Robert Rowley - Making Security Easy

[00:00:00] Hello, and welcome to this presentation video prepared by Patchstack for the Page Builder Summit event. Talk title is Making security easy, and I am your host, Robert Rowley, Patchstack security advocate. I'm here to talk about security, topic everyone is concerned about and a lot of people think is pretty hard.

[00:01:10] They just wanna make it somebody else's problem. Or I should say making somebody else's responsibility. But in this talk, I will share with you my experience of being in charge of security for many hosting providers and share that it's actually pretty easy to do the security basics. And in turn, by doing something easy, you will stop sites from being compromised in the first place.

[00:01:32] But not only that, by doing security easily, you will be more valuable to your clients and customers. This talk will be brief, starting with an introduction about myself some more. Then I will share some scary stories about sites that have been hacked, but I'm gonna be sharing these stories because the root causes of those hacks could have been easily avoided.

[00:01:54] I'll share with you the security wins that you can use to protect your websites. Finally, I'll sum up things with a little talk about responsibility, guiding you through writing a shared responsibility model, and finally showing you how you can take responsibility for security for your website projects and how that can be pretty easy.

[00:02:15] So here I am, I've been doing security for a very long time, close to two decades, working first as a security architect for dreamhost. Then later the director of security for Pagely. I was there. I was present for some massive website attacks like Timthumb, and I wrote tools that did malware detection and cleanup.

[00:02:36] While I worked at these hosting providers, these tools ran on hundreds of thousands of websites. So many sites were cleaned that I remember one dreamhost customer creating a web comic thanking me, or really, I should say the automated tool, the happy dream host security bot, I think I called it. But people really appreciate the fact that when they're having a bad day and you come to clean up a hacked site for them, it turns around their day to be a great one.

[00:03:02] And it greatly shows the value of the services you provide. Myself I've identified many vulnerabilities and various software components in systems, including WordPress core plugins and themes. I've also seen and handled a lot of compromised websites. And I'll share with some of those stories with you now.

[00:03:23] First story I'll talk about is that mass website attack I briefly mentioned called Timthumb. It's a great story to start with. It's also pretty old. The Timthumb attacks started around 2011 and about a decade ago, and they continued through about 2013 or 2015. There was a lesson we can learn from. It was all about

[00:03:44] patching and the time it takes to patch, the Timthumbs are a great example of compromise being caused by an out-of-date insecure component, not getting patched. The Timthumb attacks were popular with attackers because with a single request, if successful, the attackers could upload a PHP back door, basically take over the whole website if they wanted.

[00:04:05] The reason they were so prolific for so long was because the slow time to patch or MTTP, a bit of security jargon called mean time to patch or average time to patch. The Timthumb library had such a slow, average time to patch because the developers who put Timthumb into their themes were including it as a library.

[00:04:30] And then those theme developers needed to update their version of the Timthumb library. Then push that update to all of their customers. And there were hundreds, if not thousands of themes that were needed to apply the patch. Then they had to push it to the hundreds of thousands of users and users who actually use their web, the themes on their websites.

[00:04:51] And that simply took very long to get distributed across the web. This again is the difficulty in patching and the distribution time for each patch led to a very long MTTP or mean time to patch, which means that simply sites were vulnerable much longer than they should have been giving the attackers a very wide window of time to attack websites and compromise them and take them over. The lesson we can learn from this is that the faster you patch, the better.

[00:05:21] A second lesson is that developers and dev designers supply the patch. But you need to trust your developers and designers to provide the patch. Insecure components are not only the only way that websites get compromised, though. Insecure passwords or compromised credentials lead to far more attacks on average.

[00:05:42] Let me share a little story of how I tracked the Brute Force Bots that hit WP login pages all over the web. Years ago, I set up a WordPress honeypot and a honeypot is a server that acts like a bait for attackers and I monitored the attack details specifically. I logged the username and password that these bots were attempting against my website.

[00:06:04] And this is what I learned. Bots mostly tried obvious passwords like password123, or funny enough, I noticed in the top 10, there were words like Naruto and Pokemon. The bots also were a little bit clever. They scraped details from the website and domain registration or other public details and used those in the passwords as well as attempted those names in the registration as the username.

[00:06:31] So don't use your domain name or street address number or birthday or any sort of public details as the only level for your password. Now eventually I registered about, for about think it was about one year or so of logging all of these attempted usernames and passwords, my honeypot WordPress site actually got its password Brute force,

[00:06:49] it got compromised. This was surprising, but not a surprise because I set it up with a username as being admin and the password that was password. Exactly p a s s w o r d. You may think that I probably should have had that site hacked a lot sooner. And he would be right. But I also had two factor authentication or two FA set up that's those little six digits that change every 30 seconds or so you have to look 'em up on your phone.

[00:07:18] Sometimes I turned that on for the admin user with the password of password and it shows that two factor authentication does a great job at protecting users, even when they have the worst possible password. it protected the site against Brute force attacks up until it eventually did get hacked in. So how did it get hacked again?

[00:07:38] One day what happened was there was a plug-in auto-update process that ran and I believe it updated the two FA plug-in that I was using. In the process of that auto-update, because I wasn't paying attention, it accidentally disabled the two FA functionality of that pump that plug-in, the site was pretty much hacked within an hour or so of the plugins 2FA functionality being disabled

[00:08:03] and my hosting provider's anti malware detection tools noticed the hack, took down the website and emailed me right away. The next day I ran through the investigation or incident response to find out exactly what happened. And I confirmed that the attack factor was a compromised password. And I noticed that the 2FA plugin was disabled.

[00:08:24] So in the end, I chose to shut down that project, cuz I had already learned so much, I'd gotten a great list of passwords that were being attempted. And I learned the dangers of auto-updating that sometimes happens. And I wanted to mention too, what a great job by the hosting provider. They were monitoring my website, detected a compromise and took immediate action.

[00:08:45] So I brought up those stories on purpose. Most hacked WordPress sites tend to be caused by one or two main issues, either password compromise or an insecure component. Password compromises of course are very dangerous. Let's face it. Brute force attacks are rampant. My WordPress honeypot prove that to be true for me.

[00:09:06] Password giving guessing bots are also very clever. So the solutions here are to enforce secure passwords. If you can introduce your users to a password manager, to really help them learn how to use secure passwords, very long, secure passwords, complex passwords on their websites. And if you need to walk your users through how to set up 2FA authentication on their websites, as long as the plugins don't get disabled, this thing works like gold to protect against brute force attacks.

[00:09:35] Next step was always updating. Please, always, definitely do the security updates. And this is going beyond turning on those auto-updates, which those can go wrong sometimes, but it's much worse to leave those plugins as never updating at all. So at least do auto-updates. If you can definitely, you must do security auto-updates or security updates.

[00:09:59] There's a bad little problem here though, is that sometimes plugins don't receive security patches. Some plugins get abandoned by their developers that leave sites with no options and no awareness of the risk of a vulnerability existing in a plugin until it is too late. This is why you need to have some sort of monitoring as well.

[00:10:19] Somebody like the Patchstack app or Patchstack database which provides a security operation center of sorts, which is a bird's eye view of all of your man managed websites with the Patchstack plugin installed that tell you what version of each plugin you're running and alert you. If there's a known vulnerability in one of those installed components, just lets you know that you need to patch right away.

[00:10:41] Finally, that last bit I mentioned where my hosting provider saved me, monitoring from malware or indicators of compromise is critical. It is, of course not your first line of defense. This is your safety net. Your first line of defense is still your passwords and updating your software.

[00:11:01] Don't rely on waiting for your sites to get hacked, to find out you needed to update a plugin, just pay attention to when the plugins have vulnerabilities. In a professional note for monitoring file systems as a systems administrator. This is ideally done using the operating system, not through the web application.

[00:11:20] Think about it. If a web application is compromised well, and the web application itself is scanning files for a signs of compromise. The malware running on the web application processes may see that you're scanning and simply hide itself from that scanner. Now, not all malware, of course, but I have seen proof of concepts that show malware skipping around files, avoiding detection from scanning that originated from the web application itself.

[00:11:44] So it's way better to do using the operating system. Not to mention the bigger issue of resource differences between a web application, trying to perform a file system scan versus tools that the operating system has available. The operating systems tools are designed for file system. Scanning web applications are not designed for server management.

[00:12:04] So simply use what's available in the operating system. The good news is there are already tools available. Open source tools like mal debt are written to scan file systems for signs of any sort of malware, especially web based malware. This is where the best solution is to always have a hosting provider who's capable of handling operating system and malware detection for you.

[00:12:29] Of course, you may have to pay extra because of course they're doing some more responsibility for you, but sometimes you may find that some hosting providers have baked this into the cost of their hosting service. Just to be more competitive, some charge only a few extra bucks a month. That's fine. You may need to reach out to your host and ask them questions about this.

[00:12:49] Let's move on the topic of taking responsibility. I would like to introduce you to a little bit of a information security corporate jargon called a shared responsibility model. We use this to make three easy security wins, and we will identify who is responsible for what, based on these easy security wins

[00:13:08] I just showed you.

[00:13:11] starting with what is a shared security model? Like I said, it's jargon. That is a framework that we use to write a policy, which outlines, which parties are responsible for what aspects of security for a project. Every project has a different shared security model. And I'll outline one for you here using a hypothetical case of a website for a brick and mortar shop where a business owner doesn't wanna focus on security or their website

[00:13:37] so they've hired an agency. Of course, the agency isn't writing all of the code, they're using custom secure, custom existing open source software components for their client's needs. So now we add another party, the open source software developer, and they have some responsibilities as well.

[00:13:57] Finally, we have the hosting provider, which is actually could be a cloud service provider or a managed host, or maybe an unmanaged host, but they're who the agency use, who uses to run the web servers, which one the run the web application. We'll be starting with three easy wins. Like I just talked about secure passwords, software updates and monitoring.

[00:14:16] I will add a few more things, but for now we need to start identifying some responsible parties starting with the first item of who is responsible for choosing good secure passwords. Any guesses? Is it the site owner, the developer, hosting provider or agency. Really, that's a trick question. This is everybody's responsibility.

[00:14:37] Everybody has a password and they need to choose good secure passwords that are not getting compromised. If it's a critical system or an admin account, I really hope that they're using two-factor authentication to make sure that password or that account never gets compromised. Let's move on to updates. Who handles the updates?

[00:14:57] Updates for different roles, mean different things for the hosting provider. This likely means that they're responsible for the operating system and the software it's running on the server. As long as you're paying for a managed server, you should expect that the management, it includes updating the software, like the operating system and everything underlying cuz vulnerabilities can exist in that.

[00:15:17] next. We have developers and designers. They provide updates in the form of patches for their code for their projects. Now it's a time to ask, should the site owner handle updates or maybe the agency could handle updates. Remember by taking on more responsibility, you show more value. And this is a conversation the agency would have with the site owner to ask them, do they have time to do a website maintenance themselves, or would they like to charge a nominal fee or, the site owner can pay the agency a nominal fee to continuously monitor and update their site for them.

[00:15:51] Now let's talk about monitoring. Again, this is the safety net. Like I mentioned before, it's something that's more efficiently done using the operating system, not via web application. This means it is for sure something the hosting provider could, and should do. I'm aware that some options to scan for malware using a web application exist, but it's really, that's only good if it's your last choice.

[00:16:16] If you have SSH access and you can install your own tools or the hosting provider, doesn't provide them for you. Try to install your own system for malade to scan your website's files on your own. Let's add some more bonus responsibilities, starting with backups. These can be done by the host or the agency, or maybe even the site owner themselves just like monitoring file systems.

[00:16:41] It's something that's best done at the operating system layer. Not something you wanna run inside a web application, but whatever works for you. Web based application backups can add some additional components which you'll need to monitor the security on for, and they may take up additional resources when they run their backup processes, but find out whatever works best for you.

[00:17:02] Ideally again, use something that's on the operating system layer and be sure to back up always backup before doing an update on the sites that you manage. Now let's talk about communicating security. These are some examples of things that agencies can look for in their developers, designers, and hosting providers.

[00:17:21] They wanna ensure that they do seriously take security. They do take security seriously. And I don't just say that after every breach. Is the developer designer honest about security patches. You can find this out really easily by looking at the project's change logs. A history of security patches is a good sign.

[00:17:41] All code is likely to have security bugs in it. What's important is that, when those bugs were patched, so you can avoid using the insecure versions. Furthermore, does the project have a vulnerability to disclose your policy? This is something you will see in large open source projects.

[00:18:00] Apache has one. WordPress has one. Indianhex has one, Joomla and Drupal have one. They all do. But the smaller add-on components you may have used for the client, those may be hit or miss. Now, this is not to be confused with a bug bounty program, but it is similar. A vulnerability disclosure policy can be simple in small.

[00:18:21] It could be just a page or a blurb on the project's website or in their readme files. All it needs to say is. Hey, here's how you report security issues in this project. That's really the basics. It tells security researchers what to do. More importantly for this model it shows us that the developers have a process in line to handle security bugs, which is yet another good sign.

[00:18:47] Another bonus is you can reach out to your hosting provider and start asking them some questions. Do they help out with incident response or would you have to handle that yourself? Or do they help out with your, what if your site was compromised? Would they be able to help you clean it up? Would they refer you to a third party?

[00:19:04] Would they maybe be able to help you review the logs or the cause of the hat, the attacks. This is really it fits better into the next line, which is the most important step: verifying. Because somebody needs to actually take the time to check with the various projects, parties involved in the project and make sure that they're aware of the responsibilities that they have.

[00:19:26] It will help us identify if there are any gaps where no one is taking care of that aspect of security. This is also a good time to start thinking about costs. Costs may be something like donating, so that free and open source software developers project that you're using that you're relying upon or

paying for an add-on from your hosting provider or any other third parties that do handle aspects of security that you do not have the time or resources.

[00:19:50] But remember, the agency is adding security, therefore adding more value to the project that is for the site owner. Site owner will likely wanna pay them more money, or at least they'll outcompete their competitors. Also remember that the agencies choose their hosting providers, choose the developers and designers and the projects they use on those websites.

[00:20:11] They can always choose the ones with better security. The agency handles security like a professional, is a sort of agency that is already thinking about writing their own security model, shared security model for their business. And here is what that shared security model may look like after you're done.

[00:20:29] In this case, the site owner is only responsible for making sure their passwords are good and secure. Then they pay an agency a little bit more money to handle their website updates for them and monitoring for vulnerable components. The agency in turn supports the developers, the designers, and the hosting providers who are responsible for all those other aspects of security.

[00:20:51] And they know that they will take care of those things for them. In the end, the whole project, which is just one website, has a strong security stack from the hosting provider to the site owner. And we know who's responsible for what aspects of security, because we've gotten it all written down. Now I've shared some easy steps with you about how to secure websites.

[00:21:14] And I've shared with you a model you can use to plan out the security responsibilities for a project. And I hope that I've shared some interesting ideas that have got you thinking. I now wanted to share a few stories about businesses who currently use patchstack and use it to create new revenue streams for their business.

[00:21:30] One, a very large hosting related business and the other, a very small agency, both patchstack customers. Starting with the hosting related business you probably heard of plus score Cpanel plus earlier this year announced a tool called the WP toolkit it's available in your plus and Cpanel hosting panel for all of users, the WP toolkit supplies, a whole suite of tools to manage WordPress websites straight from the hosting panel.

[00:21:56] This includes a set of security related tools, which are powered by the patchstack's vulnerability database. Patchstack provides them with intelligence, something that is hard to manage. And in turn, the WP toolkit uses their intelligence to help inform their users about insecure components installed on their WordPress websites.

[00:22:14] This information is available right in the Cpanel hosting panel. So there's no need to run external tools, no need to log into every site to check each one. It's all right there in one central dashboard. So far, they said it's been a great success. And I need to mention that WP toolkit does a lot more than security notifications.

[00:22:32] So go ahead. If you're a user of Cpanel or plus check it out. But Patchstack is not just for the big players in the market. We have plans for agencies and individual site owners. One such agency was Toronto signing digital marketing, who offers website care packages and security services to their customer.

[00:22:52] Stunning digital marketing uses the Patchstack app, a dashboard that we provide integrated with our plugin and the owner of Stunning digital marketing, Rob Karen told me that he appreciates how relatively little time it took him to get hundreds of websites moved off of their old security plugin that he was using and set up with the Patchstack app.

[00:23:11] Rob was delighted at having a security operation center. Remember like a bird's eye view of all of the components his customers ran on their website and he can manage it from one central place. And Rob is not new to the concepts of security or technology. Before running stunning digital marketing agency, he worked as a corporate tech administrator.

[00:23:34] He knows about security and he appreciates the fact that the Patchstack app saves him a lot of time and headache when it comes to spotting and addressing security issues for hundreds of clients. In turn, Stunning digital marketing clients, they trust Rob with their websites. They don't mind paying a little extra for the website care package.

[00:23:54] Because they see the value in the extra work and care that Rob puts in for their safety. Either of these stories could be you or your agency or business or your side hustle. And that thought is what I will end this presentation on. Thank you very much for your time for watching this video and thank you page builder summit for giving me us the time to talk about patch tech and some security concepts that anybody can benefit from.

[00:24:21] We've also included a giveaway for the attendees who are actually going to the Page Builder summit, then the page builder powerpack. So please go ahead and check that out if you received one. For the rest of you, have a great day.