PRESENTER INFORMATION

Primary Speaker Name or Pseudonym\*: Jay Lagorio

Primary Speaker Title and Company (if applicable): Independent Security Researcher

Additional Speaker Name(s)/Pseudonym(s)\* and Title(s): N/A

Have any of the speakers spoken at a previous DEF CON? If Yes, which speaker, what year(s), and which talk(s)?

Jay Lagorio:

2019/DEF CON 27: Hacking Wetware with Open Source Software and Hardware (Bio Hacking Village)

Primary Speaker Email Address: jay@lagor.io

Backup Email, in case of communication failure (optional):

Primary Speaker Phone Number:

Co-Speaker(s) Email Address(es): N/A

Co-Speaker(s) Phone Number(s): N/A

Speakers' Social Media / Personal Site Information (Twitter, Facebook, website, etc):

Twitter: @jaylagorio

Github: @jaylagorio

Web: <https://lagor.io/>

Would you like your submission sent in anonymously to the review board?: No

NEW! What do you bring to the table, how will you or your presentation(s) contribute a new perspective to the content at DEF CON?

One of my best skills is sharing technical content with audiences that include people with different levels of expertise to make the information accessible and understandable. While I consider DEF CON to be where the pinnacle of technical research is found, I also want to contribute to the development of junior- and mid-level researchers who will come up behind the seniors as the future of the community. This presentation will include a mix of reverse engineering web apps and Linux binaries, some cryptography, and some hardware work all threaded together in story-based format that I think will be easily digestible and exciting for more junior folks while taking the occasional detour to dip into the advanced technical aspects for the more senior audience members as well. I’m not a professional speaker but I’m experienced on a stage so I will be engaging, I will be well-rehearsed, and I will nail my time.

NEW! If your presentation is accepted, do you consent to inviting the primary & secondary contacts to the official DEF CON Speaker's planning project on "Basecamp"? We will use the email addresses you included on your application (unless stated otherwise). DEF CON staff, and other speakers would be able to see your email address once you accepted the invitation. Yes

PRESENTATION INFORMATION

Date of Submission: 16MAR2022

Presentation Title: Tear Down this Zywall: Breaking Open Zyxel Encrypted Firmware

Length of presentation: 45 minutes

Is there a demonstration? No

Is there audience participation? No

Are you releasing a new tool? No

Are you releasing a new exploit? No

Have you submitted or do you plan on submitting this topic to any other conference held prior to DEF CON? No

Are you submitting this or any other topic to Black Hat USA? No

Are you submitting this or any other topic to BSidesLV? No

Sometimes a good CFP makes a great fit for a DEF CON Village too. The review board will recommend some talks to be presented exclusively in a village, or as an encore to a main stage presentation. Do you consent to allow DEF CON to forward your completed submission to relevant DEF CON Village lead(s) for their consideration for village content? Yes

ABSTRACT

How do you go bug hunting in devices you own when the manufacturer has slapped some pesky encryption scheme on the firmware? Starting from an encrypted blob of bits and getting to executable code is hard and can be even more frustrating when you already know the bug is there, you just want to see it! Join me on my expedition to access the contents of my Zyxel firewall’s firmware using password and hash cracking, hardware and software reverse engineering, and duct taping puzzle pieces together. We’ll start with a device and a firmware blob, flail helplessly at the crypto, tear apart the hardware, reverse engineer the software and emulate the platform, and finally identify the decryption routine – ultimately breaking the protection used by the entire product line to decrypt whatever firmware version we want.

Note: Your abstract will be used for the website and printed materials. Summarize what your presentation will cover. Attendees will read this to get an idea of what they should know before your presentation, and what they will learn after. Use this to inform attendees about how technical your talk is, what tools will be used, what materials to read in advance to get the most out of your presentation. This abstract is the primary way people will be drawn to your session, but should not give them the entire content of the talk. We request you keep your abstract well over 140 characters, but at or under the 1337 maximum.

SPEAKER BIO(S)

Jay Lagorio, a software engineer and independent security researcher, has been building computers and networks and finding ways to break them nearly his entire life. Being a nerd that likes to dig too far into things spilled over into the real world and he accidentally became a licensed private investigator. Releaser of the occasional tool or writeup on Github, he wishes he had enough time to do all the hacker things and crush griefers in GTA Online every day. He received a B.S. in Computer Science from UMBC and an M. Eng. from the Naval Postgraduate School.

Note: This text will be used for the website and printed materials and should be written in the third person. Cover any professional or hacker history that is relevant to you and the presentation, you may include past jobs, tools that you have written, etc. Let people know who you are and why you are qualified to speak on your topic. If you prefer, you may write your bio under a pseudonym. Please include Co-Speaker and Panel Bios when possible.

REFERENCES:

Please provide a simple bibliography and/or works cited. List sources you have used (whether referenced or not) in the process of finalizing your presentation. Please remember to credit prior works and acknowledge others. References will be posted online with your talk information. We want attendees interested in your talk to be able to research what has been helpful for you in developing this presentation.

* Undocumented user account in Zyxel products (CVE-2020-29583): <https://eye.security/en/blog/undocumented-user-account-in-zyxel-products-cve-2020-29583>
* Conversation with Niels Teusink via Twitter DM to get vulnerable firmware, ask whether he’d mind if I turned my work into a talk (he said go for it, screenshots can be provided)
* Cracking a Zip File Password with John The Ripper: [https://www.golinuxcloud.com/john-the-ripper-password-cracker/#Cracking\_a\_Zip\_File\_Password\_with\_John\_The\_Ripper](https://www.golinuxcloud.com/john-the-ripper-password-cracker/)
* Crack legacy zip encryption with Biham and Kocher's known plaintext attack: <https://github.com/kimci86/bkcrack>
* Datasheet for Samsung K9F1G08U0 NAND Flash: <https://www.datasheetarchive.com/K9F1G08U0-datasheet.html>
* Datasheet for Psion PS2251-50-F USB Flash Controller: <https://www.kynix.com/Detail/246017/PS2251-50-F.html>
* Motherboard USB Pin-out diagram: <https://frontx.com/cpx108_2.html>
* Win32DiskImager: <https://sourceforge.net/projects/win32diskimager/>
* Binwalk: <https://github.com/ReFirmLabs/binwalk>
* “USGs Firmware Thread” comment about zld\_fsextract: <https://www.dslreports.com/forum/remark,26961186>
* Ghidra: <https://ghidra-sre.org/>
* QEMU Usermode Emulator: <https://wiki.debian.org/QemuUserEmulation>
* EMBA Firmware Analyzer: <https://github.com/e-m-b-a/emba>

DETAILED OUTLINE:

* Intro / whoami
* It all started with an email
* Zyxel sends a message to its customers saying it shipped a security flaw in production firmware and to update immediately, and early reporting indicates it’s a backdoor account
* Surely they couldn’t have shipped a backdoor account in a firewall?
* They did: <https://eye.security/en/blog/undocumented-user-account-in-zyxel-products-cve-2020-29583>
* It’s a good thing I save all my previous firmware downloads, let’s look at the bug
* Uh oh, it’s encrypted but I still want to look at it. No amount of Google-fu reveals any information at all into the extraction process. Now what?
* Firmware file analysis
* The firmware is a password protected zip file with some extra bytes on the end; you don’t need to enter the password in the normal device upgrade flow.
* Some files in the archive might be the same as files available on the Internet, so let’s try an open source tool to crack the compressed file
* <https://github.com/kimci86/bkcrack>
* Hash cracking didn’t work for whatever reason, so let’s try brute-forcing the password for a bit while we think of other ideas
* Tried a bunch of default passwords available around the Internet
* [https://www.golinuxcloud.com/john-the-ripper-password-cracker/#Cracking\_a\_Zip\_File\_Password\_with\_John\_The\_Ripper](https://www.golinuxcloud.com/john-the-ripper-password-cracker/)
* A day or so goes by while I was brainstorming and no password was found
* This is a chicken and egg problem: In the firmware is the software that can decrypt the firmware, but I don’t have the software outside the encrypted firmware to use it. Or do I?
* Hardware Teardown
* I have a previous model Zywall from the last upgrade lying around, let’s crack the case
* Storage on a separate board – it’s built weird, but it’s actually a secret USB disk! (pictures will be included)
* <https://www.datasheetarchive.com/K9F1G08U0-datasheet.html>
* <https://www.kynix.com/Detail/246017/PS2251-50-F.html>
* <https://frontx.com/cpx108_2.html>
* Cut a USB cable in half, throw wires into the socket and plug it in (pictures will be included)
* The disk reads, so image the disk with Win32DiskImager (<https://sourceforge.net/projects/win32diskimager/>)
* Throw the image in binwalk for extraction (<https://github.com/ReFirmLabs/binwalk>)
* Profit! We have the plain-text firmware that was loaded on this device, but what about decrypting arbitrary firmware files?
* Trace the Upgrade Process
* What happens when you want to update the firmware? You upload the file to a web page on the device and it does the rest
* Device runs an Apache web server, so look at the static page
* Static page submits to firmware\_upload-cgi in cgi-bin, run strings
* Contains string "/util/zld\_fsextract" – “file system extract?”
* Get that file, throw the file name into Google
* One single result and no binary: <https://www.dslreports.com/forum/remark,26961186>
* Run the file command to see what kind of executable it is (MIPSN32)
* Run strings to see if there’s anything interesting
* A usage string
* 7za
* /tmp/\_\_XXX\_\_7zip\_listfile
* Confirm code paths and see if we can find any other interesting points with Ghidra ([https://ghidra-sre.org](https://ghidra-sre.org/))
* Found the reference to run 7za -p (specifying password)
* It’s now clear that this binary is running 7zip to decrypt and decompress, and it would have to determine the password and pass that to 7zip. But my PC doesn’t run MIPS.
* Emulation to the Rescue
* Rather than virtualize a whole MIPS system or try to run the firmware itself, use user-mode emulation on Linux to run the single binary
* Method of choice: QEMU: <https://wiki.debian.org/QemuUserEmulation>
* Run by itself to see if it spits out the usage, then try to run as specified with syscall tracing
* Syscall tracing reveals the read of the extra bytes from the end of the file
* Also reveals the parameters sent to execve to run 7za, including the password!
* Tracing variables back using Ghidra, we can identify the routine that uses the extra bytes on the end of the firmware to generate the decryption password
* How cool would it be to use binfmt to run MIPS binaries transparently to the user cross compile any tools needed to interact with it, and script it more easily?
* Very cool, except that this doesn’t work because binfmt isn’t supported on Windows Subsystem for Linux for some reason
* Turns out, by writing a simple C one-liner and compiling for the native architecture, you can have it spit the password out to the console by replacing 7za with the resulting binary
* This binary is from the previous model, how do I know it’ll work on firmware for newer models?
* The devices use the same processor architecture, firmware formats between model revisions appear the same (encrypted zip, trailing bytes)
* There’s no information whatsoever on the Internet as to how to decrypt these files – from the vendor perspective, if it ain’t broke, why fix it?
* Spoiler: the newer models have the same binary so this will work just fine
* Now you can feed the extractor any firmware file and it quickly outputs the password for easy extraction with 7zip. Success!
* What About the Original Bug?
* It turns out my copy of the buggy firmware was corrupted, and Zyxel memory-holed the vulnerable versions from the Internet – seriously, that stuff is all gone somehow
* So I DM’d the researcher (Niels Teusink) to ask if he wouldn’t mind sending me a copy of his firmware, and he did! (screenshots provided)
* Using the now very quick process above, I was able to extract the firmware
* Since we already knew the IOCs I grepped for them to find the resulting files
* Results:
* A backdoor user in the /etc/passwd file with a hard-coded password
* The password, while very strong, was hard-coded into a binary on the device in plain-text
* Future Work
* Turn the deobfuscation algorithm into a Python script so you can just point it at a firmware file and get the password (and shell to 7zip if you want)
* Knowing the algorithm also means you’ll be able to generate your own firmware updates and root the device because you’ll know what password to encrypt your zip file with and how to make the stock firmware update process derive that password for extraction
* Or you can just cheat! Set the password on your encrypted zip file to the same password from an existing firmware, append the trailing bytes from that file to your file, and there’s your update package
* Run firmware through Emba
* I’m not a bug hunter, but if I was I would want something automated to churn through firmware files because at the end of the day I’m a lazy developer just like every other developer
* Once the firmware files are decrypted you can run them through EMBA, an automated firmware analysis platform (<https://github.com/e-m-b-a/emba>)
* Started out as a command-line only tool, but it’s recently gotten a really sweet web front-end that you run locally
* From our process above, it can do everything after the unzip process including automated binwalk, check against known CVEs, and even emulation of the target binaries for the whole spectrum of static and dynamic analysis
* It generates a slick report at the end that shows everything it found and boy can it find a lot.
* Track whether and when Zyxel makes a breaking change to this process
* They’ll have to update existing devices to understand the new upgrade mechanism first, and those updates will be vulnerable to this process
* You’ll notice language like “you have to install version X before you can install version Y” in future updates because firmware before version X won’t know how to handle the new update mechanism and firmware released after won’t conform to the old scheme
* You break that “transition” version open using what we already know and trace its upgrade process to figure out how you’re going to extract firmware for version Y and beyond
* Questions and Acknowledgements
* Happy Bug Hunting!

Note: This is the most important section on the application. You must provide a detailed outline containing the main points and navigation through your talk - show how you intend to begin, where you intend to lead the audience and how you plan to get there. Your outline should be in simple text. Please do not submit slides, Docs, or PDFs as an outline. If you are submitting a panel, it’s encouraged to list what each panel member will contribute to on the outline, as well as the estimated time budgeted for those bullet points. The review board likes submissions that include references to prior works and research you used in developing your presentation. The more detailed your outline then the better we are able to review your presentation against other submissions (and the higher chance you have of being accepted).

Good Outlines are discussed here:

<https://writingprocess.mit.edu/process/step-2-plan-and-organize/creating-detailed-outline>

<https://www.defcon.org/html/links/dc-speakerscorner.html#nikita-cfp>

<https://www.defcon.org/html/links/dc-speakerscorner.html#leah-cfp-process>

BAD outlines look like these:

i. intro

ii. something

iii. something else

iv. conclusion

v. q&a

WHY DEF CON?:

When COVID hit I was sent home while my office got its plan together. I wasn't able to return to the office for quite some time, so with a lot of unexpected free time, I was able to really dig into this project and cover several technical disciplines in the process. DEF CON being virtual in 2020 was a real high point in a bad time, and while I wasn’t able to assemble this talk to submit for the hybrid 2021 event for several reasons I wanted to give something interesting back to the community that helped keep me sane as we gather back together again.

I put this talk together in the spirit of the kind of DEF CON talk my friends and I like to see: rather than focus on a single technical topic the whole time, this talk weaves its way through an involved process that crosses multiple subject areas along the way and ends in cracking open an entire product line’s proprietary firmware. It includes discussion of the dead ends and stretch goals that didn’t work out to maintain the realism of the research process. If it catches someone’s interest, the end includes follow on work and next steps people can use to build their skills using this as a starting point. My hope by the end is for there to be people who will now be able to do what they know (hunt bugs) since I’ve done the first step and shown them something they may not have known (how to bypass the protection mechanism keeping them from even getting started).

Note: This is your opportunity to directly address the DEF CON review board as to why you think your talk is good for our hacker con. More and more we are seeing submissions that are well suited for cyber security industry conferences, developer conferences, or local meetups but may not align with the board's view of the DEF CON spirit. The bumper sticker "Keep Infosec out of Hacking" comes to mind. We get many great submissions but they aren't always aimed at the 30,000 hackers at our con. This frequently results in the review board saying "Unfortunately, this is not a DEF CON talk". DEF CON is a hacking con, so how does your content fit in to that culture, spirit, and subject matter?

SCHEDULING AND EQUIPMENT REQUIREMENTS

Is there a specific day or time by which you must present? No

Will you require more than 1 projector feed? No

Are there any other special equipment needs that you will require to successfully present your talk? No

SUPPORTING FILES:

Note: Additional supporting materials such as code, white papers, proof of concept, etc. should be sent along with your email submission. Additional files that may help in the selection process should be included. We are not asking for a complete presentation for this initial submission and full slides will only be required if you are selected for presenting. It is the submitter's responsibility to remove any PII from any attached slides, white papers, or supporting materials, and to appropriately sanitize any metadata in the provided content.

SUBMISSION AGREEMENTS

Please read and accept these terms by inserting your name where noted. Failure to do so will render your submission incomplete. Please read these carefully as some of the terms have changed.

Grant of Copyright Use

I warrant that the above work has not been previously published elsewhere, or if it has, that I have obtained permission for its publication by DEF CON Communications, Inc. and that I will promptly supply DEF CON Communications, Inc. with wording for crediting the original publication and copyright owner. If I am selected for presentation, I hereby give DEF CON Communications, Inc. permission to duplicate, record and redistribute this presentation, which includes, but is not limited to, the conference proceedings, conference CD, video, audio, and hand-outs to the conference attendees for educational, on-line, and all other purposes.

Terms of Speaking Requirements

1) I will submit a completed presentation, a copy of the tool(s) and/or code(s), and a reference to all of the tool(s), law(s), Web sites and/or publications referenced to at the end of my talk and as described in this CFP submission for publication on the DEF CON media server, to be released the day of the conference, by 12:00 noon Pacific time, July 15, 2021.

2) I will submit a final Abstract and Biography for the DEF CON website and Printed Conference Materials by 12:00 noon Pacific time, June 15, 2021.

3) I understand if I fail to submit a completed PDF presentation by July 15, 2021, I may be replaced by an alternate presentation or may forfeit my honorarium. This decision will be made by DEF CON and I will be informed in writing of my status.

4) I will include a detailed bibliography as either a separate document or included within the presentation of all resources cited and/or used in my presentation.

5) I will complete my presentation within the time allocated to me - not running over, or excessively under the time allocation.

6) I understand that DEF CON will provide 1 projector feed, 2 screens, microphones, wired and/or wireless Internet. I understand that I am responsible for providing all other necessary equipment, including laptops and machines (with VGA output), to complete my presentation.

7) If applicable, I will submit within 5 days of the completion of the conference any updated, revised or additional presentation(s) or materials that were used in my presentation but not included on the conference media server or conference proceedings.

Terms of Speaking Remuneration

1). DEF CON will provide 4 nights hotel per accepted presentation for the primary speaker only. The hotel will be at the DEF CON Venue properties, and of DEF CON’s choosing. I understand I will need to confirm my hotel nights and submit my preferences by the date listed in my official acceptance letter. I understand that I will be responsible for my own travel expenses, unless prior approval is made with special exception.

2) I understand that DEF CON will issue one $300 payment per presentation to the primary speaker only. Payment will be made in the form of company check. I may choose to waive my $300 honorarium in exchange for 3 DEF CON Human badges, received at the start of the conference. I may also choose to donate my honorarium to charity.

3) I understand that I may receive payment on-site at the conference. If selecting the $300 payment as my honorarium, I must provide a valid name and postal mail address so that the payment may be mailed. In some rare cases, I may be required to complete a W8 (Non-U.S. Citizen) or W9 (U.S. Citizen) before payment is issued.

4) I understand that I will be paid within 30 days from the end of the conference, after I have completed my presentation. I understand that should my talk be determined to be unsuitable (e.g. a vendor or sales pitch, a talk on the keeping of goats, etc.) after I have presented, that I will not receive an honorarium.

As detailed above, I, (insert primary speaker name), have read and agree to the Grant of Copyright Use. I, (insert primary speaker name), have read and agree to the Terms of Speaking Requirements. I, (insert primary speaker name), have read and agree to the Agreement to Terms of Speaking Remuneration or I will forfeit my honorarium.

PRESS CONTACT

I, (insert primary speaker name) understand that DEF CON's official Press Liaison & Staff may contact me. I consent to be contacted in order to arrange interviews with the media. My contact information will not be given to third parties without my consent.

< OR >

No, I (insert primary speaker name), don't want to be contacted by DEF CON's press staff for any reason. Our policies, including our privacy policy are located here: <https://www.defcon.org/html/links/dc-policy.html>