Certbot Usability Case Study

MAKING IT EASIER TO GET HTTPS CERTIFICATES

Author: Soraya Okuda

View this report online:
https://www.eff.org/wp/certbot-usability-case-study-making-it-easier-get-https-certificates

Certbot Usability Case Study

MAKING IT EASIER TO GET HTTPS CERTIFICATES

Soraya Okuda
Education & Design Lead

OCTOBER 23, 2019

Table of Contents

Introduction

When EFF launched the Certbot tool in 2016, our goal was to help website administrators secure their sites with HTTPS certificates. Since then, our technology and design teams have received feedback from users about their barriers to using Certbot, how they find it, what makes it useful, and what a more usable Cerbot might look like for different people.

Here, we'll give a behind-the-scenes look at how EFF is thinking about making it easier for everyone to secure their websites with free HTTPS certificates. This case study covers a year of internal user research and design strategy sessions with the goal of redesigning our Certbot website. Beyond securing the web, we hope this design case study will be helpful to other nonprofit organizations and independent groups who are also figuring out how to do smaller scale user research for their own technology projects.

What is Certbot? What were we trying to change?

EFF launched Certbot in Spring 2016, as a command-line tool to be installed on a server, helping people to automatically enable HTTPS on their website using a Let's Encrypt certificate. Below is a screenshot from our old instruction generator website, which you can access [via the WayBack Machine](#).
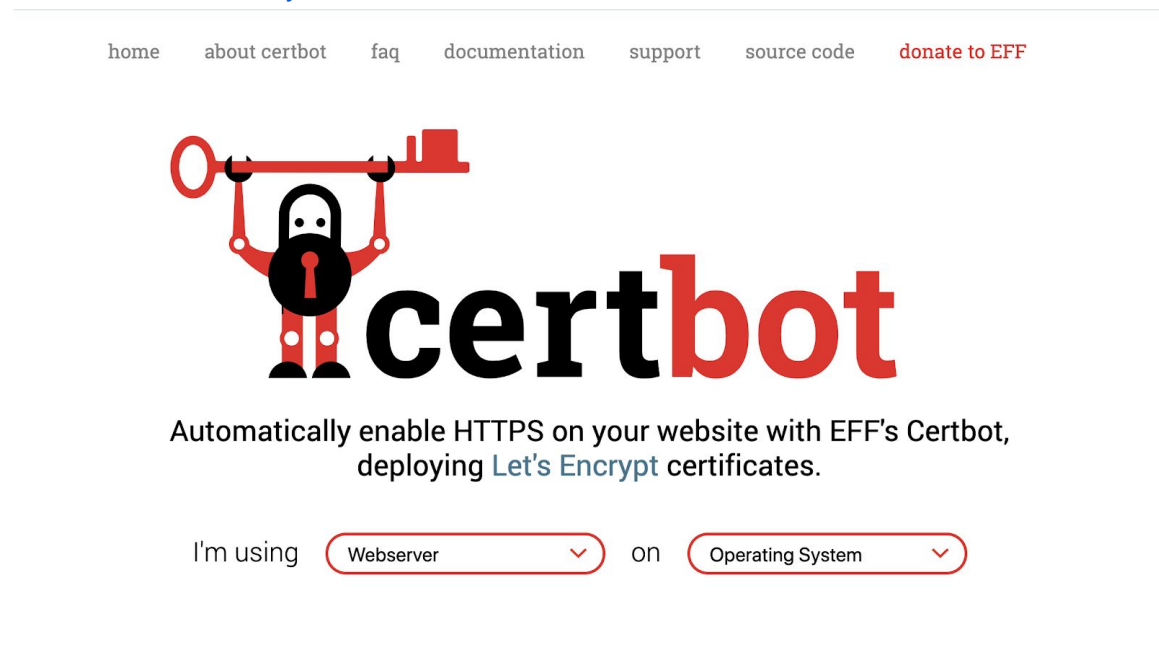
After receiving many similar questions in the Let's Encrypt community forum and encountering confused users in person, we came to realize our old site designs were not aligned with our actual users' understanding of our service. We were making some assumptions that were embedded in the design. Primarily, we assumed that many of our

users were familiar with the command line, with the requirements for Certbot, and that they would know how to look for the appropriate instructions for their website setup.
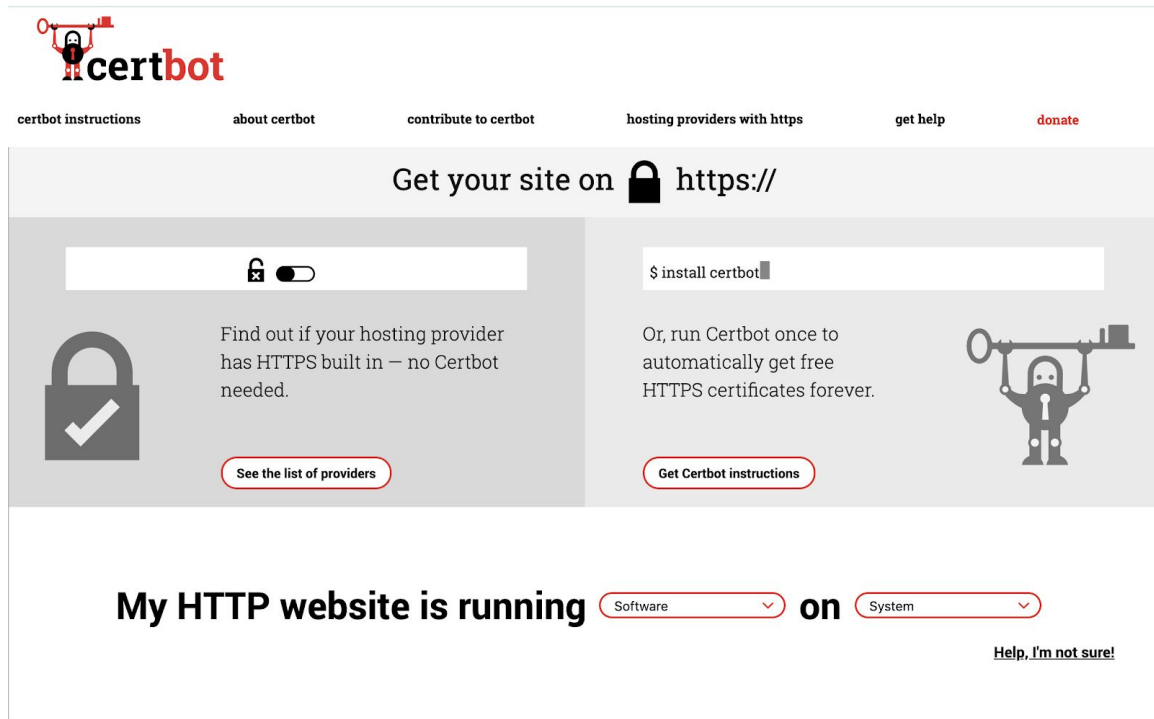
Rather than work from assumptions, it was important to our team to make our design changes based on *user research* — that is, studying people's perceptions, behaviors, and comprehension to improve the experience of a project. By doing this, our intention is to remove points of confusion and frustration, and to make a product easier to use overall. In this way, user research can inform a design to make a product more *usable*, or easy to use.

In the case of a large tech company, user research might include a large team of researchers, designers, and developers working on a single or constrained set of products. In the case of a smaller organization like ours, it is typically a more ad hoc process as time allows, while balancing other products.

Our user research and design team for Certbot included the following people:

Designers	*Developers*	*Project managers*
Michelle Chang Soraya Okuda	Brad Warren Erica Portnoy Mona Wang Seth Schoen Syd Young	Jeremy Gillula Lena Gunn Max Hunter Kim Carlson

We would like to thank our many gracious testers and interviewees for their time and feedback in making Certbot better.

For this redesign, we prioritized making improvements along these lines:
- A list of hosting providers that offer HTTPS as part of their standard hosting packages
- A glossary with simple definitions relevant to Certbot users
- New explainers for our Certbot command-line tool
- Clearer paths to get help and to contribute to Certbot

To settle on these changes, we first identified our goals for doing user research. Before approaching the bigger question of how we do user research for Certbot, we had to think about what we hoped to accomplish: what kinds of users are we talking to, and what do we hope to learn more about?

We then took steps to scope our user research efforts, keeping in mind our time and budgetary constraints. The overarching steps were:

1) **Identifying what we didn't know** about our users and were curious about;
2) Creating a plan for **recruiting and screening volunteers**;
3) Meeting with these volunteers through **user testing and conducting interviews**;
4) **Summarizing the research** and reflecting on actionable changes;
5) **Designing, writing, building and launching the new site**;
6) **Revisiting the design** after our launch and further user testing.

For a full overview of our steps, see our Appendix.

Certbot's User Research Process

After reflecting on past feedback and thinking about Certbot's design, we knew that we needed to do more work. Given that EFF limits how much information is collected about users, we had minimal quantitative metrics around who uses the site. We decided to focus our efforts on qualitative work—primarily through interviews and user tests. Our team wanted to understand the HTTPS ecosystem before making significant design changes, and we decided to do a small-scale round of user research.

We wondered:
- How do people get to the Certbot website and command-line tool in the first place?
- How do people get HTTPS certificates overall? What part of the process doesn't align with their expectations? What's their understanding of how it's supposed to work?

Simultaneously, we began scoping our plan for user research with website administrators.

We determined that the goal of our user research would be to figure out how people get information about how to get a certificate—not necessarily information about using Certbot, because sometimes Certbot is the wrong choice for their needs. After having found that information, we wanted to know how these website administrators use Certbot to get a certificate.

We ran a small set of user tests with gracious volunteers who wanted to get HTTPS on their website.

We identified our qualifiers for our small set of testers as:
- Having experience running a website on a server. At minimum, they needed an awareness of SSHing into a server and their server environment.
- They needed to be patient people willing to troubleshoot alone.

How We Recruited Our Volunteers

After sending a few calls for participants via social media, we chose seven participants to run user tests with over a video chat platform. We selected these seven people for user tests, based on the parameters that they had experience running a website on a server, the ability to SSH into their server, and already had server software up and running. We did not require experience with HTTPS.

Additionally, we ran a separate series for a group of technical users who dropped by our table for informal tests and feedback at a hacker conference (HOPE 2018). We also ran three user interviews with professional digital security trainers from the community,

and looked at [existing research on the usability of Certbot and Let's Encrypt](#) to examine how other testing sessions were modeled.

During these user tests and interviews, we designated one EFF person to facilitate the discussion and one EFF person to take notes, while other colleagues observed. At the end of each session, we conducted a fifteen-minute debrief among our research group over areas of misunderstanding or confusion, and typed up notes summarizing our takeaways.

Details on Testing: "Get HTTPS"

Before running our tests with external volunteers, we ran one internal test for practice with a beginner colleague.

We identified our task length as anywhere from as short as 1 minute to as long as 2 hours. The basic prompt was "You want to get HTTPS for your website."

We did not direct them to use Certbot specifically, and avoided mentioning Certbot during the tests.

Throughout the process, we did our best to watch them move through the task without reacting, occasionally reminding our testers to [think aloud](#) and to share what they were looking for. Our volunteer testers included people with basic familiarity on command line navigation (e.g. knowing how to navigate directories), to deep knowledge with Linux systems administration for corporations.

Our prompt was broad enough to allow volunteers to explore a variety of strategies for getting HTTPS on their website. Our testers shared their screens as they narrated: We watched people as they Googled questions and error messages, were actively troubleshooting and typing commands until they worked, mentioned services they use, pointed out Control Panels for various web hosting providers. We monitored what they typed into their terminals, and so on, until they were able to get the green lock beside their website URL.

Results: Puppersonas

At the close of these user tests and interviews, we developed personas reflecting patterns of attitudes, backgrounds, use cases, and frustrations we noticed. And because our team likes dogs…the personas featured dogs.

Personas often include a name, picture, and occupation. Personas are helpful for designers and developers to return to in the design process, as they help remind us about the needs of people using our products. Personas might include threat models (particularly for products addressing high risk users, such as the excellent digital security personas of the USABLE project), considerations around disabilities, considerations around stress cases, financial and technical limitations and so on, and may include quite a bit of detail around the user's context. For Certbot, it was helpful to include what kinds of questions our personas might have, areas of misunderstanding around HTTPS and certificates, their motivations, types of operating systems and server environments they might use, why they might be on the website, and whether Certbot is even appropriate for them.

Below are three example personas that we continually referred back to as we imagined their movement through the Certbot site. See more persona details in our Appendix..

SASHA

Our dream

*"How do I get the green padlock that I would see at Google?"*

Image - By EFF - CC BY

Sasha is Our Dream, in the sense of "the person we imagined our user to be", not "who we wish our user was." She is used to running and securely logging into a remote server, and meets the requirements for using Certbot comfortably. She's pretty used to using the command line and administering her remote server (running software like Apache or Nginx). She has her own art project HTTP website already up and running.

Sasha has gone through the process of installing Certbot and following the instructions generated from certbot.eff.org for her setup. However, she gets an error message. Sasha might find the wording from the Certbot terminal confusing, and she's not sure where to look for good resources on debugging and troubleshooting for her specific errors. She's not yet sure if it's a Certbot-specific issue, if it's related to her DNS configuration, or if it's related to her site.
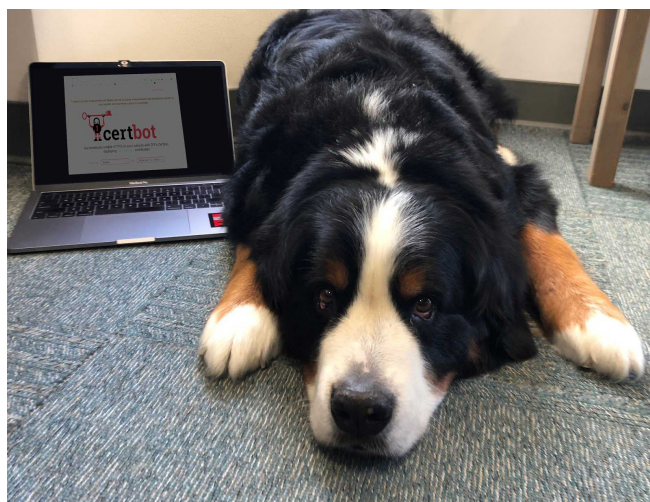
For Sasha, we need to provide more resources for next steps. Sasha helped remind us of the following considerations:

Now that you've downloaded Certbot, what's next? What happens if you've downloaded it successfully, but can't get Certbot running? Where do you go for more information if you've successfully gotten an HTTPS certificate, but aren't sure it's meeting standards? What do you do if Certbot works perfectly? Is there anything else you should do? What's a wildcard certificate?

For users like Sasha, we focused on improving our wording for next steps and [for getting help](#), as well as improving our definitions by building this [glossary](#).
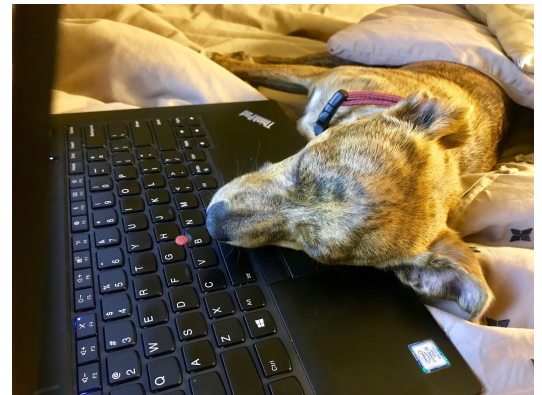

JORDAN

Encryption Professional

"Certbot is so easy to use! Using the standalone plugin manually every three months is so easy."

Image - By EFF - CC BY

Jordan (Encryption Professional) is a seasoned user of Certbot, and is constantly making websites, whether for work or for personal amusement. Jordan is manually renewing their HTTPS certificates.

We would love to see them taking advantage of Certbot's automatic renewal option. This would save them some effort and reduce the risk of forgetting to update the HTTPS certificate.

For Jordan, they're more likely to pay attention to prompts in the terminal—we need to help them understand automatic renewal a little bit better. What this means for us is providing additional prompts in command line responses for how to set up automatic renewal. Additionally, we determined that highlighting the automatic renewal use case through our writing and in our documentation might help move Jordan toward this preferred behavior.

JACK

Shared Hosting User

"I remember having trouble with Certbot, so if I was to do this on my own again, I'd probably try this [other service] out."

"[Web hosting provider service] is super easy, for free."

Image - By Kt mac32 - Kate Richardson, Public Domain

Jack is looking for something fast, easy, and usable—he just wants an HTTPS certificate for his wedding invitation website, and a separate certificate for his small business. Jack might think he needs Certbot, but he may already have a free HTTPS option with the hosting provider he's using. Jack is using a hosting provider (e.g. DreamHost) and may not realize he would have to follow his hosting provider's tutorial to turn on HTTPS.

Jack is a very common user, but we didn't notice him in prior versions of our Certbot site. Through user research, we realized we needed to prioritize him and his questions. What's unusual about Jack (Shared Hosting User) is that we actually think Certbot is inappropriate for what he wants to achieve (wanting a quick, easy way to get certificates).

We want to direct him away from using Certbot, and to point him to easier, more friendly options for getting an HTTPS certificate—for example, perhaps using Squarespace or WordPress, which offer HTTPS certificates to users by default.

We were thinking about what resources would be more useful to users like Jack, and specifically, we thought about how the Let's Encrypt Community Forum's helpful volunteer-curated post on hosting providers that offer HTTPS certificates could be more useful to the broader community if it was easier to find. This informed our design for this chart for finding if your hosting provider offers HTTPS certificates as part of their hosting package.

Designing Based on What We Learned: Incorporating User Feedback

After creating the puppersonas, our team circled back with our broader team to act on our findings.

Our design team printed out the personas and distributed them during a meeting. Using sticky notes, we facilitated an exercise where we asked everyone to independently jot down as many ideas as they could for what a given persona might need (one idea per sticky note), such as the changes listed in the three examples for Sasha, Jack, and Jordan. We reviewed all the ideas together, clustered similar ideas, and had people rank which ideas and features for Certbot we should prioritize in the redesign.

We then conducted a prioritization exercise for how we would approach new features to Certbot. We broke apart the site redesign into two pieces: improving user understanding of Certbot, and usability of the site. Over the next few months, we began to act on these findings through the actual redesign process.

*A photo of multiple papers and post-its that came out of site architecture brainstorm session between a developer and designer—we started by reimagining the flow from the front page of certbot.eff.org, and asking "What is the journey that people are taking as they look for information?"*

We imagined how people navigated from the front page to multiple pages on the site, which helped us to flesh out what was needed.

In the process of these brainstorms, we determined that the certbot.eff.org site could have a bigger purpose of helping guide users through getting a HTTPS certificate.

The reimagined workflow for certbot.eff.org prioritizes helping users to 1) find their hosting provider, and 2) understand if Certbot is right for them.
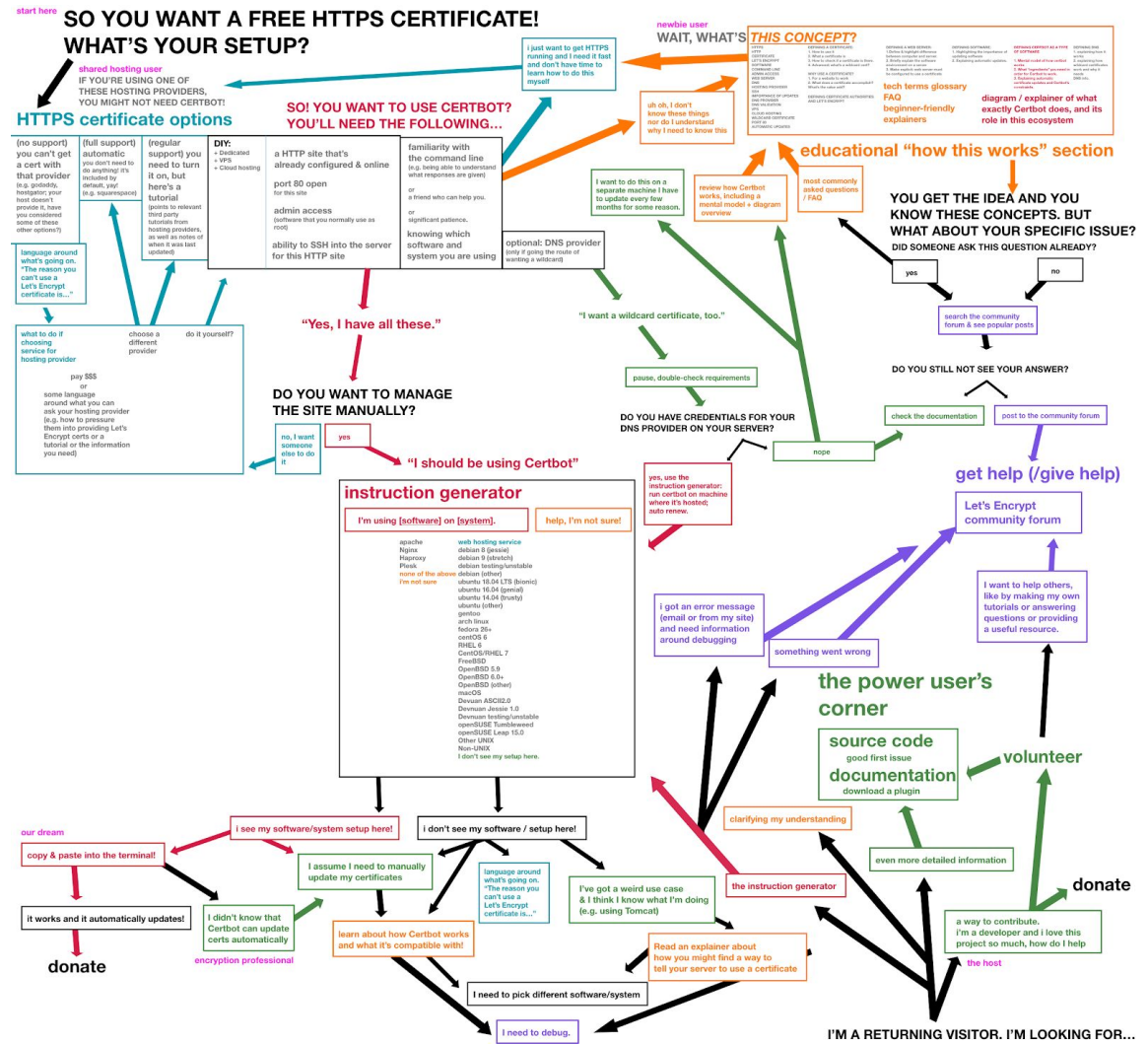
The flow chart is based on the post-it brainstorming sessions with developers—we used this for mapping out the pages, as well as imagining the personas navigating the site from these different angles. (The hot pink text designates parts of the website flow with specific personas in mind.)

After ironing out the user journey, we then partnered closely with a web developer to determine constraints for what this redesign might entail.

We went through 17 drafts of wireframes, from mock-ups to low-fidelity versions to high-fidelity clickable prototypes. Simultaneously, we wrote and edited new content for the site, prioritizing comprehension for beginner users and setting clearer expectations for whether Certbot might be right for them.
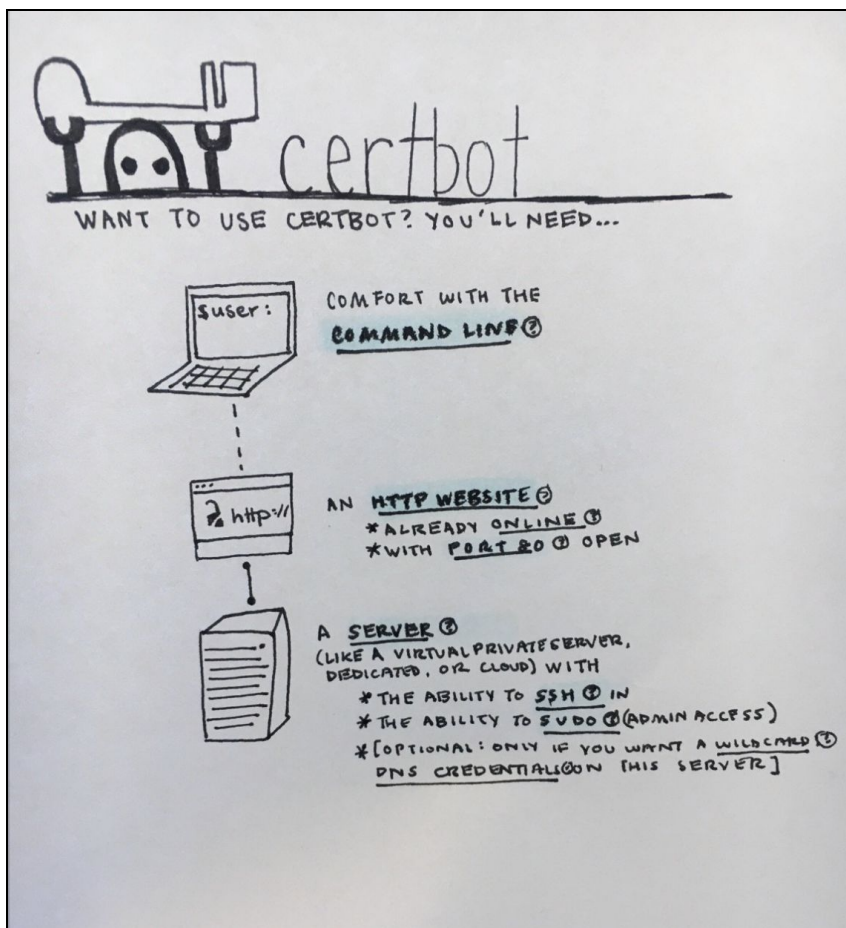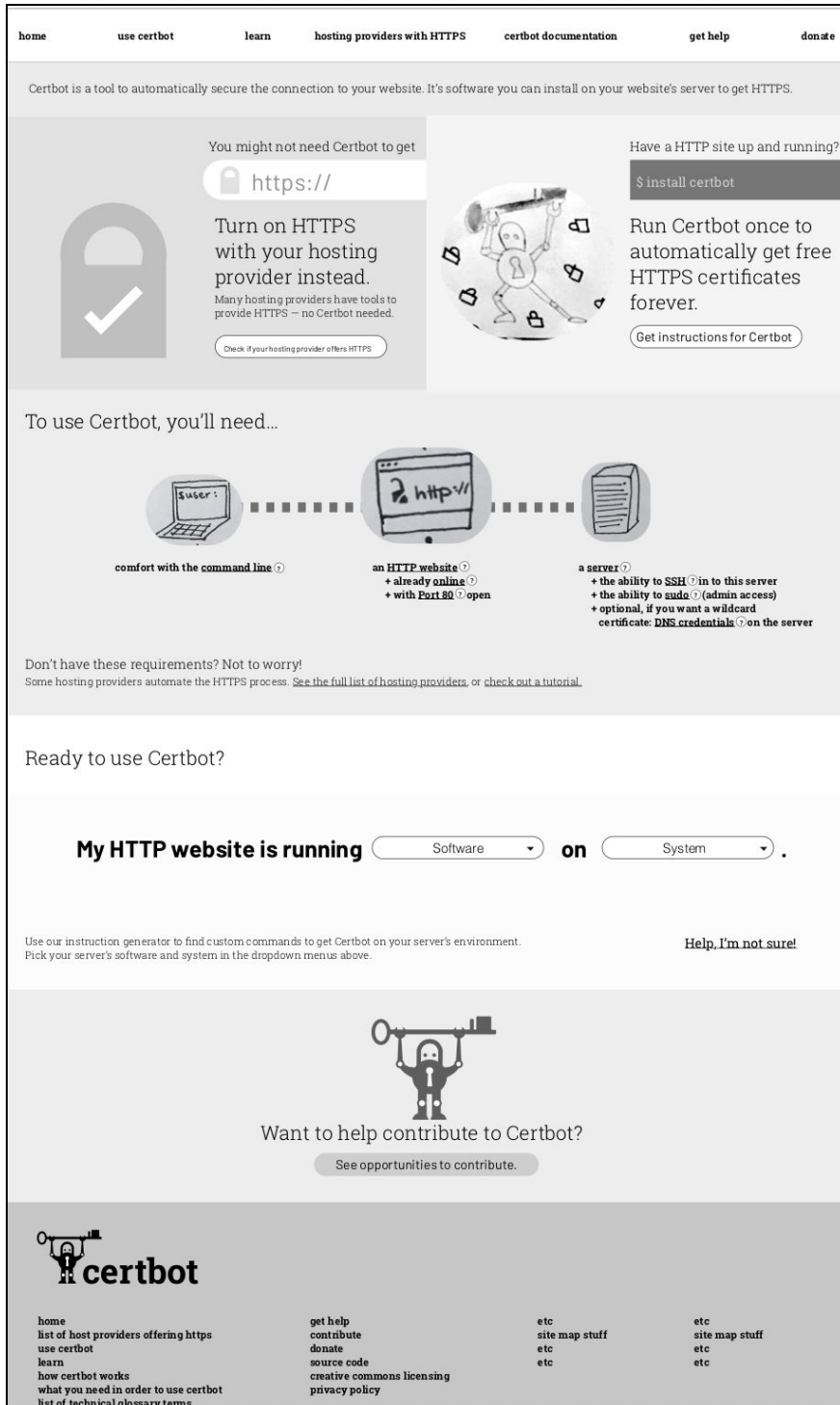
Above is one of our first wireframes for the Certbot site.

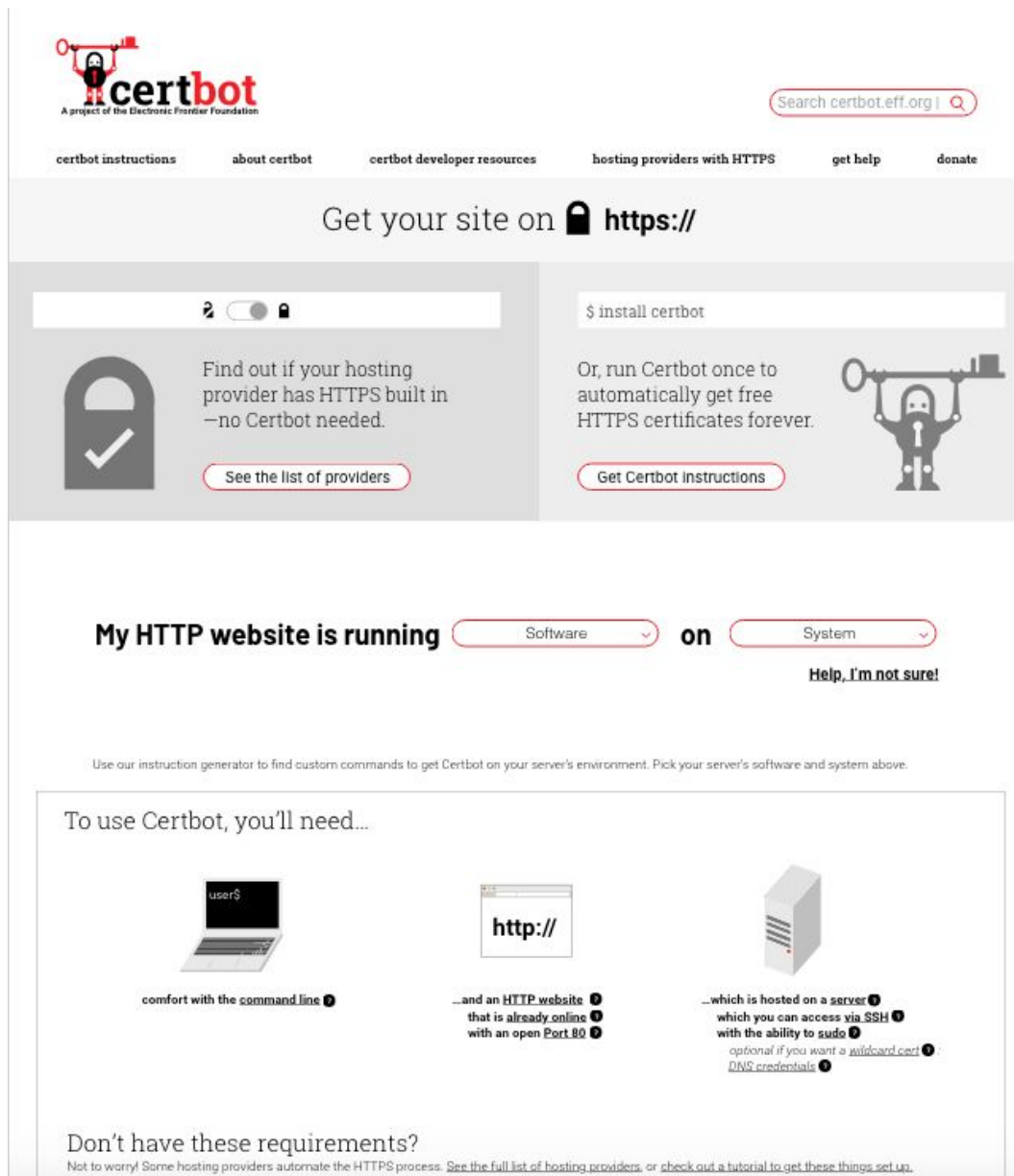And here's the high-fidelity design we landed on, seventeen drafts later.

We launched the redesigned site in June 2019. We still have improvements to make and features to roll out, so there's more to come!

Next Steps: Checking Assumptions and Making Improvements

It's important to check our assumptions. We conducted a small additional round of user tests after launch in late July 2019, using a user testing web service for one mobile tester, one tablet tester, and one desktop tester.

For the prompts in our remote user tests, please check our appendix.

Through these user tests, we realized that many of our goals had been met in terms of user comprehension, though we still need to make improvements in making the web hosting provider page easier to find. We've also received feedback from people in the international web development community and are exploring ways to make Certbot easier to use and understand for all. Subsequently, we've been working on improving our wireframes for the site and examining how to make the Certbot instructions and commands more usable.

If you're a Certbot user, please feel free to drop us feedback on our website repository on GitHub.

If you're a scrappy group of designers and developers, we hope that this post provides some insight into how we conducted user research on a limited budget. If you work with a user testing service and have extra funds for testing (e.g. additional user testing credits), our small team would be very appreciative for the support to do more user research work. And if you like EFF's free and open source projects and our mission to Encrypt the Internet, please donate! We'd be grateful for the support.

Appendix

Overview of User Research Steps for Certbot

First, we needed to set goals. This consisted of:
- Identifying what we wanted to learn, and what we didn't know about our users

After reaching agreement on what we wanted to study, we began to create parameters for our research. This included:
- Identifying the tasks we wanted our testers to complete, tied to our goals
- Creating survey questions to ask our testers to determine eligibility
- Defining parameters for who is an appropriate volunteer for our limited round of user testing and interviews
- Creating a plan for recruiting volunteer testers and interviewees
- Launching requests for volunteers and sending survey questions
- Reviewing our testers and evaluating who met our parameters

Then we began the process of user testing. This included:
- Creating a script, consistent note-taking format, and facilitation style for our user tests
- Dedicating 2-3 hours every week to setting up and facilitating each user test (over a video chat platform), as well as the debrief and notetaking period afterward

We then worked on a plan for acting on the research, and contextualizing what we learned, with the design and development cycle of the website in mind. This looked like:
- Reviewing user testing and interview results
- Paired work between the designers and developers for summarizing the research, including creating personas that reflect considerations from the interviews, tests, and interactions with the Certbot community
- Scheduling a few multi-hour meetings for doing design exercises for creating new features and improving usability, informed by those personas
- Prioritizing these new features and design changes as a team
- Creating a plan and timeline for approaching the redesign

We approached thinking about the actual redesign of the site. This took the form of:
- Revisiting the results from the user tests and information from the personas
- Scoping out the new site architecture
- Writing and editing new content, such as requirements for using Certbot, the glossary, and the new Get Help and Contribute to Certbot sections
- Simultaneously creating mockups and wireframes as working on the content, and developing for the new site
- Launching the website!]

Finally, we revisited our results after launch by:
- Conducting a limited run of user tests on the redesign, which would be reincorporated into the next iteration of designs and development

Certbot Parameters to Screen Participants

Certbot requires some background knowledge about using the command line and about how HTTPS works. There are some cases where Certbot isn't the best choice for someone. We don't want to push people toward Certbot during our user tests, but if they end up there, what can we learn?

Issues We Resolved Before User Research

- What's the profile of a new Certbot user, or someone who ends up at certbot.eff.org?
- Do people teach Certbot to others? Can we get in contact with these people?
- How can we approach Certbot users in a privacy-protecting way?
- What might a survey look like for our users?
- How do we help people through the testing process without leading/saying "Certbot"?

Screening Questions for User Tests

1. What is your experience running a website on a server? What software have you used to do so?
2. What operating system do you have experience running a website on?
3. Do you have experience with HTTPS?
4. Do you have or can you get a domain that you can use to get a certificate for?

(Bonus: How patient are you with troubleshooting?)

Qualifiers
- Somebody who has experience running a website on a server, with the ability to SSH into box, and the server software up and running. Minimum: awareness of SSH and server.
- No need to have experience with HTTPS.
- Preferable that they're setting up Certbot on a web server that they're familiar with. Probably will be with a domain they've set up themselves.
- Very patient people willing to troubleshoot alone.

Task Parameters
- Anywhere between 1 minute to half an hour (if they already have received a cert) to 2 hours.

- Having them talk aloud.

Intro Script Before Beginning Discussion

If testing...

"Hi there, my name is [X] and I'm a [position] at EFF. We really appreciate your time. We're hoping to better understand how people configure and install HTTPS certificates from the free certificate authority, Let's Encrypt. We're not testing you, but trying to understand the service—we're simply trying to understand how people use the site and how we can improve the experience."

If interviewing...

"Hi there, my name is [X] and I'm a [position] at EFF. We really appreciate your time. We're hoping to better understand how people configure and install HTTPS certificates from the free certificate authority, Let's Encrypt. We're trying to understand how people use the site and how we can improve the experience, and really appreciate your time."

Certbot User Testing Script

Our Goal

To figure out how people get information about how to get a certificate (not necessarily information about using Certbot)

Before the Interview

- Verify they have a website by asking for the link

At the Start of the Interview (3-5 minutes)

- Introduce yourself.
- "As you probably know, EFF wants to encrypt the entire web. To help us achieve that, we're doing this user study, which will help us learn how folks who have websites learn about, acquire, and install HTTPS certificates. You don't need any previous experience setting up HTTPS to be in this study—all we ask is that you've already set up a website somewhere that you can configure."
- "We're going to start by asking you some basic questions so we can group together results from people with similar experience levels."
- "After those questions, we'll have you walk us through how you would go about setting up HTTPS on your website. You don't have to use any particular tool or system—just do whatever you would normally do."
- [If you've never set up HTTPS on a website before, we'll also observe what you do to learn about how to get an HTTPS certificate, since we're curious what sort of resources people use to help themselves.]
- "Once you've got your website set up with HTTPS, we have some final questions, and then we're done!"

Questions to Ask Before the Task (3-5 minutes)

1. What does it mean to turn on HTTPS for a website?

2. What does a certificate do?

3. Have you configured HTTPS for a website before? What was the process like?

4. What server are you going to try today? How long have you been running servers like this one?

5. Any questions for me before we start?

The Task (3-45 minutes)

There are two main situations a user might be in. "Now turn on HTTPS."

1. Setting up (getting and installing) a certificate for an existing site. Say: "You have an existing website. I would like you to take steps so that when I go to it from my browser, the site has https."
2. Setting up a new site and getting a certificate for it. Say: "I'd like you to set up a new website, so that when I go to it from my browser, it will have https enabled."

- During the task, occasionally write down the time elapsed in the notes.

Things to Note Regarding the user gathering information

- If they entered any search terms, what are the search terms they entered?
- What sites do they go to? Specifically note community forums, Let's Encrypt (LE) website, Certbot website, Stack Overflow, Github, and specific pages within those sites.
- Do they use `certbot --help` at any point? Do they read the messages that Certbot prints onto the command line?
- Do they copy and paste an error message into google? If so, what was the error message?
- If they ask the tester a question, do not answer but tell them they can use the Internet. Observe if/where they post the question, and what they do while doing so. Do they search the site they're about to post to first? Do they read formula text in the input box? Which site do they go to?

Things to Note if the User is Using Certbot

- What operating system (OS) are they on?
- How did they install Certbot (OS distribution, Ubuntu PPA, certbot-auto, pip, install from source)?
- Which plugin are they using (standalone, manual, nginx, apache, dns)?
- If they're using the nginx or apache plugin, are they hitting misconfiguration errors?

Questions to Ask After the Task (3-5 minutes)

1. [If they used Certbot] What does Certbot do?

2. [If they got any sort of LE cert] How long are Let's Encrypt certificates valid for?

3. [If they got any sort of LE cert] What are you going to do about your certificate expiring in three months?

4. [If they used Certbot] Did you know about Certbot before? Let's Encrypt? What is the relationship between them?

5. [If they're installing Certbot on a preexisting site that uses Nginx or Apache] Have you edited your configuration files or are you using the default

configuration files? If you've edited them, did you use a template from somewhere? If so, where?

6. [Insert miscellaneous follow-up question here]

7. [Insert miscellaneous follow-up question here]

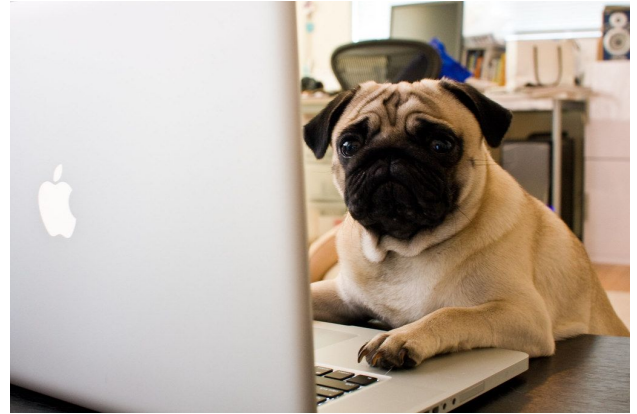8. Any final questions or comments for us?

Certbot Puppersonas

TITUS
The Newbie

"How do I SSH to my server?"

Image – By Puprin – CC BY–NC–SA 2.0

Operating System:

On a Mac or Windows machine locally

Motivations:

Titus is setting up his own website for the first time. He wants to get reliable information on how to set up this site, as well as get it configured with HTTPS. If he runs into a problem, he's likely to ask someone more experienced.

Frustrations:

He's not sure about what to expect for the process of getting a certificate. He finds the Let's Encrypt community forum overwhelming.
He doesn't know much beyond basic commands like navigating around his file system.
He doesn't know how to find what server he is running.
He doesn't know what a wildcard certificate is. He is not sure how a Certificate Authority (CA) works or what it is.
He has been told by others that Certbot is the right tool for him, and that Let's Encrypt is the right CA for him.

Goals and questions:

How do we move Titus to "Our Dream" or the "Shared Hosting" user type?
How can we help him know to copy and paste commands and errors?
How can we give some explanation as to what he's doing when he's copying and pasting commands?
How can we write a better FAQ to serve him?
How can we show how Certbot works, and what the overarching steps are? How can we communicate what to expect next?
How can we write for a beginner group?

Opportunities:

Have clear, beginner–friendly language and visual diagram that shows how Certbot works.
Have clear language for what to expect next.

JACK

Shared Hosting User

"I remember having trouble with Certbot, so if I was to do this on my own again, I'd probably try this [other service] out."

"[Web Hosting Provider Service] is super easy, for free."

Image - By Kt mac32 - Kate Richardson, Public Domain

Operating System:
On a Mac or Windows machine locally

Motivations:

Jack started making websites in the age of shared hosting and familiar with using Gitlab. He's proficient with that, can get Let's Encrypt certificates if the host provides it, but doesn't use Certbot to do so. Jack uses Dreamhost's user-facing system to do it really quickly. Jack wants a quick, easy, and inexpensive way to install certificates.

Frustrations:

He searched for answers but couldn't tell if content was out of date (e.g. looking at issues, not seeing information on relevance).
He doesn't know how to find what server he's running or what a wildcard certificate is.

Goals and questions:

How do we communicate to Jack that he has other options besides Certbot?
How can we clearly organize these other options, and point Jack to them? Should we navigate users like Jack away from using Certbot? How?
How do we assist a user who doesn't know the details of their server?
How do we boost the SEO / navigability to the list of providers? Can that be its own page?
How can we reorganize compatible providers to include mentions of updates?

Opportunities:

Give Jack a "I don't know" server option.
Direct him away from using Certbot.
Reorganize the compatible providers to include mention of updates.

KAT
The Teacher

"Certbot is great, and I love Certbot. Having free certs are important for marginalized folks who can't afford certs."

Image - By Udo Grimberg, CC BY-SA 3.0

Operating System:
Kat herself is comfortable with Linux, Mac OS and Windows environments. Students are likely just familiar with Mac OS or Windows, and are new to the following: using Nginx, Apache on Ubuntu, Cpanel.

Motivations:

Kat sets up Certbot for other people and is de facto tech support for a community. Kat is steering people toward shared hosting like Cloudflare that already has Let's Encrypt support just built in. She gives specific instructions and requires people send a template email and gets in contact with hosting providers.

Frustrations:

Kat's students might not fully understand what they're doing but they do get installed with a cert.
She thinks the terminal messages like "the expiration in 90 days..." will lose beginners.
She thinks instructions in the terminal don't really explain what's going on—her students often get confused about terminal messages, and think these messages are a step, not a recommendation.
Kat's students like numbered steps and plain english.

Goals and questions:
Can we aspire to use Simple English for documentation aimed at beginners like Kat's students? Can we breakdown the Certbot process into easier-to-understand steps?
Can we aspire to using Simple English in the command line, as well?
What can we do for users who may not know how to SSH in? Can we create a easy-to-understand diagram or over-arching explainer of how Certbot works?
How can we simplify the terminal output? Is there a way to determine what's necessary versus what's optional?

Opportunities:

Add a section that filters out people who may not know how to SSH in—saying that Certbot might not be the right solution.
Have somewhere on the Let's Encrypt site or Certbot site a visual diagram with step by step guide.
Write clearer instructions, and testing comprehension on those instructions with a beginner audience.
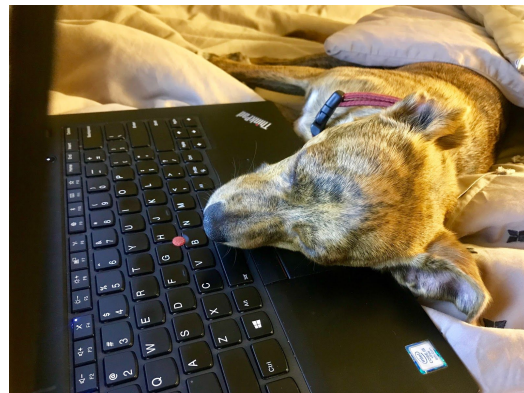Determine which terminal information from Certbot is necessary versus optional for newer users.

JORDAN

Encryption Professional

"Certbot is so easy to use! Using the standalone plugin manually every three months is so easy."

Image - By EFF - CC BY

Operating System:

Comfortable with Linux; Virtual server via service running Ubuntu 18.04; Bash or perl script + standalone

Motivations:

Jordan is a Linux expert who has a knack for finding exactly the right help pages. They get their information on how to use it from reading blog posts that say how to use Certbot with Let's Encrypt. They feel like Liriodi and Digital Ocean have the best advice. They know to go to SSL Labs. Jordan doesn't use the Let's Encrypt community forum. They are renewing their certificates manually.

Frustrations:

Jordan's colleagues have asked for their assistance in administering websites.
Jordan's colleagues are frustrated that, every three months, their website now shows a Not Secure option. They have to keep reminding Jordan to fix their site.
Jordan knows that there is a way to automatically renew their certificates, but they really prefers doing it themselves, and knowing exactly what's happening on their machines.

Goals and questions:

We want Jordan to not be manually renewing. How can we encourage automatic renewal?

Opportunities:

Put a blogpost on eff.org, urging users to please use the automatic renewal feature of Certbot.
reach out to them on Twitter.
Change the options in the command line.
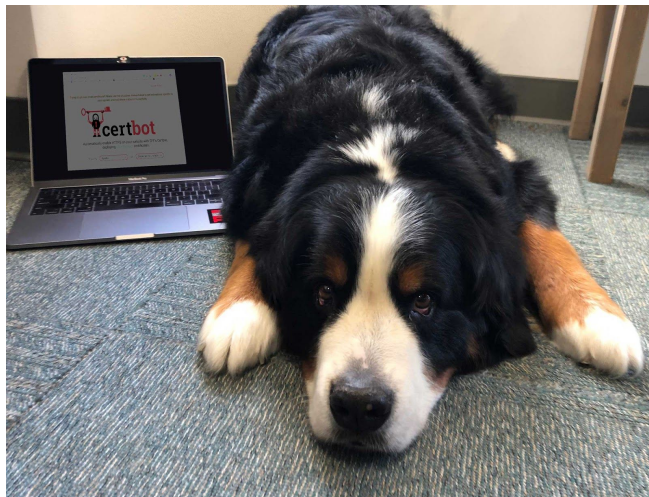Elevate the prominence of the automatic renewal option.
Make it harder to find the manual option.

SASHA

Our dream

"How do I get the green padlock that I would see at Google?"

Image - By EFF - CC BY

Operating System:
Mac OS locally, familiar with Linux, might be trying it for an unusual configuration like for Nginx on a Raspberry Pi

Motivations:

Sasha knows how to run a server and could need some hand-holding if she doesn't see a green lock when verifying that she got the result she wanted. She uses Certbot to install. She would reach out to her more experienced friends on Slack if she needed help or use Google.

Frustrations:

SSL Labs was unclear for Sasha since it was a wall of text.
Sasha's pain point is having any setup that's not the most standard common one (e.g. not Nginx or Apache).
A pain point for Sasha is that our software has a name collision with something that we reference, even if it's not in the context that we mean it.

Goals and questions:

How do we assist users with uncommon configurations?
How do we make it easier for them to find information that clearly states the environment, and when it was last updated?
Can we make a clearer, simple-to-understand version of the information output from SSL Labs? Or have some way (e.g. a legend or glossary or top tips) for users to parse that information?
Can we point to more helpful resources referenced at the Congratulations message in the terminal?
What are useful follow-up resources for beginners who have just configured their first certificate?

Opportunities:

Direct people like Sasha to whynopadlock.com.
Instead of putting out the wall of text at the end, maybe a link that says "Congratulations" that takes you to customized site with SSL and Whynopadlock result for their domain, rather than a wall of text at the end. Provide resources to point people to forums, see step-by-step instructions, donate link.

PALLAVI
The host

"I manually modified the files in the directory, and now Certbot doesn't work."

Image - By WaSu-Bio - Own work, CC BY 4.0

Operating System:
Comfortable with Linux, Mac OS

Motivations:

Pallavi runs websites for other people, either as a consultant or as part of a corporate infrastructure. She uses Nginx, Apache, or a custom version of one of those in either a distributed or single-machine environment to host multiple websites, possibly giving her clients login access. She wants certificates for multiple domains, and often finds herself having to script around Certbot to get it to do exactly what she wants, like trying to get certs onto multiple machines every two months.

Frustrations:

Pallavi wishes that Certbot were built to conform precisely to her needs—its strict directory structure and need for the `live` certificates to be in a particular place don't necessarily mesh perfectly with how she has things set up.
The Nginx and Apache plugins probably don't do what Pallavi needs, and she may have built her own plugin.

Goals and questions:

How can we motivate Pallavi to want to contribute back to Certbot?
How can we engage community members like Pallavi? Do we need someone to acknowledge and thank efforts from people like Pallavi? Should this be in the role of a community manager?
If Certbot isn't meeting her needs, we'd often love to see a patch written, if the changes she wants will be helpful to other people.
When she files an issue, we would love to turn that from being work for us to fix on her behalf to instead being work that she does to improve Certbot.

Opportunities:

Recognize when you have a user like Pallavi in the Let's Encrypt Forum or in Github, and have friendly language for them to get involved further.

Certbot Remote User Tester Prompts

"Please take a look at this site and tell us what you think it is. (You can scroll if you want but please don't click on anything yet.)
-- What's the first thing you notice?
-- What can you do on this site?
-- What services are offered on this site?
-- Who is this site intended for?

Now imagine the services offered on this site are intended for you—Imagine that you have a website. You want to get a little lock and the word "Secure" beside your website name. How would you do that?
--

If any terms or ideas are new to you, we're particularly interested in learning about what is unfamiliar or confusing. Please remember to think out loud during your test and tell us if there's any information missing.
Thank you for your help!"