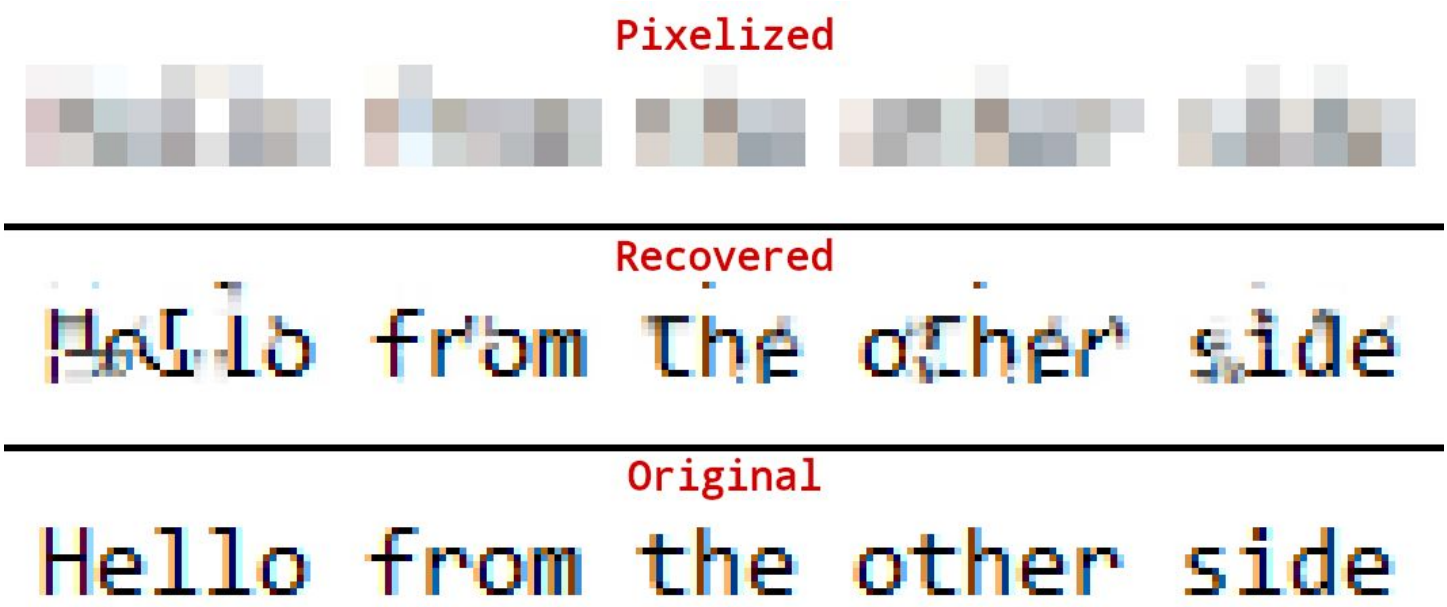
# Security Now! #796 - 12-08-20

# Amazon Sidewalk

## This week on Security Now!

At the beginning of this podcast, you're going to receive some details about another update to Chrome, and news of a few new high-profile Ransomware victims. You'll learn about a breathtaking, remotely exploitable zero-click complete iPhone security compromise, as well as another significant big step forward for DNS privacy beyond DoH. We'll explain the nature of another serious and probably lingering problem within many Android apps, and I have a few interesting bits of miscellany and SpinRite news to share. And before this is over, you will have obtained a full working sense for exactly what it is that Amazon has created and why, with their Amazon Sidewalk neighborhood IoT network concept... coming soon to all of your Amazon devices.

No one ever claimed that fuzzing, blurring or pixelating passwords was secure. But in a nice proof-of-concept, we now have proof that it's decidedly not...



# Browser News

For a change there's nothing earth shaking to report on the browser front. Chrome again updated its various desktop browser builds, bringing them to v87.0.4280.88. This closes eight holes that had been discovered by researchers, four of them regarded as high severity, though unlike the spate of recent 0-days vulnerabilities, none were known to be in use for attacks.

# Ransomware News

**Foxconn,** the world's largest electronics manufacturing company with recorded revenue last year of $172 billion and over 800,000 employees worldwide, suffered a ransomware attack at their 682,000 square foot facility in Mexico over the Thanksgiving weekend. The DoppelPaymer gang claimed to have encrypted between 1200 and 1400 servers (not focused upon measley) after first stealing 100 GB of unencrypted files, and then deleting 20-30 TB of server backups. And what do they want in return for the encryption keys to restore the server's content? 1804.0955 BTC, which is roughly $34,686,000. (I didn't realize that Sharp, Belkin and other brands were subsidiaries of Foxconn.)

Other recent ransomware victims are the **Huntsville, Alabama City Schools district** which provides education to nearly 24,000 students in 37 schools. In a message to parents, the Huntsville City Schools district wrote: "Students, families, and faculty and staff members should shutdown their district-issued devices and ensure the devices remain off until further notice. Additionally, stakeholders should avoid logging on any HCS platforms at both school and home."

**The Metro system in Vancouver, Canada** was also knocked offline by the Egregor ransomware. All of the transit services printers began spitting out ransom demands. So, as was the attacker's intention, it wasn't easy to keep the nature of the "outage" quiet for long.

Oh... And **Egregor also got Kmart**, right at the start of the busiest pre-Christmas holiday shopping season. However, the attack apparently affected Kmart's backend systems since Kmart's retail stores appeared to be operating normally.

Lastly, the massive online education company **"K12 Inc." has decided to pay a ransom** after their systems were hit by Ryuk ransomware in the middle of November. K12 creates tailored online learning curriculums for students to learn from home from kindergarten through 12th grade. And more than 1 million students have used K12 to learn from home rather than in traditional public school environments. K12 announced this week that they has suffered a ransomware attack in mid-November that caused them to lock down some of their IT systems to prevent the attack's spread. They said: "In mid-November, we detected unauthorized activity on our network, which has since been confirmed as a criminal attack in the form of ransomware. Upon identifying unusual system activity, we quickly initiated our response, taking steps to contain the threat and lock down impacted systems, notifying federal law enforcement authorities, and working with an industry-leading third-party forensics team to investigate and assist with the incident." Because the attackers — the Ryuk gang — had gained access to some back-office systems that contained student data and other information, K12 decided to pay up in the hopes of both getting back online more quickly and in keeping any sensitive data private.

# Security News

**The Apple iPhone Vulnerability**

Last week, in a lengthy and painstakingly detailed 30,000-word report written by Project Zero's Ian Beer, we learned how Ian had spent the first half of this year. Before explaining, Ian quotes from the February 2020 Offensive Con conference, during which its keynote speaker said:

> *"Exploits are the closest thing to "magic spells" we experience in the real world: Construct the right incantation, gain remote control over a device."*

So with that, Ian began his expose' by writing: "For 6 months of 2020, while locked down in the corner of my bedroom surrounded by my lovely, screaming children, I've been working on a magic spell of my own. No, sadly not an incantation to convince the kids to sleep in until 9am every morning, but instead a **wormable radio-proximity exploit which allows me to gain complete control over any iPhone in my vicinity**. View all the photos, read all the email, copy all the private messages and monitor everything which happens on there in real-time."

https://googleprojectzero.blogspot.com/2020/12/an-ios-zero-click-radio-proximity.html

In other words, in the midst of distractions from screaming children, Ian successfully discovered and fully developed a working, remote, over-the-air, zero-click, total compromise of any Apple iPhone. Doing so was pretty much everything that Apple has gone to such extremes to first — totally prevent — or if failing that, then to at least raise the bar of exploitation engineering so high as to make it incredibly difficult to weaponize.

In Ian's own words, describing the situation, he wrote:

> *"Of course, an iPhone isn't designed to allow people to build capabilities like this. So what went so wrong that it was possible? Unfortunately, it's the same old story. A fairly trivial buffer overflow programming error in C++ code in the kernel parsing untrusted data, exposed to remote attackers.*
>
> *In fact, this entire exploit uses just a single memory corruption vulnerability to compromise the flagship iPhone 11 Pro device. With just this one issue I was able to defeat all the mitigations in order to remotely gain native code execution and kernel memory read and write.*
>
> *Relative to the size and complexity of these codebases of major tech companies, the sizes of the security teams dedicated to proactively auditing their product's source code to look for vulnerabilities are very small. Android and iOS are complete custom tech stacks. It's not just kernels and device drivers but dozens of attacker-reachable apps, hundreds of services and thousands of libraries running on devices with customized hardware and firmware."*

In other words, with today's massive custom codebases, and more focus being placed upon the fun of designing and shipping new features over the drudgery of examining code for vulnerabilities, it's inevitable that these systems will incorporate a wide range of flaws of varying severity. Everyone knows that generically "bugs" of all kinds are being patched constantly. So, sobering though it is, it should come as no surprise that some of them will be really bad.

The good news is that this is Apple, not Android. So the handsets were all updated back in May with the move to iOS 13.5 and this breathtaking vulnerability was quietly eliminated.

The heart of the problem that Ian uncovered was in a proprietary Apple Wi-Fi based peer-to-peer mesh network protocol known as AWDL — Apple Wireless Direct Link. Being a peer-to-peer mesh protocol handler, AWDL needed to monitor and parse all Wi-Fi network traffic. Parsers are a form of interpreter in that they examine a flow of data for the purpose of understanding it and seeing whether it's relevant to them. What Ian uncovered was that Apple's AWDL parser contained a flaw. Knowing that, and arranging to exploit it in the environment that Apple has deliberately worked to make so hostile, are very different things. But Ian succeeded.

Ian notes that he is one guy who was working alone in his bedroom, whereas the world currently contains many powerful massively-resourced nation state well organized cyberwarfare groups. He feels quite certain that when such resources are brought to bear, other, currently unknown exploitable vulnerabilities will be — and almost certainly are being — uncovered.

I won't delve any deeper into the details of Ian's work, because he has produced a truly fabulous document which takes any interested reader step-by-step through his entire process of discovery and engineering. I strongly recommend it to anyone who's interested in obtaining a richer feel for this sort of work.

**Oblivious DoH (ODoH) — "Oh Dough!"**

You would be forgiven if you thought that ODoH stood for the "Ohio Department of Health", because it does. But once "Oblivious DNS over HTTPS" goes mainstream I would not be surprised to see the Ohio Department of Health no longer being the first search result returned by Google.

We all know about DoH. If we choose to use it, it provides end-to-end encryption between our browsers, and also more recently our entire operating system, and a remote DoH-capable DNS resolving service.

By reusing the existing and quite mature certificate and protocol technologies of HTTPS, it very nicely does what it was designed to do: It strongly prevents any local agency, such as someone monitoring a hotspot or our local ISP, from observing our DNS queries. With DoH running they see encrypted traffic and nothing else.

But there's still one glaring privacy flaw in DoH: It doesn't prevent the DoH service from knowing who's querying DNS for what. Since the DoH IP connection is point-to-point, the DoH resolver still knows who's making the query, and what domain they are querying for. You're local ISP may no longer, but the person you're asking still does.

This is the problem that the new Oblivious DNS over HTTPS has been created to resolve... and its solution is elegant because it's simple: Simply introduce a connection-forwarding middleman into the point-to-point link. Add a 3rd-party proxy which is unaffiliated with the DoH provider. Then the user's connection to their DoH provider routes through the proxy which, in turn, forwards the still-encrypted traffic to their DoH provider.

Since the encryption is still end-to-end, from the user to the DoH provider, the proxy cannot see into the traffic, so although it knows who's asking, it's completely blind to what they're asking for. The DoH provider, in turn, which decrypts the incoming traffic, knows exactly what the incoming connection is asking for, but it has no idea from whom the request has come, since it is receiving anonymized requests that have been forwarded through the intermediate proxy.

This concept, co-developed by Cloudflare, Apple and Fastly, was announced today by Cloudflare: https://blog.cloudflare.com/oblivious-dns/. They already have a trio of qualified and fully independent privacy-committed ODoH proxy providers and ODoH's formal specification is already moving forward through the IETF standardization process.

Eric Rescorla, the CTO of Firefox, says: "Oblivious DoH is a great addition to the secure DNS ecosystem. We're excited to see it starting to take off and are looking forward to experimenting with it in Firefox."

If you REALLY want to be on the bleeding edge, it's possible to get ODoH working today. But at the moment that takes jumping through a bunch of hoops. And given the speed with which DoH appeared and became widely available, I have the feeling that flipping a switch to enable ODoH won't be far away.

**Google Play Core Library problems**

Google provides Android app developers with a component called the Google Play Core Library. Google's Android developer docs described this common library component, saying: "The Play Core Library is your app's runtime interface with the Google Play Store. Some of the things you can do with Play Core include the following:

- Download additional language resources
- Manage delivery of feature modules
- Manage delivery of asset packs
- Trigger in-app updates
- Request in-app reviews

So this Google Play Core Library is a gateway for interacting with Google Play Services from within the app itself. It enables dynamic code loading such as downloading additional levels only when needed, delivering locale-specific resources, and interacting with Google Play's review mechanisms. And since this is the officially sanctioned and recommended way to do this, many popular Android apps utilize this library including:

- Google Chrome
- Facebook
- Instagram
- WhatsApp
- SnapChat
- Booking
- Edge

Facebook and Instagram alone account for 5 billion and 1 billion downloads, respectively. So just imagine the total number of Android apps worldwide that have historically incorporated this library at Google's behest.

Okay. So... the problem is, a quite serious bug was discovered inside this widespread common app library. And because the library is linked into Android apps, to become part of them, this isn't something that Google can fix with an Android update, even for Android smartphones that receive updates.

So what's the bug? I'll quote the company with the oxymoron name "Oversecured" since this was their discovery:

https://blog.oversecured.com/Oversecured-automatically-discovers-persistent-code-execution-in-the-Google-Play-Core-Library/

They explained:

> The Google Play Core Library is a popular library for Android that allows updates to various parts of an app to be delivered at runtime without the participation of the user, via the Google API. It can also be used to reduce the size of the main apk file by loading resources optimized for a particular device and settings (localization, image dimensions, processor architecture, dynamic modules) instead of storing dozens of different possible versions. The vulnerability we discovered made it possible to add executable modules to any apps using the library, meaning arbitrary code could be executed within them. An attacker who had a malware app installed on the victim's device could steal users' login details, passwords, and financial details, and read their mail.

In other words, this Google Play Core Library flaw allows any malicious app to penetrate Android's critical inter-app sandbox which exists to protect apps from having access to one another. And although Google immediately patched this vulnerability back on April 6, 2020, not all developers received the memo.

Check Point Research took a look at this and explained what it means:

> When we combine popular applications that utilize the Google Play Core library, and the Local-Code-Execution vulnerability, we can clearly see the risks. If a malicious application exploits this vulnerability, it can gain code execution inside popular applications and have the same access as the vulnerable application. The possibilities are limited only by our creativity. Here are just a few examples:
>
> - Inject code into banking applications to grab credentials, and at the same time have SMS permissions to steal the Two-Factor Authentication (2FA) codes.
> - Inject code into Enterprise applications to gain access to corporate resources.
> - Inject code into social media applications to spy on the victim, and use location access to track the device.
> - Inject code into IM apps to grab all messages, and possibly send messages on the victim's behalf.

Anyway, you get the point.  It's bad.  And in their proof-of-concept demonstration, they used a malicious app to steal a logon authentication cookie from an older version of Chrome (which was built using the original library). Once in possession of the cookie, the attacker was then able to gain unauthorized access to a victim's Dropbox account.

As I noted earlier, the library was updated back in April — so, eight months ago, right? But last week, Check Point identified 14 apps, having combined downloads of nearly 850 million that remain vulnerable TODAY. Within a few hours of publishing their report, the developers of some of the named apps had released updates that fixed the vulnerability — it only took them eight months, public shaming and outcry.

Check Point analyzed the Google Play Store contents and found that, overall, as of September, 13% of all Google Play applications used the Play Store library and of those, 8% were still using a vulnerable version. So, yes, that's far fewer than all apps, but it turns out that a few of them have massive download counts — like Microsoft's Edge for Android that was, and perhaps still is, vulnerable. The specific 14 still-vulnerable apps that stood out for Check Point due to their popularity were:

| | |
|---|---|
| In the Social category: | Viber |
| In Travel: | Booking |
| In Business: | Cisco Teams |
| In Maps and Navigation: | Yango Pro (Taximeter), Moovit |
| The Dating apps: | Grindr, OKCupid |
| Microsoft's Edge browser | |
| And the utilities: | Xrecorder, PowerDirector |

All of the apps were written to do the right thing; to use Google's recommended library for interfacing with the Play Store after the app had been obtained and run. But their developers had not kept them updated when core critical flaws were discovered and fixed. That's the mistake that they were caught making. Unfortunately, there are many lesser apps that still remain vulnerable and likely will, perhaps forever. And they will always be creating a vulnerability for their users.

This is an interesting twist, in that a widespread collection of apps, themselves all contain a common vulnerability. So updating an older OS, or even buying a brand new Android device doesn't confer protection.

# Miscellany

**Lennert Wouters:** (Regarding the Tesla Model X Key Fob hack) "The system has everything it needs to be secure. But there were just a few small mistakes that allowed me to circumvent all of the security measures."

**The mysterious power of Apple's M1 Arm processor chip:**
https://threadreaderapp.com/thread/1331735383193903104.html

1/ In case you were wondering: *Apple's replacement for Intel processors* turns out to work really, really well. Some otherwise skeptical techies are calling it "black magic". It runs Intel code extraordinarily well.

2/ The basic reason is that Arm and Intel architectures have converged. Yes, the instruction sets are different, but the underlying architectural issues have become very similar.

3/ The biggest hurdle was "memory-ordering", the order in which two CPUs see modifications in memory [made] by each other. It's the biggest problem affecting Microsoft's emulation of x86 on *their* Arm-based "Surface" laptops. [With the result being high x86 emulation overhead.]

4/ So, Apple simply cheated. They added Intel's memory-ordering to their CPU. When running translated x86 code, they switch the mode of the CPU to conform to Intel's memory ordering.

5/ With [those] underlying architectural issues [eliminated] ~~ironed out~~, running x86 code simply means translating those instructions to theirArm equivalent. This is very efficient and results in code that often runs at the same speed.

6/ Sometimes there isn't a direct equivalent, so the translation results in slightly slower code, but benchmarks show x86 being consistently at least 70% of the speed.

7/ In any case, a surprising number of popular apps already run ~~on it~~. Apple seeded developer systems a few months back, allowing people to get their code ready.

8/ Normally, that wouldn't have been enough time. When you recompile code for a new architecture, it usually breaks. But as I said above: Arm and Intel architectures have converged enough that code is much less likely to break, making recompiling easier.

9/ Apple has made surprising choices. They've optimized JavaScript, with special JavaScript-specific instructions, double sized L1 caches, and probably other tricks I don't know of.

10/ Thus, as you browse the web, their new laptop will seem faster and last longer on battery, because JavaScript [has become far more efficient], even though other benchmarks show it roughly the same speed as Intel/AMD.

**InitDisk release 2 published.**
Since last week's podcast I finally finished all of the low-level driver and benchmark demonstration work and produced two release candidates. The second one appears to be final and finished. But two people had found two problems with InitDisk, so I fixed both of those. One was a finicky security permissions issue under Win10, and the other was that on a 1GB USB stick, InitDisk was selecting 512-sector clusters. This is legal but unusual and one guy's BIOS didn't want to boot a USB prepped that way. So, I fixed that too, setting a 4K cluster lower limit on FAT32 formatted drives... and then his system booted without any trouble.

So... for anyone who's interested, InitDisk is now at release 2 and is available at www.grc.com.

## SpinRite

And speaking of SpinRite: I'm pretty sure that next week's podcast will announce that the long-awaited ReadSpeed benchmark is ready for use and testing by everyone here. The DOS based drivers and benchmarking code is finished. Now I'm working to package that into a version of InitDisk that doesn't just format and install FreeDOS, but also installs a turnkey, "Just press Enter" ready-to-run benchmark. And most of that is already finished as well. Over the weekend I released a proof-of-concept Windows app which works. So now I'm getting it polished up.

- ReadSpeed "verbose" output benchmark results: **https://grc.sc/796**
- ReadSpeed "verbose" output /1 detail level: **https://grc.sc/796a**

# Amazon Sidewalk

Amazon has recently created quite a stir by sending letters to their customers owning compatible equipment, informing them that they will be automatically opted-in to Amazon's forthcoming "Sidewalk" program.

***Note that for anyone objecting to auto-opt-in, if you update your smartphone's Alexa app right now, even before Sidewalk goes live, you may preemptively disable your network's use of Sidewalk.***

In any event, Amazon's announcement has been met with no small degree of questioning, confusion and concern over propriety and security. The tech press has been carrying stories with headlines like:

- Fox Business: Amazon to opt-in customers with Echo, Ring devices to new 'Sidewalk' WiFi-sharing feature.
  (As we'll seem that's quite not true, since it's not Wi-Fi sharing. But everyone here will know everything about it by the end of this podcast.)

- Mashable, taking no prisoners: "How to see if Amazon is stealing your internet bandwidth for Sidewalk."

- WGN-TV: Got an Echo or Ring? Soon, Amazon will use them to share your internet with a new 'Sidewalk' network – unless you opt out.

- CNET: Amazon Sidewalk has automatically switched 'on' in your Alexa app. It might be time to check your settings, if you have an Echo smart speaker or Ring camera.

- TechHive: Welcome to Amazon Sidewalk! Now, here's how to turn it off.

- Gizmodo: You Need to Opt Out of Amazon Sidewalk
  "Have you heard of Amazon Sidewalk? Probably not. But there is a good chance that you or someone you know has an Amazon Echo or Ring camera. And if you own one of those devices and live in the U.S. (or know someone who does), you need to tell them to opt-out of the service as soon as possible."

Okay. So let's all take a deep breath. I have the 13-page "Amazon Sidewalk Privacy and Security Whitepaper." So we're all going to get our questions answered. But let's first begin by looking at Amazon's description and their hopes for what it is and why they think it's a good idea to do this:

Dated: September 27, 2019 — And as we noticed from the more recent devices that are already equipped with 900 Mhz radios, they started working on this last year in 2019…

**Get connected convenience beyond your front door.**

Many of the smart devices in our homes today rely on Bluetooth and Wi-Fi connections to stream music to a nearby speaker or help a video doorbell notify us when a package is delivered. But these connections only extend so far. On the other end of the spectrum, 5G cellular is incredibly important when you need reliable, long distance, guaranteed delivery of data, but it can be complex. In the space around homes, that leaves a middle-ground for devices like sensors and smart lights that can benefit from low-cost, low-power, low-bandwidth connections. Customers shouldn't have to settle for connected devices that lose functionality past the front door, which is why we're excited to introduce Amazon Sidewalk.

Amazon Sidewalk is a new long-term effort to greatly extend the working range of low-bandwidth, low-power, smart lights, sensors, and other low-cost devices customers install at the edge of their home network. Using the 900 MHz spectrum, we are developing a new protocol we project can increase the connection range of these devices by more than one half mile/one kilometer. With Amazon Sidewalk, customers will be able to place smart devices anywhere on their property and know they'll work great, even in dead spots where Wi-Fi and Bluetooth don't reach.

Using the 900 MHz spectrum to help devices communicate isn't new. In fact, it's been around for decades, providing reliable, secure connections for long-range devices like the radios used by emergency services and the digital pagers carried by doctors on-call. It's by combining this tested communications network with an innovative new protocol developed by Amazon that we arrived at Sidewalk; a new way for the next generation of low-cost, low-bandwidth sensors and smart devices to work together to create a secure network of long-distance connections, bridging the connectivity gaps around our homes.

The immediate benefit of a 900 MHz-based network is the ability to use your favorite connected devices even if they're located far away from the router inside your home. Today, Ring Smart Lighting Bridges use connections in this spectrum to extend the range of smart lighting products, and soon additional devices including the latest generation Ring Floodlight Camera and Ring Spotlight Camera will also help customers extend the network connections around their homes and control those 900 MHz devices at much greater distances.

Better network connectivity can also help keep devices safe and up to date. Today when customers place a smart device at the edge of their home network, poor network connectivity can prevent that device from receiving important feature and security updates. By extending long-range, low-bandwidth connections using the Amazon Sidewalk network, customers won't have to worry about smart devices that don't have access to the latest security updates or work as intended because they're out of network range.

In the near future, we also see the potential to help customers get more from 900 MHz connections in their neighborhoods, creating a broad network among neighbors that can be used to extend connectivity all the way to your mailbox out at the street where a smart sensor lets you know exactly when your mail has been delivered, or to a water sensor that lets you know it's time to water the garden in the backyard.

For example, just a week ago Amazon employees and their friends and family joined together to conduct a test using 700 Ring lighting products which support 900 MHz connections. Employees installed these devices around their home as typical customers do, and in just days, these individual network points combined to support a secure low-bandwidth 900 MHz network for things like lights and sensors that covered much of the Los Angeles Basin, one of the largest metropolitan regions in the United States by land area.

This neighbor-created network demonstrates the potential of Amazon Sidewalk – a broad coverage network, great for low-bandwidth, low-cost devices, that requires no complex setup or maintenance for customers. But the benefits don't stop there. With Sidewalk, we also see the opportunity to deliver new devices and experiences that delight our customers.

As one example, this week we announced Fetch, a compact, lightweight device that will clip to your pet's collar and help ensure they're safe. If your dog wanders outside a perimeter you've set using the Ring app, Fetch will let you know. In the future, expanding the Amazon Sidewalk network will provide customers with even more capabilities like real-time location information, helping you quickly reunite with your lost pet. For device makers, Fetch also serves as a reference design to demonstrate the potential that devices connected to a broad, reliable network can provide to their customers.

Extending the convenience of a long-range network will take time, but we're already working quickly to bring this future to life for customers. For device makers, we plan to publish protocols that any manufacturer can use to build reliable, low-power, low-cost devices that benefit from access to long-range, low-bandwidth wireless connections. In the meantime, you can sign up to be notified when more information is available.

Amazon Sidewalk is a long-term effort, but we're excited to get started and can't wait to see what device makers build and how customers benefit. The possibilities are endless.

Amazon's term for one of their devices which can participate in Sidewalk is a "Sidewalk Bridge." The device itself is connected, as it always has been, to its owner's Wi-Fi network or wired LAN. But when Sidewalk mode is enabled on that device, it then brings up a new Sidewalk protocol which it transacts over either Bluetooth Low-Energy or — for more recent and I'm sure all future devices — the the non-WiFi, much lower frequency, 900 Mhz (33 centimeter unlicensed Amateur radio band) band which offers longer range, better building penetration and lower bandwidth with reduced lower power consumption.

Today's "Sidewalk Bridges" include many Echo devices and selected Ring Floodlight and Spotlight Cams. Specifically...

- All Echos ***after*** the first generation:
  - Amazon Echo (second-gen, 2017, BLE only)
  - Amazon Echo (third-gen, 2019, BLE only)
  - Amazon Echo (fourth-gen, 2020, BLE and **900 MHz**)

- **All** Echo Dots:
  - Amazon Echo Dot with Clock (first-gen, 2019, BLE only)
  - Amazon Echo Dot with Clock (second-gen, 2020, BLE only)
  - Amazon Echo Dot (first-gen, 2016, BLE only)
  - Amazon Echo Dot (second-gen, 2016, BLE only)
  - Amazon Echo Dot (third-gen, 2018, BLE only)
  - Amazon Echo Dot (fourth-gen, 2020, BLE only)
  - Amazon Echo Dot Kids Edition (third-gen, 2020, BLE only)

- **Both** Echo Pluses:
  - Amazon Echo Plus (first-gen, 2017, BLE only)
  - Amazon Echo Plus (second-gen, 2018, BLE only)

- All five of the Echo Shows:
  - Amazon Echo Show (first-gen, 2017, BLE only)
  - Amazon Echo Show (second-gen, 2018, BLE only)
  - Amazon Echo Show 5 (2019, BLE only)
  - Amazon Echo Show 8 (2019, BLE only)
  - Amazon Echo Show 10 (2020, BLE and **900 MHz**)

- Amazon Echo Spot (2017, BLE only)
- Amazon Echo Studio (2018, BLE only)
- Ring Floodlight Cam (2019, BLE and **900 MHz**)
- Ring Spotlight Cam Wired (2019, BLE and **900 MHz**)

---

https://m.media-amazon.com/images/G/01/sidewalk/privacy_security_whitepaper_final.pdf

---

A couple of points:

- A simple control is provided to enable and disable participation in the neighborhood network. When customers first turn on a new Sidewalk gateway device, they will be asked whether they want to join the network. For customers with existing devices that are Sidewalk capable, an over-the-air (OTA) update will connect them to the network—no action is needed. These customers will first receive an email about the pending update and instructions for how to disable, if that is their choice.

- As a crowdsourced, community benefit, Amazon Sidewalk is only as powerful as the trust our customers place in us to safeguard customer data. To that end, this document outlines the steps we have taken to secure the network and maintain customer privacy. These efforts are core to our mission and will continue to evolve and improve over time.

- The maximum bandwidth of a Sidewalk Bridge to the Sidewalk server is 80Kbps, which is about 1/40th of the bandwidth used to stream a typical high definition video. Today, total monthly data used by Sidewalk enabled devices, per customer, is capped at 500MB, which is equivalent to streaming about 10 minutes of high definition video.

The overall network can be visualized as consisting of (1) Gateways, (2) Endpoints, an (3) Amazon-operated Sidewalk Network Server, (4) Application Servers & (5) Message packets.

So the Gateways (also known as Sidewalk Bridges) forward packets to and from the Sidewalk Endpoints and the Sidewalk Network Server. Gateways will be Amazon devices, like the Ring Floodlight Cam, that use 900 MHz (LoRa and/or FSK, and/or Bluetooth Low Energy (BLE) to provide connection to the Sidewalk network. (FSK stands for Frequency Shift Keying. It's one means of modulating the radio frequency carrier to communicate information. "LoRa", is a clever technology using bi-directional frequency "chirps" to obtain extremely good receiver sensitivity and very low bit error rates (BER) from inexpensive chips. It enables low-data rate applications to obtain much longer range from a very low power and cost budget.)

Sidewalk Endpoints (also known as Sidewalk-Enabled devices, edge devices, endpoints, or Applications) can roam around on the Sidewalk network by connecting to Sidewalk Gateways. These endpoints are low-bandwidth/low-power smart products like leak sensors, door locks, lights, or devices attachable to valuables or for example a pet in order to know where they are. Sidewalk endpoints can be built and maintained by Amazon or by third party developers. Sidewalk Gateways can also act as an endpoint and receive Sidewalk benefits like maintaining functionality when the device falls offline. THAT is the first clear benefit we've seen. It would mean that if your own router or cable modem froze, so that your Sidewalk-enabled IoT devices — like lighting or your door lock — was offline, they could seamlessly remain connected to the Internet thanks to the neighborhood's active Sidewalk network... by essentially riding over one of your neighbor's Sidewalk Endpoints that was not off the Net.

The Sidewalk Network Server (or SNS), operated by Amazon. It's responsible for verifying that the incoming packets are coming from authorized Sidewalk devices, routing packets to the desired destination (an application server, endpoint or GW device), and keeping the network time-sync'd.

Application Servers host the Sidewalk endpoints and implement the business logic for the user experience and the desired product functionality. Application servers are managed by the Sidewalk endpoint manufacturer, which can be Amazon or a third party. So, for example, if the garage door opener manufacturer Genie were to create a smart Sidewalk-enabled garage door opener, it would normally be connected to your home LAN Wi-Fi, and it would normally be offering Sidewalk connectivity to your neighborhood. But, reciprocally, it would also be able to USE the neighborhood's Sidewalk network if your Wi-Fi was not available. So if you needed to reach it while your home LAN was down, the Genie application server would route through the Amazon Sidewalk Network Server to reach your Sidewalk-enabled garage door opener... via a neighbor's Internet connection.

And, finally... Packets (also known as Messages) are exchanged between Sidewalk Endpoints and the Application Server through the Gateways and Sidewalk Network Server.

The network's design reveals that Amazon has put a great deal of time, attention and design work into creating a system that provides the security controls that Amazon requires for the network to operate safely while also blinding Amazon to all of the network's messaging traffic.

The network used three layers of encryption, reminiscent of The Onion Router network:

1. The innermost encryption is the Application layer which protects the privacy and security of the communications between the endpoint and the Application Server. This is the layer that does the actual signaling work.

2. The Application layer encryption is, in turn, encrypted at the enndpoint by another layer using keying that it shares with the Amazon Network Server. This conceals and protects the endpoint's Sidewalk packet over the air. Therefore, plain-text data encrpyted by this layer is accessible only to the endpoint and the Amazon Network Server (SNS).

3. And, finally, the so-called "Flex Layer", is added by the Sidewalk Gateway device. It provides the Amazon Network Server with a trusted and tamper-proof reference for message-received time and adds an additional layer of packet confidentiality. the plaintext data encrypted by this layer is therefore accessible only to the gateway device and Amazon's Network Server.

As I noted above, ultimately, the communication is between the Endpoint devices and their Application servers, with the Sidewalk gateway devices and Amazon's Network Server functioning as intermediaries. Consequently, the innermost wrapper of encryption is fully end-to-end between the Endpoint device and the device's matching Application server. Neither the gateway that facilitates the communications at the neighborhood end, nor the Amazon Network Server that facilitates the communications on the Internet, are able to see what's being transacted.

This will give you some sense for the architecture Amazon has created:

Amazon has carefully designed privacy protections into how Sidewalk collects, stores, and uses metadata. Sidewalk protects customer privacy by limiting the amount and type of metadata that Amazon needs to receive from Sidewalk endpoints to manage the network. For example, Sidewalk needs to know an endpoint's Sidewalk-ID to authenticate the endpoint before allowing the gateway to route the endpoint's packets on the network. Sidewalk also tracks a gateway's usage to ensure bandwidth caps are not exceeded and latency is minimized on a customer's private network.

Information customers would deem sensitive, like the contents of a packet sent over the Sidewalk network, is not seen by Sidewalk; only the intended destinations (the endpoint and application server) possess the keys required to access this information. Sidewalk's design also ensures that owners of Sidewalk gateways do not have access to the contents of the packet from endpoints (they do not own) that use their bandwidth. Similarly, endpoint owners do not have access to gateway information. The Sidewalk Network Server continuously "rolls", or changes transmission IDs (TX-ID) and Sidewalk Gateway IDs every 15 minutes to prevent tracking devices and associating a device to a specific user. The IDs use a time-based cryptographic system like our TOPTs so that the endpoints are continuously and autonomously re-identifying themselves using a periodically changing ID, and the Amazon server shared the underlying key and thus can determine who's who. But no one monitoring the metadata could determine whether the same or some other device was communicating.

From the view of the Endpoint (the device using someone's gateway device), it is only able to view information that pertains to the normal operation of its device (i.e. whether the smart light is on or off). It is unable to see routing information, or even what gateway (if it's a foreign gateway) the smart light is receiving support from, nor any information about that GW and GW owner. The GW information is encrypted behind
the Sidewalk Network Layer and Flex Layer.

From the view of the gateway device, it's unable to see what endpoints (that they do not own) are receiving support from their gateway. They have no idea what types of endpoints are connected, nor the times in which they are connected, or information about the owner of the endpoint. All of that endpoint information is encrypted as it passes by the Sidewalk Application Layer.

At the far end, the Application Server is unable to see any information pertaining to the GW owner. It only has access to the endpoint information since those outer wrappers and metadata — like the gateway ID — will have been removed by Amazon's Network Server.

As we would hope, the registration-time establishment of unique identifying credentials assure that only trusted and known devices can enter the Sidewalk network, preventing unauthorized devices from joining. The Sidewalk Network Server (SNS), Application Server, and each Sidewalk device (both gateways and endpoints) are provisioned with a unique set of Sidewalk credentials that are used during the Sidewalk device registration process to mutually authenticate each devices' identity and to derive unique session keys between them. Rolling encryption keys are periodically derived periodically from their respective session keys.

Amazon also noted that to protect their customer's privacy, the routing data that they were necessarily using to link the location of a known endpoint device to perhaps someone else's gateway — by network but also probably by geographic location — is deliberately wiped and discarded after 24 hours.

So that's the system:

- It's not neighborhood Wi-Fi. It's an encrypted IoT communications signalling solution.

- It's initially primarily BLE, since all current devices have at least a BLE radio, and only the few newest also have the newer 900 Mhz radio that will really give the system some useful range.

- While it will need to survive deeper analysis by cryptographers and academic analysts, Amazon has clearly worked to create and deliver a state-of-the-art secure messaging solution. If this were to succeed, we might be in a world where a cool, low-frequency, low bandwidth, low data rate message-passing network was everywhere. In this world, previously registered devices could roam freely while remaining able to send and receive status update and command messages.