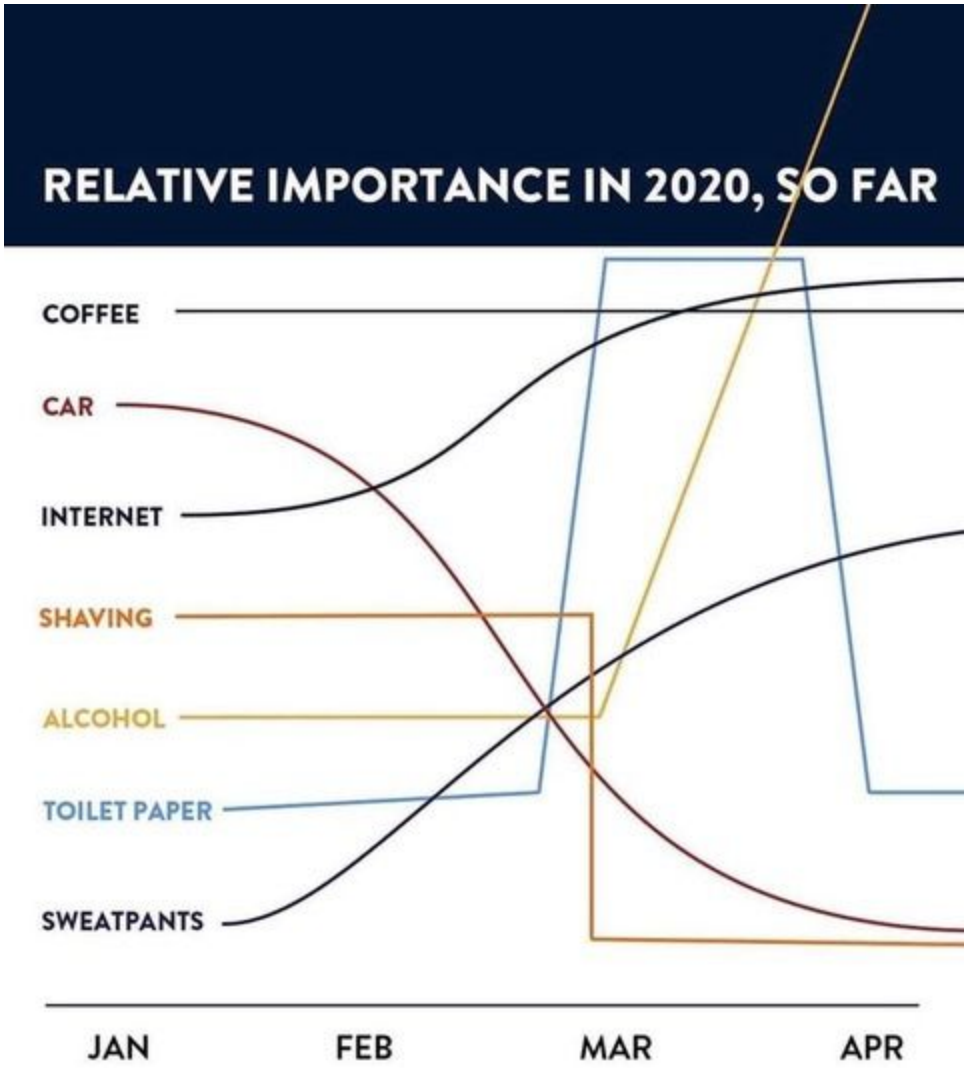
Security Now! #770 - 06-09-20

Zoom's E2EE Debacle

This week on Security Now!

This week we take an interesting new look at some new problems arising with DoH, we look at IBM's new stance on facial image recognition research, we look at two recently disclosed flaws in the Zoom client, we check on the severity of the latest UPnP service flaw and we update on Microsoft's new Edge rollout. We share a bit of miscellany and some terrific feedback from our listeners, touch on my SpinRite project progress, and then we explore last week's truly confusing Zoom encryption reports that give the term "mixed messaging" a bad rap.

"It's been a weird year"



Browser News

The Odd Case of Mozilla's DoH DDoS

The Mozilla team was busy last week. On Tuesday they released Firefox 77 for the desktop Windows, MacOS and Linux. It fixed the typical security issues. Five vulnerabilities received a high-severity score, with three of them allowing bad actors to run arbitrary code on vulnerable installations. As Mozilla puts it, a flaw classified as high in severity "can be used to gather sensitive data from sites in other windows or inject data or code into those sites, requiring no more than normal browsing actions". So those got fixed. #77 also furthered the rollout of Mozilla's WebRender project. WebRender is a new and emerging RUST-based 2D graphic web rendering system for NVIDIA-equipped Windows 10 laptops having screens of any size.

However, shortly after the release of FireFox 77 things quickly went sideways. Taking a piece of this week's listener feedback out of sequence, we have:

> Chris Miller @Mil_Fi
> *Hello Steve, long time listener of Security Now. Just want to let you know about something. I work for a fairly large county government. We have all internal users (6,000+) go through a proxy server for security purposes. Well our proxy was overwhelmed yesterday with anything going through it. It turned out that DoH was automatically enabled on just a few of those users who had upgraded to FireFox 77 (fewer than 10) and it completely crippled us. FireFox is not a browser many use in our environment, either.*
>
> *I also read the links below and saw that DoH basically overwhelmed NextDNS, the secondary provider in FireFox. Now, Mozilla appears to be slowing down its rollout tremendously. I guess the load is so much more on existing systems. Just thought you'd like to know. I am sure other enterprises will be experiencing a similar issue.*

And, indeed, Chris was right. So right, in fact, that the rollout of the ill-fated FireFox 77 was immediately halted and replaced by 77.0.1, which is what anyone who is current will now have.

So what happened?

The exact details are surprisingly thin. Over on Bugzilla, Mozilla's bug tracking site, only two somewhat cryptic explanations are found:

- "*Disabled automatic selection of DNS over HTTPS providers during a test to enable wider deployment in a more controlled way.*"

- "*We need to be able to roll this out gradually so that we don't overload any providers. Even the dry-run involves up to 10 requests per client which can be very significant when the entire release population updates.*"

Here's one thing that may be going on: Web servers are not super-happy with long duration persistent connections of the type that DoH defines and requires for performance intended to compete with traditional UDP. I ran across exactly this problem a few years ago at one phase of the SQRL project.

We wanted the site's login web page to automatically update once the user had logged on either optically with their phone seeing the unique QR code on the page, or with a SQRL client installed into the same machine. There were two ways this could be done from within a web page: Either bring up one persistent connection by having the page connect back to the server and wait for the signal to refresh the page, or sit in a loop periodically and continuously probing the server to ask whether the page should be updated.

I initially took the first approach, since that seemed much cleaner: Setup and camp out on one connection and wait for word from the server. I brought that solution online and the gang in the GRC SQRL group began playing with it. Then I quickly came to appreciate just how much web servers are designed to be inherently transactional. They want to field short-lived connections, return the data, and hang up the connections. Having many connections is no problem, but they should be coming and going rapidly. In this case the long-term static connections were making GRC's web server VERY unhappy.

Back in the early days of this podcast we talked about the older style of DoS attack (not DDoS, just DoS) where a single low-bandwidth attacker could bring down a beefy server simply by sending it a stream of TCP SYN packets. Every incoming SYN packet was a request to establish a new TCP connection. So the server would jump to action, allocate some resources to manage the nascent connection. It would record the sequence number provided by the remote client in itssss SYN packet, generate its own sequence number for its own transmissions, record any other connection-specific details provided by the caller, then generate and send its own answering SYN/ACK packet and wait for a reply. If no answering ACK was received back, it would assume that the reply was lost, so it would retry that send several times. The point is, ALL of that effort and allocated resource was forced upon the server by someone simply and mischievously sending it a stream of short and simple SYN packets.

In my case, everyone who was sitting at a SQRL login page had established a persistent connection to my web server, and it was being seriously overburdened. So I changed the login page's logic to issue the equivalent of a “TCP ping” -- a query for an named object that would immediately generate a reply and disconnect... and I never had another problem since that's what web servers are designed to handle.

The potential trouble with DoH is that it, too, replies upon the maintenance of a persistent static TCP/TLS connection across which occasional DNS queries transit. As our web browsers begin using DoH, **every single browser that's open** will have a static TCP/TLS connection established to its chosen DoH provider. I sure hope that this has been given due consideration by those who wish to move DNS over to TCP. The traditional DNS that we’ve always been using is the lightest weight query we know how to make: a single isolated UDP packet. So no matter what, we **are** heading toward a solution that is a great deal more burdensome on DNS providers than UDP has ever been.

Security News

IBM announces no more work on facial recognition

IBM's CEO, Arvind Krishna sent a letter to Congress by way of Axios and CNBC. The letter stated that the company has willfully exited its "general purpose" facial recognition business. Arvind's letter said:

> IBM no longer offers general purpose IBM facial recognition or analysis software. IBM firmly opposes and will not condone uses of any technology, including facial recognition technology offered by other vendors, for mass surveillance, racial profiling, violations of basic human rights and freedoms, or any purpose which is not consistent with our values and Principles of Trust and Transparency. We believe now is the time to begin a national dialogue on whether and how facial recognition technology should be employed by domestic law enforcement agencies.
>
> Artificial Intelligence is a powerful tool that can help law enforcement keep citizens safe. But vendors and users of Al systems have a shared responsibility to ensure that Al is tested for bias, particularity when used in law enforcement, and that such bias testing is audited and reported.
>
> Finally, national policy also should encourage and advance uses of technology that bring greater transparency and accountability to policing, such as body cameras and modern data analytics techniques.

CNBC noted that it's relatively easy for IBM to back out when facial recognition wasn't a major contributor to its bottom line. The media buzz may be as important as anything. IBM is still a major company, though, and it frequently works with governments. This could spur other providers to follow suit, and might even get some would-be customers to drop facial recognition entirely.

While some facial recognition systems may only correlate faces with publicly available data, there are concerns this could be used for tracking and profile generation that could be used to intimidate people or otherwise limit their real-world privacy.

I'm delighted by the attention this topic is generating. As we know, just because we CAN do something doesn't mean that we should.

Cisco's Talos group found two critical flaws in the Zoom client

We'll be talking about last week's mega Zoom policy issues at the end of this podcast. But in the meantime you'll want to be sure you're running the latest client.

The cybersecurity researchers in Cisco's Talos group revealed last Wednesday that they had discovered two critical vulnerabilities in the Zoom software that could have allowed attackers to remotely hack into the computer being used by group chat participants or an individual recipient. Since both flaws are -- wait for it -- path traversal vulnerabilities -- that can be exploited to write or plant arbitrary files on the systems running vulnerable versions of the video conferencing software to execute malicious code.

According to the researchers, successful exploitation of either flaw requires no or very little interaction from targeted chat participants and can be executed simply by sending specially crafted messages through the chat feature to an individual or a group. So, essentially, it was a built-in remote nuke.

The first security vulnerability (CVE-2020-6109) resided in the way Zoom leverages the GIPHY service which was recently acquired by Facebook to allow its users to search and exchange animated GIFs while chatting.

Researchers found that Zoom was not checking whether or not a shared GIF was loading from Giphy service. This allowed an attacker to embed GIFs from a third-party attacker-controlled server, which are cached and stored on the recipients' system in a specific folder associated with the application. And since the Zoom application was not sanitizing the filenames, attackers could perform directory traversal to save malicious files disguised as GIFs to any location on the victim's system, for example, the startup folder.

The second remote code execution vulnerability (CVE-2020-6110) resided in the way vulnerable versions of the Zoom application process code snippets shared through the chat.

Cisco wrote: "Zoom's chat functionality is built on top of the XMPP standard with additional extensions to support the rich user experience. One of those extensions supports a feature of including source code snippets that have full syntax highlighting support. The feature to send code snippets requires the installation of an additional plugin but receiving them does not. This feature is implemented as an extension of file sharing support."

This feature creates a zip archive of the shared code snippet before sending and then automatically unzips it on the recipient's system. And, according to the researchers, Zoom's zip file extraction does not validate the contents of the zip archive before extracting it. This allows the attacker to plant arbitrary binaries on targeted computers.

Cisco wrote: "Additionally, a partial path traversal issue allows the specially crafted zip file to write files outside the intended randomly generated directory." This bypasses a mitigation that Zoom had put into place.

The researchers tested both flaws on version 4.6.10 of the Zoom client application and responsibly reported it to the company. Last month, Zoom patched both critical vulnerabilities with the release of version 4.6.12 of its video conferencing software for Windows, macOS, or Linux computers.

As we have often observed, anyone can make a mistake. These were just a pair of them and Zoom fixed them quickly upon being notified. We might wish for perfect software, but no one is willing to pay what that would cost. So instead we muddle through with software that mostly works and fix problems as they come to light. It's not a perfect system, but it's the one we have.

Is a Stranger Calling?

Yesterday, the Internet's tech press blew up over the disclosure of an newly discovered vulnerability in UPnP (big surprise). Headlines took the form of:

ZDNet: “CallStranger vulnerability lets attacks bypass security systems”

BleepingComputer: “CallStranger UPnP bug allows data theft, DDoS attacks, LAN scans”

Tenable: “Universal Plug and Play (UPnP), a ubiquitous protocol used by “billions of devices,” may be vulnerable to data exfiltration and reflected amplified TCP distributed denial of service (DDoS) attacks.” Whoa!

Since our listeners may be encountering these dire and breathless warnings in coming days I wanted to take a moment to explain what's going on -- and what isn’t -- especially because most of the coverage is as unclear as the vulnerability's website which, naturally and predictably, makes quite a big deal about it:

https://www.callstranger.com/

Remember that the early problems with UPnP arose from the fact that sample implementation code, made available by Intel back in 2000, was quite clearly marked "Sample" and was never intended to be implemented in the field. Yet many if not all vendors grabbed that source code, compiled it for their chipsets, added it to their routers, and added a "UPnP Compatible" bullet point to their router's features list. And so, during the 20 years since, we've had many problems that are directly traceable back to that original publication of never-claimed-to-be-ready-for-use code.

Our own episode #389 resulted in me adding a public UPnP exposure test to GRC's ShieldsUP! facility and, since then, 54,954 tests have come back positive for public UPnP exposure.

The good news is, this current "CallStranger" bug is not anything like a public exposure problem. Rather, this is someone who did legitimately discover a client-side server flaw in the LAN-facing side of UPnP, which IS indeed present in most current UPnP implementations

The problem is that most implementations of UPnP including Windows 10 and almost certainly all Windows versions including servers (upnphost.dll), Xbox One, and devices from Asus, Belkin, Broadcom, Canon, Cisco, D-Link, Epson, HP, Huawei, NEC, Philips, Samsung, TP-Link, TrendNet, Zyxel and probably just about everything else. Notably there are known UPnP stacks, such as "miniupnp" after 2011 that are not vulnerable. SO not absolutely everything.

CERT's title for this is probably the most sane: “Universal Plug and Play (UPnP) SUBSCRIBE can be abused to send traffic to arbitrary destinations.”

> A vulnerability in the UPnP SUBSCRIBE capability permits an attacker to send large amounts of data to arbitrary destinations accessible over the Internet, which could lead to a Distributed Denial of Service (DDoS), data exfiltration, and other unexpected network behavior. The OCF has updated the UPnP specification to address this issue. This vulnerability has been assigned CVE-2020-12695 and is also known as Call Stranger.

So there are two problems:

By far the largest problem, because it affects potentially all UPnP-hosting devices on a LAN, which would include our routers and printers and probably anything similar that offers services, perhaps even LAN-attached IP cameras and so on, is that to do their job they are exposing LAN-facing HTTP servers. And it's this server in which a highly prevalent problem has been found.

But notice that this is all LAN-facing. In other words, an attacker need to already have some foothold inside the network to be able to abuse this aspect of UPnP. This is why in all of this coverage we see references to DLP -- data loss prevention -- because any of these UPnP devices could be turned into a proxy and used to route an attacker's exfiltration traffic out of the Intranet possibly bypassing corporate level anti-exfiltration DLP security. But, again, this only happens with an attacker who is already behind the firewall.

The related question is what about the WAN-facing interface of UPnP devices? That's what ShieldsUP! was enhanced to detect, but only for one specific vulnerable aspect of UPnP. What no one has made clear is whether this SUBSCRIBE vulnerability exists on the WAN-facing side of UPnP devices and, if it does, what that would allow attackers to do. In the best case, attacks would be limited to using the exposed UPnP device for various forms of reflection attacks. In the worst case it would allow a remote attacker to probe through the device and into the LAN behind it. At this point there's no clarity for that. Among the potential problems created by this, the researcher does state "Scanning internal ports from Internet facing UPnP devices", so that's certainly not good.

But the LAN-vulnerability side is clearly a problem for IT departments. To help with that, the researcher who found the problem has posted a Python-based vulnerability scanner on GitHub:

https://github.com/yunuscadirci/CallStranger

He explains that the script performs a series of actions:

Finds all UPnP devices on LAN
Finds all UPnP services
Finds all subscription endpoints
Sends these endpoints, encrypted, to a verification server via UPnP Callback.
Server can't see the endpoints because all encryption is done on client side
Gets encrypted service list from verification server and decrypts on client side
Compares found UPnP services with verified ones

I was a bit disturbed by the line “Sends these endpoints, encrypted, to a verification server via UPnP Callback.” and “Server can't see the endpoints because all encryption is done on client side”. That may be true, but assuming that the “server” is on the public Internet, it certainly sees the public IP that’s sending it those queries. Since this was somewhat disturbing I decided to fire the script up on my own network to see what was going on.

Upon running it, I received this:

```
Stranger Host: http://20.42.105.45
Stranger Port: 80
No UPnP device found. Possible reasons:
* You just connected to network.
* UPnP stack is too slow. Restart this script.
* UPnP is disabled on OS.
* UPnP is disabled on devices.
* There is no UPnP supported device.
* Your OS works on VM with NAT configuration.
```

The “Stranger Host” line does, indeed, appear to indicate that anyone running this script will be sending stuff to that public IP: 20.42.105.45.

But even so, my results appeared to be all negative. Since I was assembling this podcast I didn't want to spend too much time digging into it, but I was skeptical that my network had ZERO UPnP devices. So I enabled the UPnP service on my FreeBSD-based pfSense firewall and ran a different UPnP scanner. It found the UPnP service there and provided a disturbing amount of information. Among that information was the happy news that pfSense uses the “miniupnp” implementation that is known to be unaffected by this flaw.

I then re-ran the Python script and it still found nothing. But given the horrifying example he shows on GitHub, his script still might be worth running. Or if it’s possible to do this locally, as I strongly suspect is the case, then wait for one that does that.

So we are where we always find ourselves after something like this. Our networks have connected devices, many of which will likely never be fixed. None of the known problems are critical enough to force us to disconnect them, but we're left with a mild discomfort that things are not as secure as we would like them to be.

The lesson this continues to reinforce is that anything that connects needs to have a means for being updated as problems are discovered. Everything device should have some sort of home, and everything should periodically phone that home to check for any important updates. In other words, updating is every bit as important as connecting. One should not happen without the other. We're not there today, but that's where we need to aim.

Microsoft has started replace old Edge with new Edge

At long last, the Chromium-based Edge web browser is being rolled out automatically through Windows Update to everyone using Win10 1803 or later. It will be identified as KB4559309 and will replace the original Edge browser on all Win10 2004, 1909, 1903, 1809, and 1803. (And isn't it unifying that we only have Windows 10, now.)

Interestingly, whereas the original Edge could be removed, Microsoft says: "The new Microsoft Edge does not support removal." and also "The current version of Microsoft Edge will be hidden from UX surfaces in the OS. This includes settings, applications, and any file or protocol support dialog boxes" and any "attempts to start the current version of Microsoft Edge will redirect to the

new Microsoft Edge."

Of course all previous user data from earlier Microsoft Edge versions; passwords, bookmarks, open tabs, etc. will be moved over into the new Edge.

And, separately, there was also some reporting that the Win10 Start Menu had begun advertising the new Edge whenever anyone appeared to be searching for information about any other web browser. I very much like the idea of a Chromium-based Edge. And I suppose it's Microsoft's right to push whatever they wish to, since they own the platform. But it certainly doesn't give me -- nor those who were reporting on this -- any warm and fuzzies.

Listener Feedback

Luis Cruz @sprak
Barring the ability to transfer your consciousness to a new vessel, do you have a plan for SpinRite and your other work after you are incapable of maintaining it? Will it go open source? Are you mentoring Gibson 2.0 behind the scenes to pick it up?

SKYNET @fairlane32
Hi Steve, in ep 769 with automatic downloads in chrome, you have the option of choosing “Ask every time” under Chrome’s settings. Wouldn’t that prevent the drive by downloads on sites?

[Yes!, good point... but it's not the default and it's buried under Privacy & Security then "Advanced"]

Chris Rhodus @chris_rhodus
Hi Steve! Over the years I have heard you say that there is no reason to limit the maximum size of a password. I’m currently reviewing a vendor design document that has the password max limit set to 32 characters. Are there any case studies or other documents you can point me to that can be used to justify the removal of the 32 character limit. I don’t think the vendor will be happy about removing the max limit. I will need to justify the removal of the imposed limit if any opposition is encountered. Thanks!

Ed McKiver @OhWellDamn2010
Steve, just a passing note to add to other comments you might be getting... My Dell laptop auto-updated to Win-10 rel. 2004. Ever since my computer has been running EXTREMELY SLOW and sometimes clicking on Windows items or right-click menu takes minutes to come up. Went into settings to check for updates and the Settings window crashed during the check before finishing. Second attempt completed ok, but other items like ‘View update history’ is taking forever to display. Click and wait and wait and wait is the order of the day now... I have 150 GB of free space on my HD. My laptop tends to slow down when I get under 100 GB but it’s always worked normally with at least 100 GB of free space. Thought I’d put in my 2C for stopping all this major release stuff to Windows if they are constantly going to break stuff. Oh!, settings window just crashed again as it was just trying to ‘View update history’... I can’t even roll back at this point. FYI. Ed in Redlands and SpinRite owner since Ver.1.

Miscellany

- https://grc.sc/states (https://www.endcoronavirus.org/states)
- https://grc.sc/countries (https://www.endcoronavirus.org/countries)

SpinRite

I don't have a lot of exciting news to share on the SpinRite project. We found a few remaining problems with the FAT partitions I was creating with the size of their root directories. That's fixed and has been well torture tested by the gang in the newsgroup. It was suggested that we needed a completely fail safe means for selecting the drive whose contents we intended to permanently wipe out and destroy through reformatting. So I added the technology for InitDisk to watch all of the system's drives after asking the user to confirm the drive they wish to use by physically removing and then replacing it. That worked quite well. But it worked so well that I then wanted to see how it would be if that were the only means we used for specifying the drive to reformat and use. So that's what I'm currently working to bring online.

It so nicely solves other problems, too, such as using a drive from Linux or Mac that's been formatted with a file system that Windows doesn't mount. Such a drive would not have a drive letter to use for specifying the drive to use. Previously, I solved this by allowing the user to specify the Windows physical disk number as displayed by Windows Disk Management. But that's asking a lot of the casual user. Another problem were the SanDisk USB drives that were made to be Windows 8 Certified by appearing to be a fixed drive rather than a removable drive. (That policy requirement was short lived, but many such drives exist.) If it's interface is USB and it can be removed and reinserted on demand, it's removable. So, inserting the target drive while InitDisk watches very nicely solves the "this is the device I want to use" problem. And once the candidate drive is present, InitDisk displays a bunch of information about the drive for additional confirmation.

So we're just about at the end of this particular sub-project and we'll have some very nice technology for all of GRC's subsequent bootable projects.

Zoom's E2EE Debacle

Yes, just when everything appeared to be going so well... the day after our podcast last week which celebrated the quite rational and well supervised planned evolution of Zoom's security architecture, during a call with financial analysts to discuss Zoom's latest financial results, CEO Eric Yuan confirmed that Zoom won’t be offering end-to-end encryption on free accounts. What?!?!

Eric was widely reported to have said, and I quote from multiple sources: “Free users, for sure, we don’t want to give that [end-to-end encryption]. Because we also want to work it together with FBI and local law enforcement, in case some people use Zoom for bad purpose.”

Oh boy.

Not surprisingly, my Twitter feed lit up with our listeners asking whether I had seen this latest from Zoom? ... and many saying that that's it... "no way" are they going to be using it from now on. <sigh>

To give everyone a sense for the industry's reaction to this revelation:

- The Verge: "Zoom says free users won't get end-to-end encryption so FBI and police can access calls."

- The Guardian: "Zoom to exclude free calls from end-to-end encryption to allow FBI cooperation."

- Engadget: "Zoom explains why free users won't get end-to-end encrypted video calls."

- USA Today: "Zoom CEO: No end-to-end encryption for free users so company can work with law enforcement."

- The Next Web: "Zoom won't encrypt free calls because it wants to comply with law enforcement."

- Tech Crunch: "Zoom faces criticism for denying free users e2e encryption."

... and I could keep going, but everyone gets the idea.

The problem, of course, is that this is seen as purely a profit motivated policy, since strong end-to-end encryption is desirable and only paying customers can get it. And the argument about compliance with law enforcement? "So, you want to allow Zoom to comply with law enforcement... except if people pay for the service... in which case Zoom's newer and better end-to-end encryption will explicitly not allow for any compliance with law enforcement?"

So, you're making money by marketing your hostility to law enforcement. Of course, we've seen that before. That's the stance that Apple has explicitly and loudly taken. But Apple's encryption is ubiquitous. Apple doesn't allow anyone to NOT have full end-to-end encryption on any of their person-to-person connections -- text, voice or video.

I was as stunned as anyone by this news, especially given the mature supervision that Zoom was now receiving. But the CEO had spoken, and on this he seemed quite unambiguous. So I thought I'd reach out to Alex Stamos to see what he might know. I brought him up in TweetDeck and discovered that he follows me on Twitter. So that meant I could DM him directly and hopefully not be lost in the noise.

I shot Alex a DM to make sure that he was aware of this mess, then I continued poking around and quicky discovered that he was quite well aware indeed. He had posted a series of Tweets that attempted to repair the damage. But, frankly, they only further complicated things.

His tweet stream read: https://twitter.com/alexstamos/status/1268061790954385408

Some facts on Zoom's current plans for E2E encryption, which are complicated by the product requirements for an enterprise conferencing product and some legitimate safety issues.

All users (free and paid) have their meeting content encrypted using a per-meeting AES256 key. Content is encrypted by the sending client and decrypted by receiving clients or by Zoom's connector servers to bridge into the PSTN network and other services.

Zoom does not proactively monitor content in meetings and will not in the future. Zoom doesn't record meetings silently. Neither of these will change.

Our goal is to offer an end-to-end encryption solution that provides a stronger guarantee.

Zoom is dealing with some serious safety issues. When people disrupt meetings (sometimes with hate speech, CSAM, exposure to children and other illegal behaviors) that can be reported by the host. Zoom is working with law enforcement on the worst repeat offenders.

Making it possible for hosts to report people disrupting their meetings even under E2EE is solvable. The likely solution will be a content ring-buffer of the last X seconds on the host's system that can be submitted to Zoom for triage and action.

The other safety issue is related to hosts creating meetings that are meant to facilitate really horrible abuse. These hosts mostly come in from VPNs, using throwaway email addresses, create self-service orgs and host a handful of meetings before creating a new identity.

Zoom's Trust and Safety team can, if they have a strong belief that the meeting is abusive, enter the meeting visibly and report it if necessary.

As you see from the E2E design, there is a big focus on authenticating both the people and the devices involved in E2E meetings. If properly implemented, this would prevent Zoom's employees from entering a meeting, even visibly. There will not be a backdoor to allow this.

Zoom's E2EE implementation will need to be opt-in for the foreseeable future. A large portion of Zoom's meetings use features that are fundamentally incompatible with E2EE (PSTN phones, H323/SIP room systems, cloud recordings, cloud transcription, streaming to YouTube, etc).

So we have to design the system to securely allow hosts to opt-into an E2E meeting and to carefully communicate the current security guarantees to hosts and attendees. We are looking at ways to upgrade to E2E once a meeting has started, but there will be no downgrades.

So this creates a difficult balancing act for Zoom, which is trying to both improve the privacy guarantees it can provide while reducing the human impact of the abuse of its product.

Lots of companies are facing this balancing act, but as a paid enterprise product that has to offer E2EE as an option due to legitimate product needs, Zoom has a slightly different calculus.

The current decision by Zoom's management is to offer E2EE to the business and enterprise tiers and not to the limited, self-service free tier.

A key point: organizations that are on a business plan but are not paying due to a Zoom offer (like schools) will also have E2EE.

Will this eliminate all abuse? No, but since the vast majority of harm comes from self-service users with fake identities this will create friction and reduce harm.

This is a hard balance. Zoom has been actively seeking input from civil liberties groups, academics, child safety advocates and law enforcement. Zoom hopes to find a common ground between these equities that does the most good for the most people.

[Then Alex finished with a three final points:]

1) Most of the people I interact with know this, but I've been working with Zoom as a consultant and helped with the E2E design.

2) None of the major players offer E2E by default (Google Meet, Microsoft Teams, Cisco WebEx, BlueJeans). WebEx has an E2E option for enterprise users only, and it requires you to run the PKI and won't work with outsiders. Any E2E shipping in Zoom will be groundbreaking.

3) At no time does Zoom turn over encryption keys to LE. The issue here is whether Zoom's own employees can enter spaces they host, which is how all major trust and safety teams operate and which is precluded by good E2E.

Alex posted all of that on June 2nd. Then, two days later on the 4th it was contradicted by a Zoom spokesperson who confirmed that free users will be covered by Zoom's AES 256 GCM encryption, but chats will not be covered by additional end-to-end protections:

"Zoom's AES 256 GCM encryption is turned on for all Zoom users - free and paid. Zoom does not proactively monitor meeting content, and we do not share information with law enforcement except in circumstances like child sex abuse. We do not have backdoors where participants can enter meetings without being visible to others. None of this will change.

Zoom's end-to-end encryption plan balances the privacy of its users with the safety of vulnerable groups, including children and potential victims of hate crimes. We plan to provide end-to-end encryption to users for whom we can verify identity, thereby limiting harm to these vulnerable groups. Free users sign up with an email address, which does not provide enough information to verify identity.

The current decision by Zoom's management is to offer end-to-end encryption to business and enterprise tiers. We are determining the best path forward for providing end-to-end encryption to our Pro users.

Zoom has engaged with child safety advocates, civil liberties organizations, encryption experts,

and law enforcement to incorporate their feedback into our plan. Finding the perfect balance is challenging. We always strive to do the right thing."

Uhhhh. Okay. So, today... we still have no idea what they intend. They are clearly asked over and over do you, or do you not, encrypt all Zoom video. And they answer: "Right."