CTRL+CLICK CAST

http://ctrlclickcast.com/episodes/securing-site-content

CTRL+CLICK CAST #079 - Securing Site Content with Rachel Andrew

CTRL+CLICK CAST is proud to provide transcripts for our audience members who prefer text-based content. However, our episodes are designed for an audio experience, which includes emotion and emphasis that don't always translate to our transcripts. Additionally, our transcripts are generated by human transcribers and may contain errors. If you require clarification, please listen to the audio.

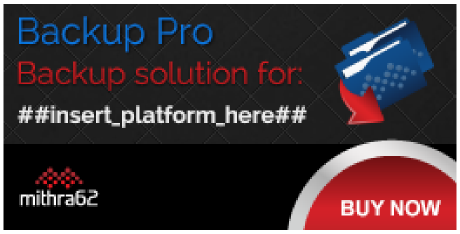
[Music]

Lea Alcantara: From Bright Umbrella, this is CTRL+CLICK CAST! We inspect the web for you! Today Rachel Andrew joins the show to talk about securing site content. I'm your host, Lea Alcantara, and I'm joined by my fab co-host:

Emily Lewis: Emily Lewis!

Lea Alcantara: This episode is brought to you by mithra62's Backup Pro, a complete backup solution for WordPress, ExpressionEngine 2 and 3, Craft, PrestaShop and concrete5. We use this one ourselves for some of our client sites. It's insanely customizable and includes automated backup integrity, eight built-in storage locations, console routing. Basically, Backup Pro was built to make disaster recovery as painless as possible. Just visit backup-pro.com to get started.

[Music ends]



Emily Lewis: Today we are excited to have Rachel Andrew on the show. Rachel is a front- and back-end web developer, author and speaker. She's also co-founder of the CMS Perch, a Google Developer Expert and an Invited Expert to the CSS Working Group. She writes about business and technology on her site at rachelandrew.co.uk. Welcome to the show, Rachel.

Rachel Andrew: Hi, it's great to be here.

Lea Alcantara: So Rachel, can you tell our listeners a bit more about yourself?

Rachel Andrew: Yeah, I'm, as you can hear, a Brit. [Laughs]

Emily Lewis: [Laughs]

Rachel Andrew: I'm currently sat in Bristol in the UK, although on Friday, I'll be flying off to Orlando for *An Event Apart*. So I do quite a lot of travel. I travel to conferences and speak mainly about CSS at the moment. Other than that, I am a runner so as I quite love running, I run wherever I go.

Emily Lewis: [Agrees]

Rachel Andrew: And so I've sort of enjoyed that. That's mainly how I keep fit. And in the distant past, I was a dancer. That was my sort of original background, I trained in dance.

Lea Alcantara: Oh.

Rachel Andrew: So…

Emily Lewis: What kind of dance?

Rachel Andrew: Ballet and contemporary originally.

Emily Lewis: Oh wow! Did you do like point ballet?

Rachel Andrew: Yes, yeah.

Emily Lewis: Wow! Wow! That's amazing to me.

Rachel Andrew: Yeah, so that's was, yeah, way, way back, and in fact, my daughter is now training to be a dancer.

Emily Lewis: Oh, wow!

Lea Alcantara: Oh, very cool.

Rachel Andrew: Haven't managed to put her off... [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Emily Lewis: Well, before we dive in today's topic, Rachel, it's been a while since we talked to your partner Drew [McLellan] about Perch CMS, and I'm sure some of our new listeners would love to hear how you got started building Perch.

Rachel Andrew: Yeah, sure. So really it came out as something that we needed as an agency. At the time, we were a development agency building sites for clients, and most of our clients were design agencies, so they would have some projects that they couldn't do the development side of and so we'd ended up doing that. Most of those are fairly big CMS sort of projects. And then the same agencies often have kind of those little projects or the sort of project where they'd build a static site and then at the last minute, the client had turned around said, "Oh, I'd like to go ahead and edit the text on the home page. Can you just do that for me?"

Lea Alcantara: [Agrees]

Rachel Andrew: And so that was really the initial idea for Perch, it was very much just that kind of drop-in CMS for those kinds of times when you don't need a really big thing, but you just need something that's very quick to implement, but it also cares about content and it allows you to structure your content in those scenarios as well as, you know, like a big sort of WYSIWYG pit and not being actually to control the stuff.

Emily Lewis: I think the other thing I like about Perch, which is why we like all of the CMSs we choose to use is that it also gives you control over your front end.

Rachel Andrew: [Agrees]

Emily Lewis: So you're not stuck with stuff you don't want in your code.

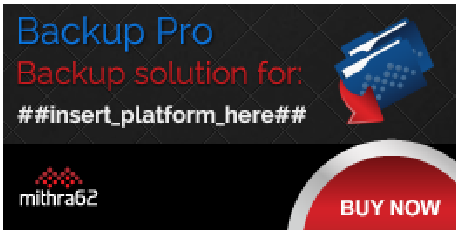
Rachel Andrew: Yeah, and that was at the time was really unusual. Pretty much everything else out there, it was just big textarea with the WYSIWYG and that was it. There wasn't really…

Emily Lewis: [Agrees]

Rachel Andrew: I mean, there's now a few more options for doing kind of structured content, and people have realized that this is really important, but at the time, there was pretty much nothing, which is why also the big framework that we were using for the bigger projects took this structured content approach and we wanted to have something small that have the same kind of idea behind it.

Emily Lewis: Well, we're going to talk about Perch a little bit more. In this episode, we're all going to sort of share our own experiences with securing site content. Let's first start with what securing content means. Rachel, how do you define it?

Rachel Andrew: Well, it's kind of things. I think one of the things that we're thinking about is actually securing content or perhaps having some sort of members' only content, content that's kind of behind a pay wall, which a lot of the new sites are starting to do. You have to pay a membership and then you can get to see some content, so that's kind of a way of securing content.



And there’s also things, if you’re selling, for instance, software, like we do at Perch or have an e-book, for instance, you might want to secure that somehow so that people have to pay before they get access to it and they can download it. So that’s kind of one set of security things about actually giving people access to stuff based on them paying for it.

Emily Lewis: [Agrees]

Rachel Andrew: But kind of tied up with that and related as well is this issue of security on the web, and people moving to HTTPS and TLS in order to have a secure connection between the person accessing the site and your site itself, and that’s something which I think has become quite a hot topic recently for various reasons, and I think when we’re discussing this episode, we thought that it would be good to cover too.

Lea Alcantara: Yeah, I definitely agree with Rachel’s explanation. It sounds like security at the end of the day is making sure that content is only accessed by the people you wanted to be accessed by. When we had an episode with Matthew Weinberg on security audits, his general explanation was that “security is this idea that nobody should be getting access to your data or have the ability to make changes to your data without your permission.”

Rachel Andrew: [Agrees]

Lea Alcantara: So I think that HTTPS and the TLS stuff is the other side of that.

Emily Lewis: [Agrees]

Lea Alcantara: That's kind of like hinting on that, but also with the content, yeah, making sure that people or bots don't get access to the content that only certain people should.

Emily Lewis: [Agrees]

Rachel Andrew: Yeah. I mean, that might be as simple as you've got some client projects that you would like the clients to be able to log in and have a look at, but you don't want them to genuinely be out there on the internet with all these kinds of information.

Lea Alcantara: Right.

Rachel Andrew: Or it could be that you're developing a site for, say, a school and this information which should only be for parents to look at, so you wanted to make sure that nobody else can get in and look at – I don't know – report cards of the children or information about what they're doing and things, which might be quite sensitive if it got out there.

Emily Lewis: Right.

Rachel Andrew: So there are all kinds of reasons, not just e-commerce-type reasons why you might want to tie down stuff.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

Emily Lewis: I have to claim a little ignorance here. I'm familiar with HTTPS and setting up SSL certificates, but I've not heard the acronym TLS. What is that, Rachel?

Rachel Andrew: TLS, that's a very good question. So TLS stands for Transport Layer Security, so that it's kind of the next generation of SSL.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

Rachel Andrew: But genuinely people talk about this as SSL. It's cryptographic protocol basically, so when people talk about HTTPS, they're actually talking about TLS generally.

Emily Lewis: Oh, okay.

Lea Alcantara: Right.

Rachel Andrew: Or if someone talks about like an SSL certificate, then generally these days, it's a TLS certificate they're talking about.

Emily Lewis: Oh, so it's like the type of certificate and it's tied to the configuration on the server.

Rachel Andrew: Yes. So yes, it's basically to do with HTTPS certificates, and it's really just the name of a protocol that you use to ensure that security.

Lea Alcantara: Right.

Emily Lewis: Well, let's talk a little bit more about these components. Rachel, do you feel like the core requirements for securing content is this certificate and having your pages render in HTTPS?

Rachel Andrew: That's kind of the starting point. If you're going to have a secure web on your site at all or you're going to be selling things, then you want to make sure that people are connected in a secure way because if they're not, there's obviously the possibility that someone who shouldn't be able to is going to be able to intercept that connection and, for instance, get the password so they can log in. It's a bit like why you wouldn't want to just use regular FTP to transfer things to your site because by doing that you're transmitting their password in plaintext and anyone could be sniffing that and getting that information and be able to log into your server.

Emily Lewis: [Agrees]

Rachel Andrew: And it's the same with connecting with the browser. If you're transmitting content in plaintext, then something on the server or running on that computer could actually be accessing that to someone out there on Wi-Fi if you're on a public network and potentially be accessing that data out as it's being transmitted.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

Rachel Andrew: So if it's important that something is secure on your site, you want to make sure that people's connection to your site is also secure.

Emily Lewis: And in terms of getting HTTP running SSL or a TLS certificate on your server, in my experience, it's relatively straightforward. Admittedly though, I've never done a really advanced implementation of securing content.

Rachel Andrew: So it should be straightforward. If you're building a new site, I mean, anyone building a new site now should put that site straight on to HTTPS. Don't worry about putting it on HTTP.

Emily Lewis: Oh.

Rachel Andrew: Just put it on HTTPS. You won't have any problems. The issue comes when you've got an old site that's got loads of old content that perhaps links to images somewhere else that are on HTTP.

Lea Alcantara: Right.

Rachel Andrew: Perhaps you're using a CDN or you're using some third-party sort of like JavaScript badge thing that you drop onto the site, and that's not HTTPS, because the minute you've got mixed content, you've got some content loading via HTTPS and some not, you start getting these areas in the browser saying, "You've got mixed content here," which to someone visiting your site, even though it might not be a security issue, they might think, “Oh, I’m getting a security warning about this website,” and be worried about it.

Emily Lewis: Right.

***Timestamp**: 00:09:52*

Rachel Andrew: So it can be reasonably tricky to take an old site that’s got a lot of content and move it over because you’ve got to sort out all of these things that might cause a warning. For instance, on my own site when I went to HTTPS, I had to remove the Lanyrd JavaScript badge because they didn’t have a TSM (Trusted Service Manager) point for that and so it was throwing these mixed content warnings. So sometimes you’re going to find that something you’re using doesn’t work anymore because they don’t have HTTPS and you do.

Emily Lewis: [Agrees]

Rachel Andrew: So that sort of stuff can be a bit of a problem, and just digging back, you’ve got a lot of blog posts and articles and things, going back and finding which ones have got things in that you might need to change the links or whatever, that could be a bit of a pain.

Lea Alcantara: Yeah, it’s interesting. Our client actually ran into this issue very recently where we did a bunch of caching for them, and before everything was HTTP, and without talking to us or whatever because they’re just moving on and the site was launched and all that fun stuff, they decided to serve their site on HTTPS. But because of the way we cached certain things like their navigation and when it's generated, the CMS still had HTTP on their domain.

Rachel Andrew: Right, yeah.

Lea Alcantara: And so the next time I visited their site, certain images like banner images were completely 404-ing or just not even showing up.

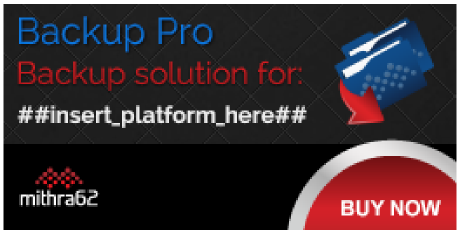
Rachel Andrew: [Agrees]

Lea Alcantara: And everyone was like, "Well, what's going on here? Because everything else seems to be working, why are the images not working?" And it took a while because you're looking at the page and it looks like everything is working, and then I realized, "Oh, the domain now says HTTPS, and all the links are pointing still to HTTP as well as the graphics and that's why it wasn't showing." So, definitely when you are moving to a new protocol, you have to think about like all the link relationships within our site.

Emily Lewis: And caching, like you said.

Lea Alcantara: Yeah, yeah, yeah.

Rachel Andrew: Yeah, I mean, yeah. So we have this problem with our sites because we used Varnish a lot, which is like a cache, and Varnish doesn't support HTTPS, so if you move to SSL, you kind of have really…



Emily Lewis: Really?

Lea Alcantara: Right.

Rachel Andrew: You have to do something to kind of move around that. First, we use Ngnix as a sort of proxy basically to sort of proxy those requests, which works quite well.

Emily Lewis: [Agrees]

Rachel Andrew: But that's the whole of the thing you're going to have to learn about in the setup if you're just doing this yourself.

Emily Lewis: [Agrees]

Rachel Andrew: And I think, increasingly, a host that's going to offer good support for doing this and they're going to help people. There are probably fewer people that are building small sites for clients and things that have the sort of setup that we have.

Emily Lewis: [Agrees]

Rachel Andrew: And so if your host knows what to do and can help you with it, then that's great, and really, if your host isn't helpful with this stuff and you're not someone who is managing all of your own servers, you're probably ought to be looking at moving to a host that is good at this stuff at this point.

Emily Lewis: Agreed.

Lea Alcantara: [Agrees]

Rachel Andrew: Because it's becoming so important.

Emily Lewis: Yeah, I much prefer managed hosting. I don't have this level of technical knowledge, so it's nice having somebody who does. [Laughs]

Rachel Andrew: Yeah. I mean, this is something to kind of figure out. If it's not your thing, then it's better to find somebody else whose thing it is and let them worry about it and help you with that thing.

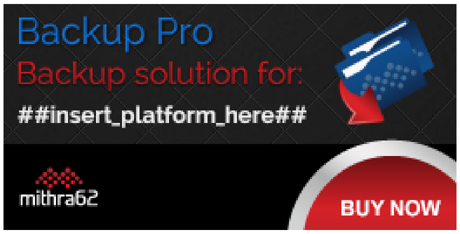
Emily Lewis: Absolutely.

Lea Alcantara: Absolutely.

Emily Lewis: So with regard to this sort of requirements in the basic understanding, is SSL required to run an HTTPS connection or you could do HTTPS without an SSL certificate?

Rachel Andrew: No, I mean, that's basically how you get HTTPS is you have an SSL certificate.

Emily Lewis: [Agrees]



Rachel Andrew: I mean, you can do it locally. You can do things like self-signed certificate, which basically lets users test this stuff, but you really need to a proper certificate for your live server because they kind of have two parts to them. One is that they enable you to have a secure connection to the server, between the browser and the server, but the other thing they do is they verify kind of who you are, particularly the sort of the more expensive certificates you can get.

You can get certificates that kind of quite a lot of checking that the company is who they say they are, which for sort of e-commerce sites and things, that starts to become an issue. There's kind of trust level there saying that, "Yes, this site is exactly who they say they are. I am connected to PayPal and not some scam sites." You know?

Emily Lewis: [Agrees]

Rachel Andrew: So there's kind of two levels to be at, so you need to have that certificate or some certificate in order to be able to have a secure connection, and then the certificate that go onto a public site also have this sort of side effect of also kind of verifying you and adding to the trust of your site.

Emily Lewis: Now, you mentioned it a little bit earlier in this episode, but I also saw a piece you wrote for Smashing Magazine about HTTPS everywhere. You said earlier that someone, if they were starting a brand new site, "Go ahead and set it up with HTTPS." So, why is there a trend for HTTPS encryption by default?

Rachel Andrew: There's a really big push for this to basically to happen. A lot of this is coming from the browsers and, particularly from Google, that really everything should be secure. We should be having a secure web by default, and so that's kind of been encouraged in all kinds of ways. Chrome is starting to actually sort of label HTTP connections as non-secure.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

Rachel Andrew: In January of next year, if you've got an HTTP page with a password box on it, for instance, like a login form, they're actually mark it specifically as not secure so that people know that when they add their password to that page, it is being transmitted in clear text.

Lea Alcantara: Interesting.

Emily Lewis: [Agrees]

Rachel Andrew: So that's going to start… I think, I mean, certainly the stronger the CMS... I think we're going to start seeing problems to that because obviously people download Perch and they install on their own hosting, and then their clients log in to the admin to edit their pages. Now, if people haven't hosted on HTTPS, the client is going to come to log in to their CMS in January in Chrome and they're going to see on the page, "Not secure."

Emily Lewis: Right.

Rachel Andrew: So that means that those of us who have content management systems are going to have to do a level of educating to people so they know that really they should be deploying on HTTPS.

Lea Alcantara: Oh man.

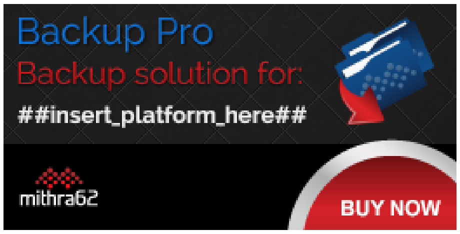
Rachel Andrew: Otherwise, they're going to have these questions from their client saying, "Well, why is this not secure?" There's that. There are also some of the newer APIs, browser APIs, things with geolocation and stuff, going to be made HTTPS only kind of. Let's say, if you're asking someone to share their location, you should be asking them to do that securely, not over HTTP.

Emily Lewis: Right.

Rachel Andrew: So you could actually lose access to certain APIs if you don't have a site that says HTTPS.

Lea Alcantara: Wow.

Rachel Andrew: And also just it's going to stop being a ranking signal, and I think there are all sorts of reasons how, and certainly Google, but other browsers too, are starting to say, "By default, you should be serving these sites in this way." So yeah, I think it's time. [Laughs] You know, if you're building a new thing, just put it on HTTPS at the start because you'll bypass a lot of the problems if you start out with HTTPS anyway.



Emily Lewis: What if you need to transition a site to HTTPS, do you have a recommended approach?

Rachel Andrew: It really depends on the site, I think, and the sort of content you've got. I think it's toughest for large sites, particularly if a lot of their content isn't content managed where that's got lots and lots of static pages, and a lot of sites out there are like that, because going through all of that stuff is going to be hard. But at least if you can get your new stuff and your main pages secure, you could always then work through the others, but certainly, anything where you've got login forms, anything where people are transmitting data, that, at the very least, should be transitioned sooner rather than later.

Emily Lewis: [Agrees]

Rachel Andrew: If you just got a bunch of static content that was less important anyway, then you can probably start to try and do that as you go along.

Emily Lewis: All right, let's get into some examples of project situations where we've each applies some of this securing content for clients. Lea, why don't you start with one of our non-profit clients, their member association, and they've been your clients since you and I even joined forces.

Lea Alcantara: Yeah.

Emily Lewis: Can you describe what you built for them and how you secured it?

Lea Alcantara: Sure. So primarily the security situation for them is that they have member-specific contents, so that they would have to pay to view this specific type of content, and the main way we were able to provide that was installing something like ExpressionEngine where their member once they get their membership paid and all that fun stuff, they have a specific login to themselves, and once they log in, then they're able to search the member directory, find specific files that are member specific, et cetera and so forth.

Emily Lewis: And so just to interrupt real quick, the payment isn't happening through this, it's just the login.

Lea Alcantara: Yeah, it's just through the login. The payment is through PayPal. They have a very, very simple membership thing. They have basically a PayPal widget where their admin just plugs in what the membership cost is, then once that's all paid for, then their administrative person manually adds them to a member.

Emily Lewis: To EE.

Lea Alcantara: Yeah, to EE, yeah, and then once that's done, then that's essentially it. They're in the system and then I set up a time where their membership gets expired and then they get an email, that kind of stuff, but otherwise, it's a super, super simple setup.

Emily Lewis: [Agrees]

Lea Alcantara: Nothing is like really majorly automated. But once they have the login, then they have access to all these pieces of content. Now, sometimes, and I know I made this mistake, people stop at that as in like, "Okay, now, they've got this content where I wrapped the template saying like, 'You have to be in this particular member group to view this content,' and that's that." But you might have noticed that I mentioned, "Oh, they also need access to specific files. Well, okay, so they might not have access to the page that lists the links, but did you remember to secure the folder where those files and PDFs and stuff are?"

***Timestamp**: 00:20:07*

Emily Lewis: [Agrees]

Lea Alcantara: So the other thing that I had to make sure after launch, and I made the mistake that I didn't do this until afterwards [laughs], but make sure that those specific folders are not spider-able.

Emily Lewis: [Agrees]

Lea Alcantara: Make sure that once someone does have access to those pages, to those links, add "Disallow to robot.txt" through the specific items and then make sure that you've got like no-archive, no-index, and no-follow in all the meta tags as well.

Emily Lewis: [Agrees]

Lea Alcantara: So there were a lot of people who were logging into certain pages and they're like, "Oh, but I found it on Google. Well, that sucks."

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Emily Lewis: [Agrees]

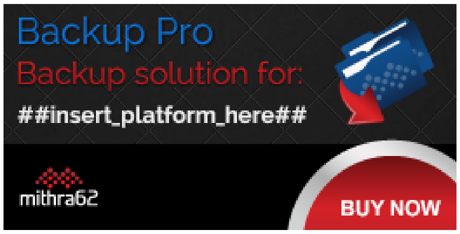
Lea Alcantara: Yeah. But simply making sure that Google doesn't even spider it. So it wasn't necessarily there's a nefarious thing, because sometimes when we talk about security, that we're thinking there's like a nefarious hacker or certain situations, but Google's entire MO is to spider a content.

Emily Lewis: [Agrees]

Lea Alcantara: And if you give them access to that content, they're going to spider that.

Emily Lewis: And make it public, right?

Lea Alcantara: Yeah, exactly, so make sure that you deal with that. If you also have downloads, ExpressionEngine has this add-on called Link Vault, so when we're talking about security, we're sometimes talking about access to things that only certain people want, but once they have access to



that, maybe you still don't want them to know where those files exist. Like for example, you don't want them to know that it's the download folder or something like that.

Emily Lewis: Oh, okay.

Rachel Andrew: [Agrees]

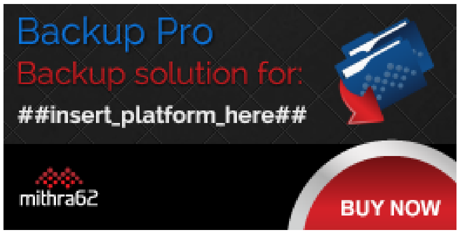
Emily Lewis:

Lea Alcantara: So then you want to have something called obfuscation where you kind of hide where the link is and you've got some gobbledygook in the URL structure, so that the only way they can download that file is to click the actual link so they can't share it with anyone. So even if they share the gobbledygook, it will check if you're logged in and it won't give you that download.

Emily Lewis: [Agrees]

Lea Alcantara: If you don't have obfuscation, then they do know that it's on a download folder and then they can deal with that, and most people who aren't like hyper-technical won't have too many issues with that, but this does give you like that added level of security where if you really want to make sure these links aren't passed around, you need to make sure that the links are extremely difficult or not even accurate because it's obfuscated

Emily Lewis: [Agrees]



Rachel Andrew: Yeah, I mean, really the way I just do it is to keep this out of your site root.

Emily Lewis: Right.

Lea Alcantara: Yeah.

Rachel Andrew: If anything is in your site root, there's a possibility that someone can get to it, so generally what we do is we put things a level up. They're actually not accessible via the web at all unless they're served by whatever system you're using, and therefore you can do a check at that point, "Oh, this person have downloaded this file."

Lea Alcantara: Right.

Rachel Andrew: And you can also do things like log the number of downloads and so on. If you wanted to, if you want to check that, yeah, someone isn't sharing something around or what have you.

Emily Lewis: I love that suggestion, and also it just reminds me to point out, especially to maybe someone who might be new to CMS development, you also want to make sure if you're using a CMS, don't install that on the public root, but install it above it. That's going to give you some extra security for your CMS.

Lea Alcantara: Yeah, absolutely, and the other thing that was useful for our particular client is securing content via .htaccess in their IP address, and this is mostly relevant during development, especially if you're working on your client's development server and their specific items and they've got some certain protocols where only certain people have access. So our client Nichols College, our college client, and they manage all their own machines, so when we were developing on their dev server, they actually gave us specific permission, like our special machines and our specific IPs' permission to access any of their CMS, FTP and admin items. So if they didn't have our IPs, Emily and I wouldn't be able to even get into the CMS regardless if we had a login.

Emily Lewis: So Rachel, I recently, well, maybe like a month ago, I listened to the Perch Podcast #54, and you talked about selling content behind a paywall with Perch Shop, which is your e-commerce option.

Rachel Andrew: [Agrees]

Emily Lewis: Can you talk a little bit first though about how Perch, the CMS, can support secure content, and then a little bit more about Perch Shop?

Rachel Andrew: Yeah, sure. So we've had for a long time, we've got this sort of Members App, which is if you've got a Perch license, it's a free download. All of our apps and add-ons are free for Perch, so if you've got a Perch license, you can use any of those you want, and so Members is one of them, and so Members let you do sort of a membership site essentially, so you've got people who log in and you've got the normal sort of members sort of fair functionality like editing of account and password resets and all sorts of stuff, but then once you've got Members, you can create tags essentially which say, "Well, can this member access this page?"

And so you can give a member a tag and maybe it could just be that you have a tag, which is secure content, for instance, and if it got that tag, then they can view the content, and that might be entire pages, it might be parts of pages or it could be downloadable files or anything else that you want them to have access to. So that's something that we already had, we've had members for quite a while, and so Shop really just works with that. One of the things that we wanted to do with Shop is to make it really great for selling secure content, allowing people to sell e-books, that sort of stuff.

Emily Lewis: [Agrees]

Rachel Andrew: Because that's why I could use it, particularly Perch, the smaller product, people will, let's say, want to sell an e-book off their site, that seemed like a reasonable thing you want to do. So what we do with that is the Members App and Shop work together, so your customers in Shop become members.

Emily Lewis: Oh.

Rachel Andrew: And one of the things you can do is when you create a product in Perch Shop, you can say, "When someone buys this product, assign this tag to them." So once you get a successful payment, it can assign them a tag and then that could give an access to some content, so it's a really simple way of getting up and running selling either content behind the paywall or the downloadable product or whatever.

Emily Lewis: Excellent, and so in terms of not even using e-commerce, the Perch Members App, is that what gives you the ability to I guess have like a login?

Rachel Andrew: Yes, yes.

Emily Lewis: Okay.

Rachel Andrew: That lets you do all your sort of members' things, and yeah, so the tags that we assign, they can persist forever, or you can set them to expire so like if someone is buying access for a year or for instance, if you're just using Members, say, for a school, you might set the tag "parent" to expire at the end of the year, for example, or whatever.

Emily Lewis: [Agrees]

Rachel Andrew: And then you could reapply them to people who should still have access, so the Members App is kind of a separate thing, but it works really nicely with Shop to do the kind of shop in a way you want someone to be able to pay, and then then need to be logged in.

Emily Lewis: Right.

Rachel Andrew: And I've done that with my own project. I have a CSS workshop where it's sort of like a layout, an online layout workshop, which is all secure content, and people can buy and then get in and be able to access the things straightaway.

Emily Lewis: And I just wanted to reiterate just to make sure we're covering our bases for listeners is that you need to have everything running on HTTPS with the SSL certificate for this to be truly secure and working correctly, right?

Rachel Andrew: Yes, and also if you're using things like Stripe or what have you to do your payment, as it's one of the options we support, you want to be running that on a secure page on your site when it launches the Stripe JavaScript and things as well as the security shoot. It shows that you care about that stuff.

Emily Lewis: [Agrees]

Rachel Andrew: And you want your customers to feel confident that everything is going to be fine, if they give you their card details, and so yeah, I mean, it's just the right thing to do, particularly if you've got people logging into content.

Emily Lewis: Absolutely, it builds trust.

Rachel Andrew: [Agrees]

Emily Lewis: Just continuing on the e-commerce theme, I just wanted to mention a few points about ExpressionEngine e-commerce. So I've used both CartThrob and Expresso Store very much like Perch, and really any e-commerce needs SSL and HTTPS, simply they will comply with merchant account requirements, but beyond those requirements, which as we've discussed, it's up to the dev to set up.

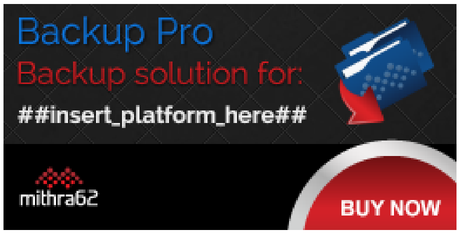
CartThrob and Store are really straightforward in terms of supporting secure transactions because they both come with pre-made stores, templates, that you can then build off of and customize, and then both also offer documentation, but I do think CartThrob's documentation is a little deeper and probably a little more user friendly than I feel like Store is.

But what I thought was great about both of them, and I think this comes into play and it sounds like Perch is the same way, is that CMS does and add-ons give you a way to offer some of this more robust functionality without you having to do custom coding, and that just is a really nice thing to be able to do if this is not your wheelhouse to roll this stuff yourself.

So yeah, I like CartThrob and Expresso Store from the simple standpoint of you can really get started with securing products, securing membership information, securing the credit card transaction with very little understanding of the underlying things because it really just gives you step by step of what you need to do, including with CartThrob, they have detailed documentation on just your core PCI compliance, which I think is really useful, and that's related to credit card processing.

Rachel Andrew: Yeah, I mean, the problem with e-commerce is there are so many potential moving parts, and there are so much into slanting that not only the person building the site needs to have it all, but also the end client because like they're going to have to, for instance, set up their merchant account or at least get that account with Stripe. They're going to have to complete the forms for PCI DSS. There's a whole bunch of stuff that people have to know once you head that down that road.

***Timestamp**: 00:30:15*



Emily Lewis: [Agrees]

Rachel Andrew: And as a developer of an e-commerce platform, there are so much that you have to put in place because when someone says, “I want to sell a product,” there’s just a whole bunch of stuff that then rolls from that. You have things like discounts. You have shipping and all the different kind of shipping options.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

Rachel Andrew: You have all the tax you have to deal with. [Laughs]

Lea Alcantara: Right.

Rachel Andrew: There are so many things, and when you’re actually assessing platforms to use, you have to look whether the platform that you’re going to use, is that going to support all the stuff that you need and that your client needs?

Emily Lewis: [Agrees]

Rachel Andrew: Because it really is something where you need to sit down and actually spec out probably exactly what the requirements are because it’s the sort of area where very quickly you go down this kind of, “Oh, but we have to be able to do a buy-two-get-one-free-type offers,” and then your platform doesn’t allow you to do that.

Lea Alcantara: [Laughs]

Rachel Andrew: That might be a deal breaker, yet it might be of no consequence at all to the next client.

Emily Lewis: [Agrees]

Lea Alcantara: Right.

Rachel Andrew: So there’s just so much stuff there.

Emily Lewis: Has that been a challenge for you as the creator of Perch to make sure that your documentation, the online information, kind of Git provides developers that support as they’re assessing, but then also as they’re building?

Rachel Andrew: Yeah, absolutely. I mean, as I say, there are literally hundreds of features that we could add to Shop, so we try and make sure that we’re very clear as to which features we currently have and which features we don’t have and which things we’re planning to add.

Lea Alcantara: [Agrees]

Rachel Andrew: And we've actually made our road map public for Shop so that we have it where it helps us to see what people are really interested in because we often get feature requests and that kind of just people making a shopping list of possible features they might like as opposed to if I don't have this is a deal breaker.

Emily Lewis: [Agrees]

Rachel Andrew: And having the road map public and that people vote on the features just helps us see like which of those features are more important, because no solution is going to offer everything, so that thing is very important, any solution is clear, which things they're doing and which they have decided not to do, because that lets people make good decisions as to what they're going to use.

Emily Lewis: [Agrees]

Lea Alcantara: So I'm a little bit curious. I just want to take a step back because we've been kind of driving home that if you can use HTTPS, especially if it's a new site, just do it and possibly start considering transferring your current client sites into HTTPS, but how would you communicate that to a client, especially when we're talking about, "Yeah, just get an SSL certificate."

That's an added cost. It's inexpensive, relatively speaking, but then there are also different types of SSL certificate levels, I believe. How would you communicate that to a layman, which is generally your client?

Rachel Andrew: Well, I think that you can really work, particularly now that Google is kind of saying, “Well, it’s really important that you go to SSL. If you don’t, then we’re going to start announcing in the browser that your site isn’t secure.”

Emily Lewis: [Agrees]

Rachel Andrew: So that’s a pretty good reason for your client.

Emily Lewis: [Laughs]

Lea Alcantara: [Agrees]

Emily Lewis: Right.

Rachel Andrew: Clients tend to go by SEO, and they care about Google. [Laughs]

Lea Alcantara: [Laughs]

Rachel Andrew: So while I think some people feel maybe that Google is trying to sort of push this, sort of hammer this home, because they can do because of their position. Actually, having a secure web is quite a good thing, and Google are in a kind of unique position in that people care about ranking well in Google.

So if Google say, “Well, to rank well, you’re going to need to be with a secure page and you’re going to need to use an HTTPS,” then clients are far more likely to say, “Oh yes, okay, that’s obviously the thing we have to do.” And let’s say for new site, there shouldn’t be too much trouble, and they can actually get free SSL certificates, providing they’ve used letsencrypt.org to get a free certificate, and yeah, I mean, it’s not too expensive to do if you’re doing it from the outset.

Emily Lewis: [Agrees]

Rachel Andrew: I can see clients being sort of a bit difficult about it if they’ve got a big site and you’re saying, “What it’s going to take is several days to get this site so you’re not going to have mixed content warnings and things.” You can understand the client thinking, “Well, really, what’s in it for me if we do this?”

Lea Alcantara: [Agrees]

Rachel Andrew: But I think if you’re starting a new project at this point, then it’s just worth pointing to the stuff from Google and saying, “You want to do this now because it’s going to be expensive later on to do. I think it’s going to become more and more important, not less important that you do it.” So I would say it’s the things you just be putting now into proposals and saying, “This is what we need to do because it’s going to be a problem if you don’t.”

Emily Lewis: Good suggestion.

Lea Alcantara: So Em, I know you built a corporate intranet where the entire system, all the content, was behind a secure login. Why don't you talk about that a little bit?

Emily Lewis: Yeah, really, my solution was a mix of what Rachel has described with Perch's Members App, and just to be specific, I didn't use Perch, but the concept of using that app to add like essentially a tag or a flag on a member record, and then as well as what you were describing you did for AOHNA, our non-profit client, with securing files like that. So essentially, it was the only public facing thing was a login screen, which used ExpressionEngine out of the box login capability with their built in Members, but the login then prompted them to walk through two additional screens that had flags where they had to accept certain terms and conditions.

For that, really, it has little do with security, but I did use Solspace's User add-on, which hearing Rachel describe Perch's Members App, I think it's kind of similar in the sense that that allowed me to once they said, "Yes, I accept these terms," it added a flag to their record, and then when they accepted the next term, it added a flag, and if they didn't have those flags in their records, they couldn't get past those two screens or even the first one rather, but again, that's not much for security.

I think the main thing that I thought was interesting about that project, and this was so, so long ago, is that I relied on an ExpressionEngine add-on called Force SSL, and I looked on Devot:ee this morning, there are actually two that are called Force SSL, and I can't remember for the life of me which one I used, but essentially, it just lets you configure your templates, your ExpressionEngine templates, which ones should be running as HTTPS, which kind of gives you a little bit more granular control.

Frankly, given the situation, thinking about it now, I probably didn't need that because the whole thing really needed to run on HTTPS [laughs], but when you're starting out, you use what's available to you. It was very much the same thing, adding the flags to the user records to sort of validate their access, native log-in capability with username and password, and then protecting the files and assets using different – I didn't do what Rachel had suggested, which is moving those files out of the public root, which is a really good suggestion, but using what you described, Lea, with your disallows and no-index, no-follow, things like that.

It's really simple and straightforward, but there are definitely some add-ons out there. If this is something you want to explore and you're feeling a little – I don't know – like you need some support. I feel like that's why I sometimes turn to add-ons when I'm not really sure how to do it out of the box.

Lea Alcantara: Right. I think that's a lot of people though. It's just like, "Okay, what's the next step? Okay, this has the word 'SSL' in it." [Laughs]

Emily Lewis: [Laughs]

Rachel Andrew: [Laughs]

Emily Lewis: Oh, right, you've got to start somewhere. I mean, I remember this project was literally one of the first ones I did when I went freelance, and you're taking projects that are probably, you know. If you've been doing this for ten years, you know you can take a project that you may not know everything about, but you know you'll be able to figure it out and give the client a good product at the end.

Lea Alcantara: Right.

Emily Lewis: It's just sort of how you go about figuring it out and then what you learned through the years after.

Lea Alcantara: Yeah, absolutely. I mean, it was one of those things where I remember one of our clients, they wanted to have a survey on their site and they wanted to get every single piece of content from their particular clients, and I'm like, "But do you want to have their address? Do you want to have their credit card?" It's like, "No, don't put that on the survey."

Emily Lewis: [Agrees]

Lea Alcantara: Like I remember at least I still had the foresight without understanding completely. I'm like, "I don't think this is going to work out for you." But I didn't understand that I could have served it through SSL or through some other login situation at that time.

Emily Lewis: [Agrees]

Lea Alcantara: But this security burden is on us to educate the clients.

Emily Lewis: [Agrees]

Lea Alcantara: We need to learn more about it so we can make smarter decisions so that we don't have like a disastrous situation where suddenly their client stuff is compromised, and it all falls back to the website form, and it's just like a simple survey that isn't even doing anything to their business, but it's just more like a follow-up thing for them. It's not even a purchase, but they don't think about how sensitive some of that information actually is.

Rachel Andrew: Yeah, I mean, and also in various occasions, certainly in the UK where I am, there are laws around how much data that you can collect and how you store it, and that you are storing it securely.

Emily Lewis: Oh.

Lea Alcantara: [Agrees]

Rachel Andrew: So you need to actually be able to say that you're storing people's personal data. I'm not talking about credit cards here. I'm just talking about email addresses and phone numbers.

Lea Alcantara: Right, right, right.

Rachel Andrew: If you're storing people's data, you need to be showing that you're taking reasonable precautions to store that data carefully.

Lea Alcantara: Right.

Rachel Andrew: So you may well have those kinds of things that you have to be concerned about, and really, I mean, and certainly in the UK and Europe, one of those things is to not collect data you don't need.

Emily Lewis: Absolutely.

Rachel Andrew: If you don't need the information, don't collect it. [Laughs]

Emily Lewis: Right.

Lea Alcantara: Yeah, yeah, yeah, right.

Rachel Andrew: Because then it can't be compromised, and so yeah.

Emily Lewis: Right.

***Timestamp**: 00:40:00*

Rachel Andrew: After nothing, yes, the end client might think, "Well, it would be great to have all this information." So you kind of explain to them why that might be a problem.

Emily Lewis: Right. So we talked a lot about how we each have used CMSs to help us with securing site content. I'm curious, Rachel, have you ever used something like Facebook login or some type of third-party integration for the actual login part of securing content?

Rachel Andrew: We haven’t yet. It’s something that we’ll probably do. It’s certainly a thing that’s kind of on the road map that we will offer that. I think a lot of people would like to use Facebook logins or Twitter login, or depending on how technical our users are and what sort of users they are.

Emily Lewis: [Agrees]

Rachel Andrew: Probably a thing that we probably will end up rolling into kind of the Members App for Perch is that people will be able to log in with some sort of social login.

Emily Lewis: [Agrees]

Rachel Andrew: I think that’s very much the way that people are going, and there were also services out there that do things like two-factor authentication where you have to log in and use your phone or whatever to get in.

Emily Lewis: [Agrees]

Rachel Andrew: And I think that they’re a good option for people who don’t want to kind of roll all that out themselves to use some service that provides that we will be looking at, adding it if it’s an interest obviously from Perch customers for those.

Emily Lewis: [Agrees]

Rachel Andrew: Because, yeah, I think having to run the passwords, it seems like it is really an archaic thing that we have to do, and I'm sure eventually it's going to go away. It's just that we haven't really got a great solution for that yet. Nothing has really come through that said, "Yeah, this is going to replace password logins."

Emily Lewis: [Agrees]

Rachel Andrew: But I think that there are solutions around, and increasingly, people are going to want to use them.

Lea Alcantara: So we've been talking a lot about HTTP, HTTPS and security certificates. However, I've heard about this new protocol called HTTP/2. Can you tell us a bit what that is, and why sites are moving to this protocol?

Rachel Andrew: Okay, so HTTP/1 or HTTP/1.1 is essentially what I suppose you call it the web runs or it has run on for – well, since forever. So that's Hypertext Transfer Protocol [HTTP]. So this is a really, really old protocol. It was defined in '91. The last major revision was in '99. So websites that were done…

Emily Lewis: Oh, wow.

Rachel Andrew: Yeah, really. [Laughs] So websites done then were quite different to what we're doing now, and the problem with the old version of HTTP or HTTP/1 is that it doesn't perform well when you're retrieving lots of resources, and the average modern website has lots and lots of resources, lots of images, lots of style sheets, JavaScript files, all kinds of things that we're trying to get down the pipe to display the website, and HTTP/1.1 does not perform that well with that, and so there was a sort of a project called SPDY, which came from Google, which attempted to address this, and then it sort of formed that HTTP/2 or H/2 came along, which basically is just a modern version of the protocol that we use for our web browsers to observe and to get the content into our browser, and the kind of key thing about that is that where at the moment, you know, if you're using HTTP/1, it will be good advice to turn all your images into a sprite, for example, so you're only sending one thing down the wire and not lots of things.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

Rachel Andrew: Well, actually, with H/2, it might be better to have things all split up because you don't have this problem with everything trying to get down one pipe where it's…

Lea Alcantara: One pipe, yeah.

Rachel Andrew: So it might be performant to have all your images separately and be able to serve exactly what you need in each page rather than trying to serve a big sprite, which is unused by everything, and there's a whole bunch of optimizations and things that are going to be different about how we developed in the future once we're using H/2 because it's been designed really for the sort of stuff that we're trying to do on the web. So yes, it's a new protocol. So the reason it ties into security is that in order to use H/2, you need to be on SSL. You need to be serving your site over HTTPS.

Emily Lewis: Oh.

Lea Alcantara: Right.

Rachel Andrew: So that's kind of the first thing you need to be able to do before you can move to H/2 is that you need to be serving your site over HTTPS in order to be able to take advantage of the new protocol.

Emily Lewis: [Agrees]

Rachel Andrew: And some shared hosts are just starting now to offer HTTP2. If you're bringing your own servers, it's something that you can do. I haven't done it yet on ours, but I will do because actually the support out there in the browsers is very, very good. So all the modern browsers do support that now, I think it certainly does, Firefox, Chrome, Safari, Opera and all those browsers already support H/2 so you've got the client support. It's really whether you have the support on your server and your site is also on HTTPS.

Emily Lewis: [Agrees]

Rachel Andrew: And then once you’ve done that, once you’ve moved over to the new protocol, instead of thinking about what are the development proxies you’re using, if that’s going to be helpful once you’re using that.

Emily Lewis: Oh, wow! What we do is constantly changing. That’s a big shift. It sounds like a big shift.

Rachel Andrew: Yeah, I think there’s quite a lot, and there’s a whole bunch of stuff around this. It’s very hard to just talk about. I wrote and asked on Smashing Mag which we went through some of the sort of main points you might need to think about. But these things we like at the moment. A lot of us concatenate all of our CSS into one like big file, but maybe actually if you’re using H/2, maybe the better thing to do would be to have lots of small files that are just for the page that’s being viewed, and have really sort of have just the data that you need on that page.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

Rachel Andrew: So you breaking things down more into modules as well as than having one enormous file, and so I think it’s still early days in terms of those all working out how we’re going to use this.

Emily Lewis: [Agrees]

Rachel Andrew: But certainly, yeah, a big change is kind of happening under us really as we speak as I'm sure.

Emily Lewis: I'm guessing, what I'm hearing from you though, it's maybe not time to explore H/2 for production client sites, but maybe making a move to HTTPS for existing sites and new sites is a good way to kind of get ready for this shift?

Rachel Andrew: If your host supports H/2 and you're able to use HTTPS, there's absolutely no reason not to move to it immediately.

Emily Lewis: Okay.

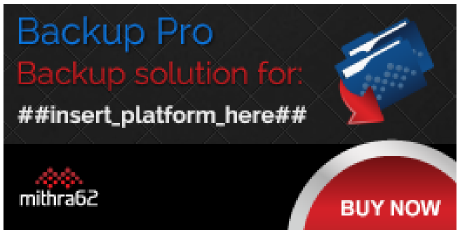
Lea Alcantara:

Rachel Andrew: If the browser comes to your site and doesn't support it, this browser is incompatible, so that browser will just use the old protocol.

Emily Lewis: Ah, okay.

Lea Alcantara: Right.

Rachel Andrew: The kind of the tipping point would be if more of the people that come into your site are going to benefit. So if you've got 70% of people on browsers that support H/2, then you far best to optimize for them than the older browsers because more people are going to benefit from it.



Emily Lewis: [Agrees]

Rachel Andrew: So it's kind of looking at logs. You can look on Can I Use to see which browsers now support H/2 and then you can compare that to your data. I think for a lot of people, the sticking point will be getting to HTTPS and also whether their hosting company upgrades their server support, but all of the web servers now, Apache, can do H/2, and I mean, Ngnix, so I think all of the main servers can do it.

It's whether you're host have upgraded yet, but I mean, again, it's something that you can, if you're really going for performance and you've got 31 browsers come in to your site, you might want to move to a hosted support sooner rather than later because it's going to make a difference. There are various sites that will show you the visible difference when you load a lot of those assets over the two protocols, and it's really interesting, and hopefully it's going to make a big difference in terms of the speed of the web.

Emily Lewis: Good stuff.

Lea Alcantara: Yeah. That's just so fascinating. I just feel like in some ways, even with technology moving really quickly, as you mentioned, how like, "Okay, maybe sprites might not be the answer with this HTTP/2." But going back to the old school way of just like separating it all, but have it on a different protocol essentially would be faster. It means that all the stuff we learned ten years ago might still be relevant? [Laughs]

Rachel Andrew: [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: Like it feels like even with responsive design and all this stuff, like the more things change, the more they stay the same.

Rachel Andrew: Yeah. I think it's going to be quite interesting. I think a lot of people haven't come across this as yet.

Emily Lewis: Right.

Rachel Andrew: Because it's kind of hidden from you, you know?

Emily Lewis: Right.

Rachel Andrew: It's a thing that happens between the browser and the server, and unless you're particularly interested in servers, you may not even be aware which protocol your particular hosting company is using.

Emily Lewis: Right.

Rachel Andrew: I think the issue is that it starts to become useful to figure that out and to optimize for it.

Emily Lewis: Right.

Rachel Andrew: I think if people are saying any kind of build scripts. You're using Gulp, or something like that, to minify files and things, that kind of makes it easy to make the switch here. You can build your site in such a way that you can, for instance, sprite your images or not sprite your images, or concatenate your CSS or not concatenate your CSS.

Emily Lewis: [Agrees]

Lea Alcantara: [Laughs]

Rachel Andrew: So you can kind of be in a position where you can decide how you do it and so you can actually make those decisions. If you've got a site, you decided you want to hold back or you can't move the hosting to H/2 at the moment, but you'd go into it in the future and you could actually build it in such a way that makes it quite easy to sort of rebuild all your assets and using it in a different way.

Emily Lewis: Right.

Lea Alcantara: Very cool. So before we wrap up, what's on the horizon for Perch?

Rachel Andrew: So we've got a UI redesign in progress.

Emily Lewis: [Agrees]

Rachel Andrew: Obviously, Perch is quite an old product and we've never done sort of a big UI redesign, mainly because people use Perch. They put this information to their clients. They want to be able to upgrade it without their client sort of going back and go, "Oh, everything has changed." We try not to do that.

Lea Alcantara: Right, right.

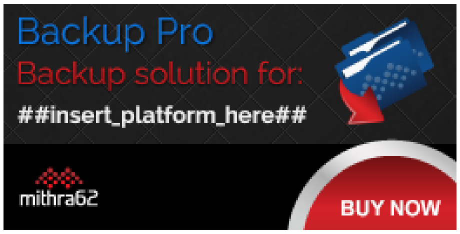
Emily Lewis: [Laughs]

Rachel Andrew: Because that actually is a working site. If you've got 50 sites with 50 clients who all use the software and then suddenly all changes, that could actually be quite a big problem, but it's kind of got to a point where more people want to go to managed sites for mobile devices, which obviously wasn't an issue when Perch first came out because it was sort of before it was responsive design really.

***Timestamp**: 00:50:02*

Lea Alcantara: Yeah.

Rachel Andrew: And we've done bits to kind of retrofit in responsive design, but we wanted to do a really sort of solid responsive rebuild to really take advantage of the way people want to manage sites today. So that's kind of the next thing really that we'll be shifting. So sort of our next release of Perch will be this UI redesign, which we're busy doing, so that's big and that's quite exciting and it's starting to look really nice. I can't wait to actually have it on my own site now.



Emily Lewis: [Laughs]

Rachel Andrew: As I was sort of saying, “Oh yeah, it actually does. This is looking really good. It’s getting more easy to use.” So that’s cool. We try to keep it looking like Perch and feeling like Perch.

Emily Lewis: [Agrees]

Rachel Andrew: But just sort of make it easy to use on all the different devices people want to be using.

Emily Lewis: Yeah, it’s a good client selling point.

Lea Alcantara: [Agrees]

Emily Lewis: We’ve got one last question, which comes from listener Mark Phoenix, who hosts the Relative Paths podcast.

Rachel Andrew: [Agrees]

Emily Lewis: He wants to know how do you find time to fit in working on Perch, speaking at conferences, working on grids and layouts, and all the rest.

Rachel Andrew: I’m fairly organized. Also, I’m good at working on the road, so you know?

Emily Lewis: [Agrees]

Rachel Andrew: If I got on a plane and it's a 10-hour flight, that's ten hours of work.

Lea Alcantara: [Agrees]

Rachel Andrew: I can't remember. I very, very rarely will watch a film. I go on a plane to work. I get to my hotel, I sat at my desk and I work. [Laughs] So I'm pretty good at working on the road and so I couldn't do all this if I wasn't.

Emily Lewis: [Agrees]

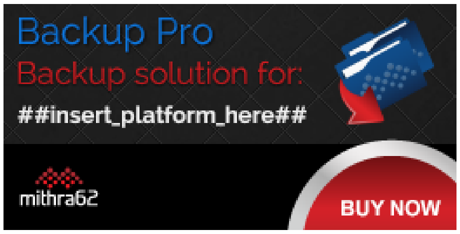
Rachel Andrew: I mean, I just try to be organized with this stuff and I rely very much on OmniFocus. I have things. I have contacts. Even when I am flying, I have contacts or when I'm tired, you know?

Emily Lewis: [Agrees]

Rachel Andrew: So I'm always trying to do whatever is relevant in the position I'm in. So yeah, I think, certainly with the travel, you kind of have to get yourself really organized in order to get all of that stuff done.

Lea Alcantara: [Agrees]

Emily Lewis: And have a strong work ethics and self-discipline.



Rachel Andrew: Yeah. Well, you see, that goes back to dancing. If you’re doing point ballet, you’re training yourself. [Laughs]

Emily Lewis: Training, yeah.

Lea Alcantara: [Laughs]

Rachel Andrew: You know how to work. [Laughs]

Lea Alcantara: I love it.

Emily Lewis: [Laughs]

Lea Alcantara: Full circle. Speaking of full circle, we’ve got our Rapid Fire Ten Questions so our listeners can get to know you a bit better.

Rachel Andrew: [Agrees]

Lea Alcantara: Are you ready, Rachel?

Rachel Andrew: Okay.

Lea Alcantara: First question, morning person or night owl?

Rachel Andrew: Morning person, but I used to be a night owl. I've retrained myself.

Emily Lewis: [Laughs] What's one of your guilty pleasures?

Rachel Andrew: I don't really have any. This I find very strange question. It's like if I like them, would I have it and I wouldn't feel guilty about it?

Emily Lewis: [Laughs]

Lea Alcantara: Nice.

Rachel Andrew: Yeah. [Laughs]

Emily Lewis: I love that answer. [Laughs]

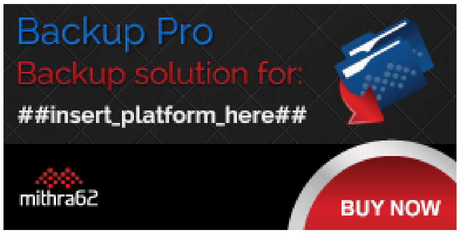
Lea Alcantara: That's great. What software could you not live without?

Rachel Andrew: OmniFocus. It's probably the only thing keeping me on a Mac at the moment.

Emily Lewis: [Laughs] What profession other than your own would you like to try?

Rachel Andrew: Probably something in fitness.

Emily Lewis: [Agrees]



Lea Alcantara: What profession would you not like to try?

Rachel Andrew: Oh, anything that involve being on a telephone. [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Emily Lewis: If you can take us to one restaurant in your town, where would we go?

Rachel Andrew: Oh, I don't know. [Laughs] Oh, no, no idea. Probably, there's a place I like called Zazu Kitchen, which is really just sort on the road from us and they're very friendly. It's kind of our local, so that's probably where I'll take you.

Emily Lewis: Is it like local food with like sandwiches.

Rachel Andrew: Yeah, yeah, very much. It's very much local food and it's just a small local chain of a few restaurants and they're really cool, so yeah.

Lea Alcantara: If you could meet someone famous, living or dead, who would it be?

Rachel Andrew: The French author, Sidonie-Gabrielle Colette.

Emily Lewis: Oh.

Lea Alcantara: Oh.

Emily Lewis: If you could have a super power, what would it be?

Rachel Andrew: Oh, probably like being able to teleport.

Lea Alcantara: Nice.

Emily Lewis: [Agrees]

Rachel Andrew: I do like planes a lot, but it would be much quicker to get to places if I didn’t have to sit in them.

Lea Alcantara: Right. What is your favorite band or musician?

Rachel Andrew: Ani Difranco

Emily Lewis: Oh nice. Last question, pancakes or waffles?

Rachel Andrew: Neither, they’re both gross. [Laughs]

Lea Alcantara: [Laughs]

Emily Lewis: [Laughs]

Rachel Andrew: They're saucy. I had to go to another place for breakfast in a hotel that I was staying four weeks ago because they had a waffle machine and the whole place stunk of waffles.

Emily Lewis: Ugh.

Rachel Andrew: And I could not eat my eggs with this smell.

Lea Alcantara: [Laughs]

Emily Lewis: [Laughs]

Rachel Andrew: I have to go somewhere else.

Lea Alcantara: [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: Oh, I love it. I love the alternative answers to these questions.

Rachel Andrew: [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: So that's all the time we have for today. Thanks for joining the show, Rachel.

Rachel Andrew: Oh, it’s been great. Thank you.

Emily Lewis: In case our listeners want to follow up with you, where can they find you online?

[Music starts]

Rachel Andrew: I’m @rachelandrew on Twitter and pretty much everywhere else, so that’s a good way to find me. My personal site is rachelandrew.co.uk and with Perch, it’s grabaperch.com.

Emily Lewis: Thanks again, Rachel. It was great having you on today.

Rachel Andrew: Thank you.

Lea Alcantara: CTRL+CLICK is produced by Bright Umbrella, a web services agency obsessed with happy clients. Today's podcast would not be possible without the support of this episode's sponsor! Thank you, Backup Pro!

Emily Lewis: We’d also like to thank our partners: Arcustech and Devot:ee.

Lea Alcantara: And thanks to our listeners for tuning in! If you want to know more about CTRL+CLICK, make sure you follow us on Twitter @ctrlclickcast or visit ctrlclickcast.com. And if you liked this episode, please give us a review on iTunes, Stitcher or both! And if you really liked this episode, consider donating to the show. Links are in our show notes and on our site.

Emily Lewis: Don’t forget to tune in to our next episode when we’ll talk to Ilise Benun about marketing for web freelancers and agencies. Be sure to check out ctrlclickcast.com/schedule for more upcoming topics.

Lea Alcantara: This is Lea Alcantara …

Emily Lewis: And Emily Lewis …

Lea Alcantara: Signing off for CTRL+CLICK CAST. See you next time!

Emily Lewis: Cheers!

[Music stops]

***Timestamp**: 00:55:19*