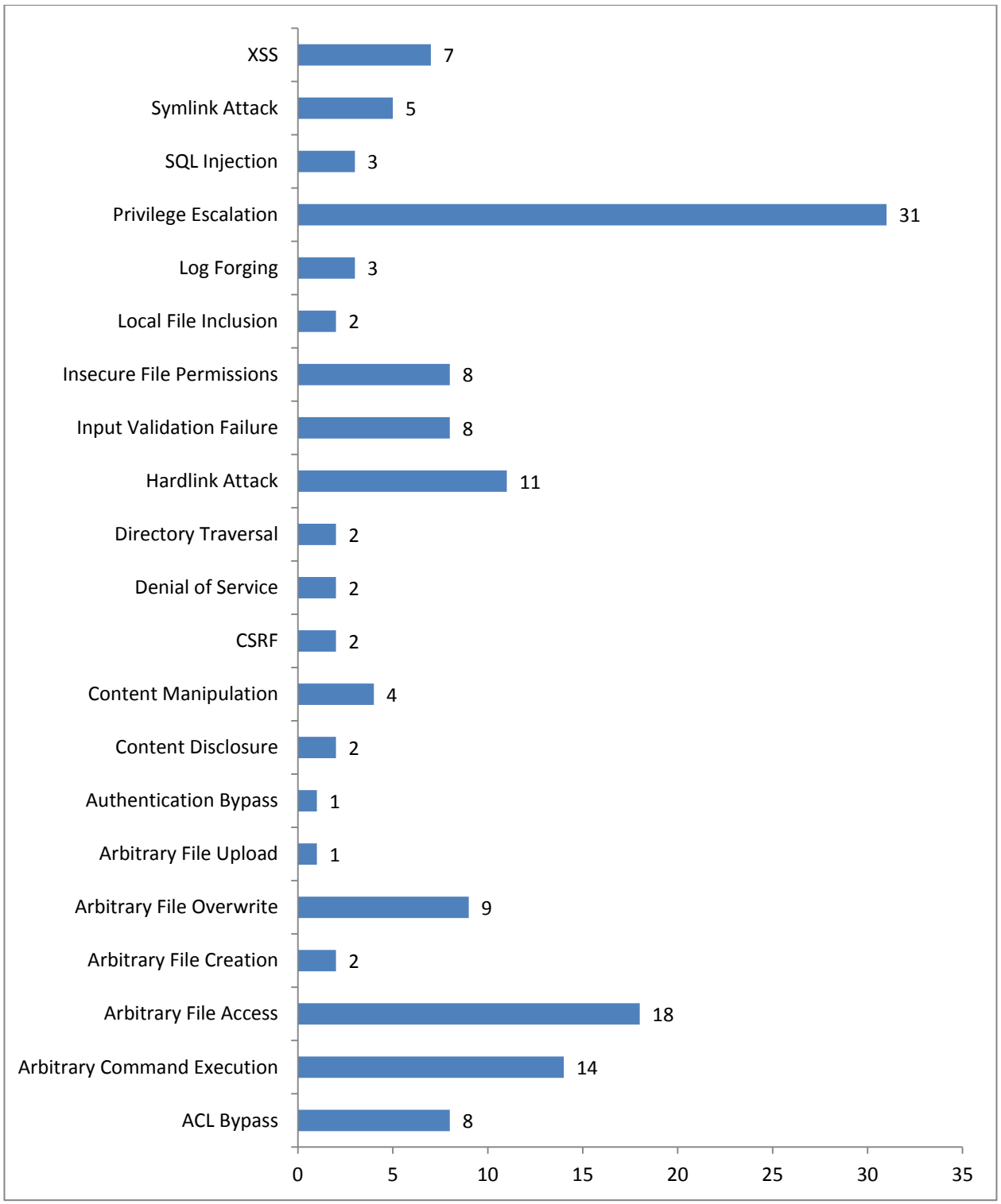
Vulnerabilities

During our first year we have encountered 21 different types of security vulnerabilities within various hosting applications. Technically, that number should be higher but we are only counting our public security advisories and not anything found during the course of paid audits and other private arrangements that now make up the bulk of our security research work.

The most common vulnerability that we managed to exploit was privilege escalations where we were able to obtain root or admin level access. We consider privilege escalations to be the most serious of security vulnerabilities simply due to the amount of access that an attacker can gain. In most cases, our privilege escalations gave us interactive access as if it were legitimately obtained making it difficult for an average systems administrator to detect.

The second most common vulnerability that we encountered was arbitrary file accesses that allowed us to view sensitive files that our level of access should not permit. For example, in some cases we were able to send a carefully crafted request to an application and it would return the contents of /etc/shadow or a config file showing various database credentials.

Other common vulnerabilities would be arbitrary command executions and hardlink / symlink race condition type attacks. Arbitrary command executions tend to be very dangerous as often times it's being done as a root level process, thus allowing the attacker to effectively take control of the server. Most of the hardlink / symlink race conditions we encountered also allowed us to gain root or admin level access.



A significant amount of vulnerabilities that we discover have one thing in common and that is the software developer has failed to sanitize the input and made the huge mistake of trusting the data being offered by the user.

The most important advice that we give software developers is to always assume that the user and even other “trusted” employees has malicious intent. Every GET and POST request needs to

be sanitized and checked against ACL's to ensure that permissions are being followed so that the request cannot exceed their authority.

As for when it comes to applications that perform file operations on a server, we always tell the software developer to drop permissions to the user when possible. Any time a root or admin level process writes to a user accessible directory, the risk for race conditions and other types of vulnerabilities is extremely high.

So many of our vulnerabilities would have been prevented had the software developer stripped special characters from every field where they are not needed, properly escaped SQL queries or used prepared statements, and dropped permissions to the user when performing file operations under their home directories.

Acceptance

For the most part, when we report a security vulnerability to a software developer they are appreciative of our efforts and understand what we are trying to accomplish. There is usually a mutual understanding that the information we have presented to them is important for the security of their product and needs to be handled as such.

We lay out clear guidelines to software developers who have never worked with us. There needs to be an understanding of the level of communication required, as in the company needs to acknowledge our report immediately and keep us updated on the status of a patch. There are also guidelines about what is acceptable in terms of responsible disclosure and if a software developer doesn't want to be upfront about the security patch with their customers, then we will do it for them in a very public way.

Unfortunately, not everyone has been receptive to our work…

In a high profile incident that occurred in 2013, a company that offers Alpha Master Reseller software thought it would be a good idea to suppress our work by threatening to sue us. That plan backfired after it ended up on Reddit and the thread on Web Hosting Talk currently shows over 130,000 views. (It's worth noting that the software in question is still vulnerable to several root exploits despite the developer saying otherwise.)

While we haven't been threatened since, we still encounter some resistance from time to time in the form of companies failing to communicate with us or refusing to take any action in a timely manner. In situations like that, we are usually forced to take public action on Web Hosting Talk in hopes of bringing attention to the situation and making the software developer re-think their plan.

Another issue that we occasionally come across are software developers who blatantly lie to their customers by telling them the security vulnerability isn't that bad. Then there are the times where the software developer will do everything right and get a patch out there as fast as possible, but call it a "routine" update as to not have to admit that their product had a security vulnerability.

If you are a software developer and someone informs you about a security vulnerability in your product, you need to communicate with them and you need to communicate with your customers.

Pride has to take a back seat because the reality is; almost all software is going to have security vulnerabilities at some point. Your customers will appreciate your responsiveness and you can make a huge deal out of telling them how you proudly resolved a security vulnerability in a short amount of time. Turn a negative into a positive!

Closing Thoughts

Our goal for the next twelve months is to continue working with software developers in the hosting industry to ensure that their software remains secure. Every single day, even on weekends, we are doing extensive security research work to help achieve that.

In addition to RACK911 Labs, we also run www.HostingSecList.com which is a mailing list to alert people of ongoing zero day threats and security advisories that affects the hosting industry. In the last eight months alone HostingSecList has grown to almost 2,000 unique subscribers with a lot of them being representatives of established hosting providers.

You may never have heard of us, but it's safe to assume that you have been directly affected by our security research work. No one really thinks about security until something bad has happened and as self-serving as it is, we know that RACK911 Labs has made a significant impact within the hosting industry for the better.

To everyone who supports our effort to help make the hosting industry secure, we thank you for your continued support and kind words.

~ Steven & Patrick ~

www.rack911.com