Security Now! #768 - 05-26-20
Contact Tracing Apps R.I.P.

This week on Security Now!

This week we begin with some browser news to examine a nifty new trick to be offered by the next Firefox 77 and we spend a bunch of time on the many new features -- and how to enable them -- being offered in Chrome's 83rd edition. We also look at Adobe's four emergency out-of-cycle patches, and a surprisingly robust and well designed new Jailbreak for iPhones. We take a look at a surprisingly powerful DNS amplification attack with a packet count multiplier of up to 1620, the sad but true complete collapse of Bluetooth connection security and the odd report of eBay scanning their user's PC's. We'll then share a bit of closing the loop listener feedback and a quick bit of miscellany, then I'm going to editorialize a bit about why I'm very sure that contact tracking apps are dead on arrival.

Browser News

Firefox 77 to pick up a nifty new trick…

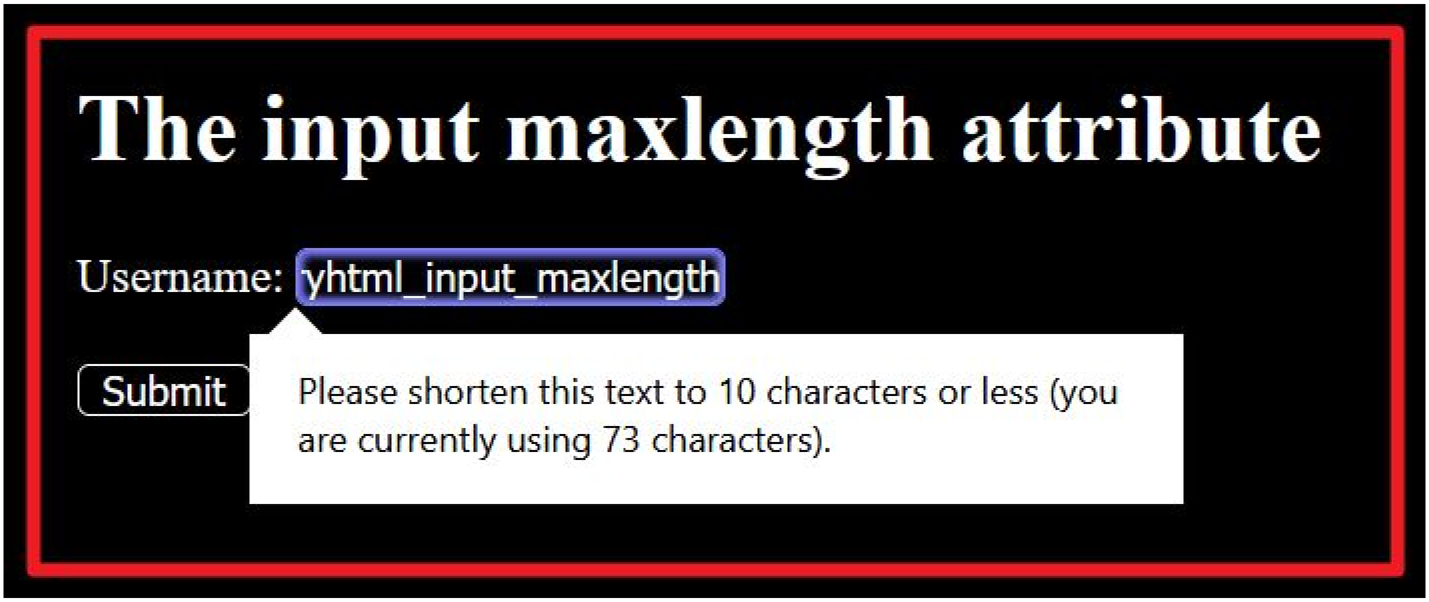
One of the attributes of fill-in fields on an HTTP web form is "maxlength". A webpage's coder can instruct the web browser to stop accepting user input beyond the specified length.

Supposedly -- though this something I don't recall ever encountering -- there are websites that use this maxlength specification on their account creation and subsequent login password input fields.

As a result, when a user pastes a longer password into a shorter field, the browser, which has been clearly instructed to admin text no longer than ‘x’ characters, will indiscriminately discard the "overage" without providing any user feedback. The website itself has no way of knowing that an attempt was made to paste more into the field, since this truncation is all handled on the browser's end.

Now... as long as the site **never** changes its mind about the maximum length of its passwords, and the user always goes about pasting the overly long password string into the too-short field, everything should be okay, since the website would always be receiving the same first ‘N’ characters of the user's pasted password. But it's conceivable that a site might wish to modernize its authentication handling and increase its password length. For example, some random HIPPA or other regulation might say “thou shalt allow passwords of up to ‘X’ characters.” So, if the site were then forced to widen it password length aperture to comply, more of the users' same password would suddenly be admitted and a great deal of breakage and hair pulling would ensue.

We are currently at Firefox 76, so the next major release will bring an interesting new feature to help us with this weird edge case: Since the web browser knows when we have attempted to enter or paste more characters into a field than that field has been configured to accept, with Firefox release 77, any field where this is attempted will be immediately highlighted with a clearly visible red warning rectangle:



The input maxlength attribute

Username: yhtml_input_maxlength

Please shorten this text to 10 characters or less (you are currently using 73 characters).

I'll conclude by re-implanting what everyone in this audience very well knows: password length limits are dumb. Period. And, just for empahsis: Dumb Dumb Dumb! There is absolutely no justification for them. Every recipient of the password should immediately hash it -- or even better, in the browser before it is sent over the wire. Then, what's being stored is a Base64 text encoding of the password's hash with the random salt that was used by the browser's PBKDF function. This way, the recipient never has it in the clear. There is NEVER any reason for any employee of the recipient to have or see the password in the clear. They never have any need or reason to send it or to read it over the phone. The only proper way to deal with passwords that cannot be supplied for whatever reason is have the user authenticate their identity through some other means -- typically by having them able to receive eMail at a pre-registered address and send a password forgiveness link to that eMail account.

A password could be required to meet minimal complexity requirements, but there's just no good reason to place a cap on a password's maximum complexity.

Chrome 83

As planned, Chrome skipped over 82. Jumping from 81.0.4044.138 -to- 83.0.4103.61 and with it we received a bunch of new goodies:

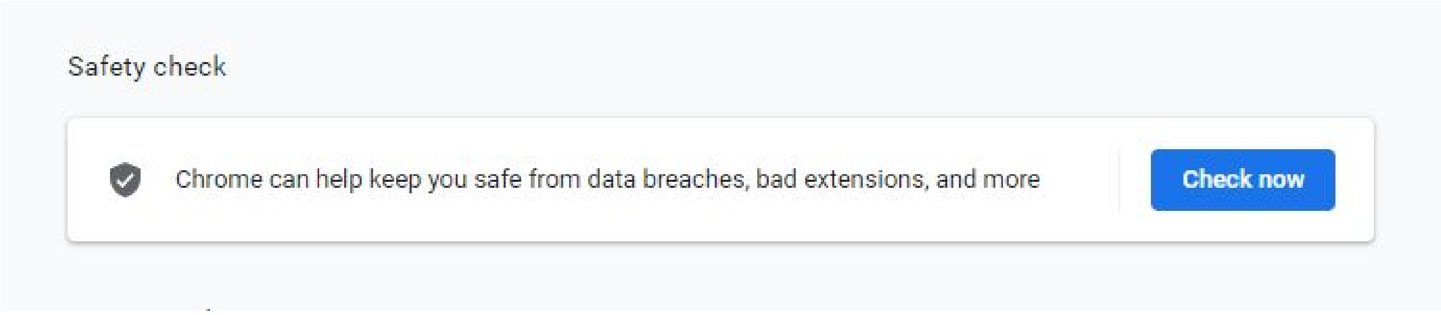
Cookie management, both for incognito and regular mode has been simplified and clarified. Global and per-site settings are clearer. The Site Settings control have been reorganized into two separate sections to make it easier to find the more important website permissions such as access to location, camera, microphone and notifications. And a newly added section shows the most recent permissions activity. The "People" item in Settings is now "You and Google" where the sync controls are located. And since Google says that many people regularly delete their browsing history, they've moved the "Clear browsing data" control for doing that to the top of the Privacy & Security section -- though you can get to it even quicker from the "More Tools" pop-up menu.

Chrome 83 also offers a new "Safety check" feature which provides a number of services: It will tell its user if the passwords Chrome is storing have been compromised, and if so, how to fix them. It will flag if Safe Browsing, the technology to warn before visiting a dangerous site or download a harmful app or extension, is turned off. It will verify that the version of Chrome that's running is up to date. And if any malicious extensions are installed it will tell you how and where to remove them.

The only problem with this is that Google plans to roll out these features over time and they are currently disabled. At least they were for me. But a bit of hunting (or Googling, as it were) revealed that by jumping WAY down a nearly endless page of configuration options, or using this link:

chrome://flags/#privacy-settings-redesign

Or goto the URL: chrome://flags and then search for "Privacy" and you'll find "Privacy Settings Redesign." which can be switched from "Default" to "Enable", and after a relaunch the main Settings page acquires a new "Safety Check" option that wasn't there before:



Safety check

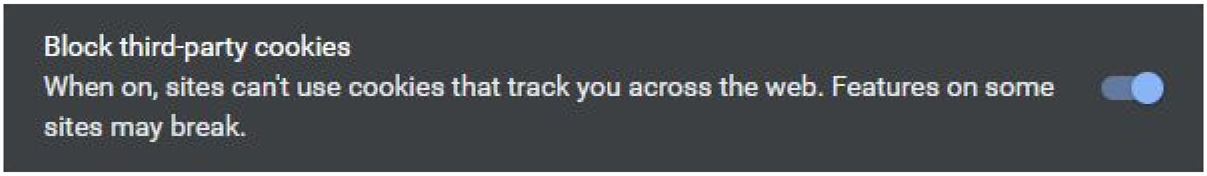
Chrome can help keep you safe from data breaches, bad extensions, and more

I ran the check and got all four checkmarks indicating that the things Chrome just checked for me were all okay.

Also with this release Chrome, Google has started blocking third-party cookies by default in Incognito mode and there's a new user interface toggle to control it. But as with the new "Privacy Settings Redesign" it needs to be enabled first. In this case on the chrome://flags page you need to search for "Improved Cookie Controls" or use this URL

chrome://flags/#improved-cookie-controls

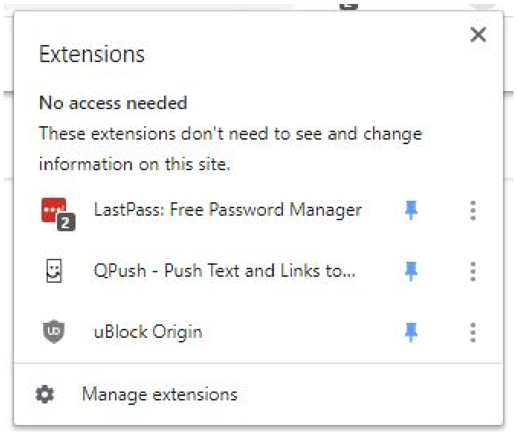
After that you'll find two entries at the top of the page: "Enable improved cookie controls UI in incognito mode" and beneath that one, "Enable improved UI for third-party cookie blocking". Not knowing why anyone would choose to NOT receive all of those improvements, I switched them both from “Default” and “Enable” and did a quick relaunch. Now my Incognito home page displays a prominent banner:



Block third-party cookies
When on, sites can't use cookies that track you across the web. Features on some sites may break.

There's also a new “Extensions Toolbar Menu”. When enabled it adds a little puzzle piece icon to the right of any extensions that may already be displaying to create a quick-access drop-down to all of the extensions' settings and management.

chrome://flags/#extensions-toolbar-menu



Extensions

No access needed
These extensions don't need to see and change information on this site.

LastPass: Free Password Manager
QPush - Push Text and Links to...
uBlock Origin
Manage extensions

The last of the UI improvements is the much anticipated “Tab Groups” feature. As with all of the other new goodies, it must first be enabled by searching the chrome://flags page for “Tab Groups”. That will reveal the three new settings for Tab Groups: “Tab Groups”, “Tab Groups Collapse” and “Tab Groups Feedback.”

Tab Groups: Allows users to organize tabs into visually distinct groups, e.g. to separate tabs associated with different tasks.

Tab Groups Collapse: Allows a tab group to be collapsible and expandable, if tab groups are enabled.

Tab Groups Feedback: Enables the feedback app to appear in the tab group editor bubble, if tab groups are enabled.

When the Tab Groups features are enabled tabs get additional tab group management items in their drop-down context menus.

And, finally, search chrome://flags for "Secure DNS" and you find the setting to enable to obtain secure DNS over HTTPS / DoH operation.

"Secure DNS lookups": Enables DNS over HTTPS. When this feature is enabled, your browser may try to use a secure HTTPS connection to look up the addresses of websites and other web resources.

This reportedly works the same way as Win10 does or will: If the DNS provider that's been configured offers DoH DNS then when this is enabled, Chrome will attempt to use DoH whenever possible.

As we know, with Chrome 79 Google pivoted back to holding that displaying the "http://" at the front of every URL is not useful, and further that the "www" or "m" (for a mobile subdomain) that a URL's full domain name might begin with is also better off being "elided" from the screen. They declared that these were "trivial subdomains".

With Chrome 83, as promised, those of us who are sticklers for details have obtained the ability to display the full unadulterated -- or un-elided -- URL. We need to jump through a few hoops, but that's to be expected. First we need to enable the new line-item in the URL's context menu by returning to the much-visited "chrome://flags" page and this time searching for "omnibox context" That will find:

Context menu show full URLs
Provides an omnibox context menu option that prevents URL elisions.

After the requisite quick restart, pull down the context menu in the URL itself and enable "Always show full URLs."   ***Thank you, Google... that wasn't really so hard, was it??***

And, this 83rd release of Chrome and repaired 38 known vulnerabilities.

Security News

Adobe's four out-of-cycle emergency updates:
Just a quick not for users of Adobe's Character Animator (CVE-2020-9586), Premiere Pro (CVE-2020-9616), Audition (CVE-2020-9618), and Premiere Rush (CVE-2020-9617): Adobe felt so strongly about four vulnerabilities, one in each of those four products, that they broke with their nominal monthly past update cycle to push these out since their previous week's regular Patch Tuesday, which was its own bumper crop. The problem with Character Animation was a remote code execution flaw. You've really got to wonder what in the world they are doing for a "Character Animation" tool to have a remote code execution vulnerability. But in any event, if you are a user of any of those four you'll want to check again for updates to any of those.

Unc0ver - https://unc0ver.dev/
There's a new iOS jailbreak in town, and as JailBreaks go, it looks VERY nice!

It leverages a previously unknown 0-day flaw in every version of iOS from 11.0 to 13.5, though, interestingly, two little ranges: 12.3-12.3.2 and 12.4.2-12.4.5, are excluded. So this works from iPhone 6S though today's latest iPhone 11 Pro Max, and it has the feel of a professional jailbreak. I've never been tempted to do that, since I'm happy to live within the confines that Apple has designed. And security is always a concern. But this work is very impressive.

As we know, the earlier CheckM8 / CheckRain jailbreak leveraged a flaw in earlier devices' boot ROM to allow a single-boot takeover of iOS -- iPhone 4S through iPhone 10. Being a leverage of a flaw in the hardware, the problem can never be eliminated by Apple. But that one won't work at all on anything newer than an iPhone 10 since Apple fixed the ROM after that. The "Unc0ver" jailbreak will work now and until iOS is patched, which probably won't take long. So, if you're curious to play with a very nicely (dare I say professionally) implemented jailbreak, now's your chance. Go to: https://unc0ver.dev/

The site explains that it's widely compatible and very stable. They explain: "Utilizing proper and deterministic techniques, jailbreak stability is guaranteed." They also claim security: "Utilizing native system sandbox exceptions, security remains intact while enabling access to jailbreak files." And under "Extensively Tested" they write: "Unc0ver has been extensively tested to ensure it's a seamless experience on all devices. Unc0ver works on all devices on iOS versions between 11.0 and 13.5. Below you can find a list of all devices that have been specifically tested."

They have a "What's New" section:

- Full-fledged support for all devices on iOS 11.0-13.5 with Cydia and tweak injection.
- Enable unrestricted storage access to jailbreak applications for sandbox backwards compatibility while keeping security intact by leaving the security restrictions enabled for system and user applications.
- Update Phone Rebel case models and bundled packages. *[Whatever that means.]*

The their so-called "Important Information" actually is important enough for me to share since it serves to provide a more clear sense for what I mean by this work's "professionalism"...

> **Important Information**
> unc0ver is designed to be stable and enable freedom from the moment you jailbreak your device. Built-in runtime policy softener allows running code without Apple's notarization and pervasive restrictions. Proper runtime modifications to iOS kernel modify security features as necessary and result in:
>
> **No Extra Security Vulnerabilities**
> unc0ver preserves security layers designed to protect your personal information and your iOS device by adjusting them as necessary instead of removing them. With this security adjusted on your iOS device, you can run your favorite jailbreak apps and tweaks while still being protected from attackers.
>
> **Stability & Battery Life**
> unc0ver is tirelessly developed and rigorously tested with software stability and battery life in mind. If you're experiencing issues with stability or battery life, we recommend searching your device for faulty tweaks.
>
> **Reconciliation of Services**
> Services such as iCloud, iMessage, FaceTime, Apple Pay, Visual Voicemail, Weather, and Stocks, have been reconciled and still work on the device.
>
> **Future Software Updates**
> The ability to apply future updates is retained. Modifications to iOS kernel are done in memory. This results in the jailbroken iPhone, iPad, or iPod touch staying operable when a future Apple-supplied iOS update is installed.
>
> ***[however...]***
>
> **iOS Updates**
> unc0ver Team strongly cautions against installing any iOS software update that breaks unc0ver as you can't re-jailbreak on versions of iOS that are not supported by unc0ver at that time.
>
> **Jailbreak Legality**
> It is also important to note that iOS jailbreaking is exempt and legal under DMCA. Any installed jailbreak software can be uninstalled by re-jailbreaking with the restore rootfs option to take Apple's service for an iPhone, iPad, or iPod touch that was previously jailbroken.

Installation through Linux only supports Cydia Impactor, so an Apple developer account is required, but Windows and MacOS methods are available to support other options.

And in related news, I'll note that the 0-day exploit broker Zerodium 14 days ago tweeted on May 13th that they would not be purchasing iOS RCE vulnerabilities for the next few months due to <quote> "a high number of submissions related to these vectors." In other words, there's apparently a bit of a market glut in iOS RCE's. (Or perhaps it's that hackers stuck at home have more time on their hands to dig more deeply into iOS and are discovering more gems?)

The NXNS Attack

A group of cybersecurity researchers in Israeli have responsibly disclosed details about a new way they worked out of using the Internet's domain name resolution system to hugely amplify (by a factor of at least 1620 packets) a DDoS attack to take down targeted websites. We are learning of it only now because the many companies who are helping to run their portions of the Internet infrastructure, including PowerDNS, CZ.NIC, Cloudflare, Google, Amazon, Microsoft, Oracle's Dyn, Verisign, and IBM's Quad9, have all since patched their software to address the problem.

When a DNS lookup is requested, the request is made to a so-called recursive DNS resolver. The idea is that if **it** doesn't already have the IP for the requested domain in its cache, it will take on the task of going out to find it for the user making the request. So begins the vulnerability...

It will ask one of the top level authoritative name servers for the IP of the nameserver that's authoritative for the second level domain, for example, attacker.com. If the domain being looked up is "noodles.attacker.com" then our dutiful recursive name server next asks the nameserver for "attacker.com" for the list of IPs of nameservers for the "noodles.attacker.com" domain. And therein lies the problem:

The "attacker.com" nameserver can provide a long list of apparently distinct nameservers but all having the same IP. At that point the recursive nameserver will begin querying the victim IP for the DNS information of "noodles.attacker.com." It knows nothing about this, so when it doesn't respond or if it responds with "Huh?!?!" the user's recursive nameserver will then try the next fake nameserver in the list it received from the "attacker.com" name server. And lots of retries also happen since DNS runs over UDP and packets can get lost. The result is a massive traffic amplification attack since the hundreds of thousands of recursive nameservers located all over the Internet can all be queried to resolve the request. This will result in an Internet-wide inundation of unsolicited DNS queries and bandwidth saturation.

The researchers said that the attack can amplify the number of packets exchanged by the recursive resolver by as much as a factor of more than 1,620, flooding the target domain with superfluous requests and taking it down.

The researchers said: "Our initial goal was to investigate the efficiency of recursive resolvers and their behavior under different attacks, and we ended up finding a new seriously looking vulnerability, the NXNSAttack. The key ingredients of the new attack are (i) the ease with which one can own or control an authoritative name server, (ii) the usage of nonexistent domain names for name servers and (iii) the extra redundancy placed within the DNS structure to achieve fault tolerance and fast response time.

They recommended that network administrators who run their own DNS servers update their DNS resolver software to the latest version.

I went looking for additional information, but the NXNS attack site is off the Net at the moment. Presumably it's under an active attack. So its homepage and research paper were unavailable.

https://www.nxnsattack.com/

BIAS - Bluetooth Impersonation AttackS
So, remember how we said that the Bluetooth pairing event is inherently insecure because that's the moment when two devices having no previous knowledge of one another are negotiating a shared key which they will henceforth share and use to re-recognize one another in the future???

Well, it turns out that, as they say, that was necessary but not sufficient. What we have as a result of new research by a group who discovered a means of later performing exactly the sort of impersonation attack that the whole Bluetooth one-time-pairing scheme was designed to prevent... is nothing less than a complete collapse of Bluetooth security.

> Abstract
> Bluetooth (BR/EDR - Basic Rate / Enhanced Data Rate -- essentially standard communicating Bluetooth) is a pervasive technology for wireless communication used by billions of devices.
>
> The Bluetooth standard includes [both] a legacy authentication procedure and a secure authentication procedure, allowing devices to authenticate to each other using a long term key. Those procedures are used during pairing and secure connection establishment to prevent impersonation attacks.
>
> In this paper, we show that the Bluetooth specification contains vulnerabilities enabling impersonation attacks during secure connection establishment. Such vulnerabilities include the lack of mandatory mutual authentication, overly permissive role switching, and an authentication procedure downgrade (to the legacy version). We describe each vulnerability in detail, and we exploit them to design, implement, and evaluate master and slave impersonation attacks on both the legacy authentication procedure and the secure authentication procedure.
>
> We refer to our attacks as Bluetooth Impersonation AttackS (BIAS). Our attacks are standards compliant, and are therefore effective against any standards-compliant Bluetooth device regardless of the Bluetooth version, the security mode (e.g., Secure Connections), the device manufacturer, or the implementation details. Our attacks are stealthy because the Bluetooth standard does not require notifying end users about the outcome of an authentication procedure, or the lack of mutual authentication.
>
> To confirm that the BIAS attacks are practical, we successfully conducted them against 31 Bluetooth devices (incorporating 28 unique Bluetooth chips) from major hardware and software vendors, implementing all the major Bluetooth versions, including Apple, Qualcomm, Intel, Cypress, Broadcom, Samsung, and CSR.

The BIAS attacks allow an attacker having knowledge of the Bluetooth address of one endpoint of a previously-established connection (which is trivial to obtain in practice) since it’s being broadcast, to successfully impersonate that device when connected to the other endpoint. So we have nothing less than a complete and total collapse of Bluetooth’s secure authentication.

What’s important is to understand that this is not the result of any bug, it’s a failure in the design of the Bluetooth standard. It’s a disaster for Bluetooth security.

The researchers tested the attack against smartphones, tablets, laptops, headphones, and single-board computers including the Raspberry Pi’s. ALL devices were found to be vulnerable to their BIAS attacks.

The standards-setting body “Bluetooth SIG” said it's updating the Bluetooth Core Specification to “avoid a downgrade of secure connections to legacy encryption,” which lets the attacker initiate “a master-slave role switch to place itself into the master role and become the authentication initiator.”

In addition to urging companies to apply the necessary patches, the organization is recommending Bluetooth users to install the latest updates from device and operating system manufacturers.

The research team concluded: “The BIAS attacks are the first, uncovering issues related to Bluetooth's secure connection establishment authentication procedures, adversarial role switches, and Secure Connections downgrades. The BIAS attacks are stealthy, as Bluetooth secure connection establishment does not require user interaction.”

So, this is patchable... but how many Bluetooth-enabled devices -- such as security systems, front door locks or other IoT gizmos -- will never be updated or patched?
https://francozappa.github.io/about-bias/publication/antonioli-20-bias/antonioli-20-bias.pdf

Is eBay port scanning their user's computers?
When I saw the headline “eBay port scans visitors' computers for remote access programs” I thought, huh?!?! But it turns out that's not really what's going on, kinda.

What IS going on, is that when a user goes to the eBay website, to in the process of bring up the eBay page, some Javascript named “check.js” runs on their own browser to internally probe 14 specific ports on that PC (at 127.0.0.1, the localhost IP) to see whether any of a set of well known remote control apps might be running and listening for incoming connections.

Program	**Ebay Name**	**Port**
Unknown	REF	63333
VNC	VNC	5900
VNC	VNC	5901
VNC	VNC	5902
VNC	VNC	5903
Remote Desktop Protocol	RDP	3389
Aeroadmin	ARO	5950
Ammyy Admin	AMY	5931
TeamViewer	TV0	5939
TeamViewer	TV1	6039
TeamViewer	TV2	5944
TeamViewer	TV2	6040
Anyplace Control	APC	5279
AnyDesk	ANY	7070

(Table from: https://www.bleepingcomputer.com/news/security/ebay-port-scans-visitors-computers-for-remote-access-programs/)

Lawrence Abrams, who covered this issue on his Bleeping Computer site wrote:

> “As the port scan is only looking for Windows remote access programs, it is most likely being done to check for compromised computers used to make fraudulent eBay purchases. In 2016, reports were flooding in that people's computers were being taken over through TeamViewer and used to make fraudulent purchases on eBay. As many eBay users use cookies to automatically login to the site, the attackers, who were able to remote control the computer, were able to access eBay to make purchases. It got so bad that one person created a spreadsheet to keep track of all the reported attacks. Many of them reference eBay.
>
> The script being used for fraud detection is further confirmed by Dan Nemec's great write-up, where he traced it to a fraud detection product owned by LexisNexis called ThreatMetrix. As part of ThreatMetrix's description, they discuss how they detect and protect sites from Remote Access Trojans (RATs). ThreatMetrix's product page explains: “Malware protection helps businesses mitigate the risk by being protected from Man-In-The-Browser (MITB), Remote Access Trojan (RAT), high velocity/ frequency bot attacks to low-and- slow attacks mimicking legitimate customer behavior, ransomware, key logging attempts, etc,”
>
> While the programs being scanned for are all legitimate, some of them have been used as RATs in phishing campaigns. Regardless of the reasons, port scans like this are intrusive and not something that many users would want to happen when visiting a site.

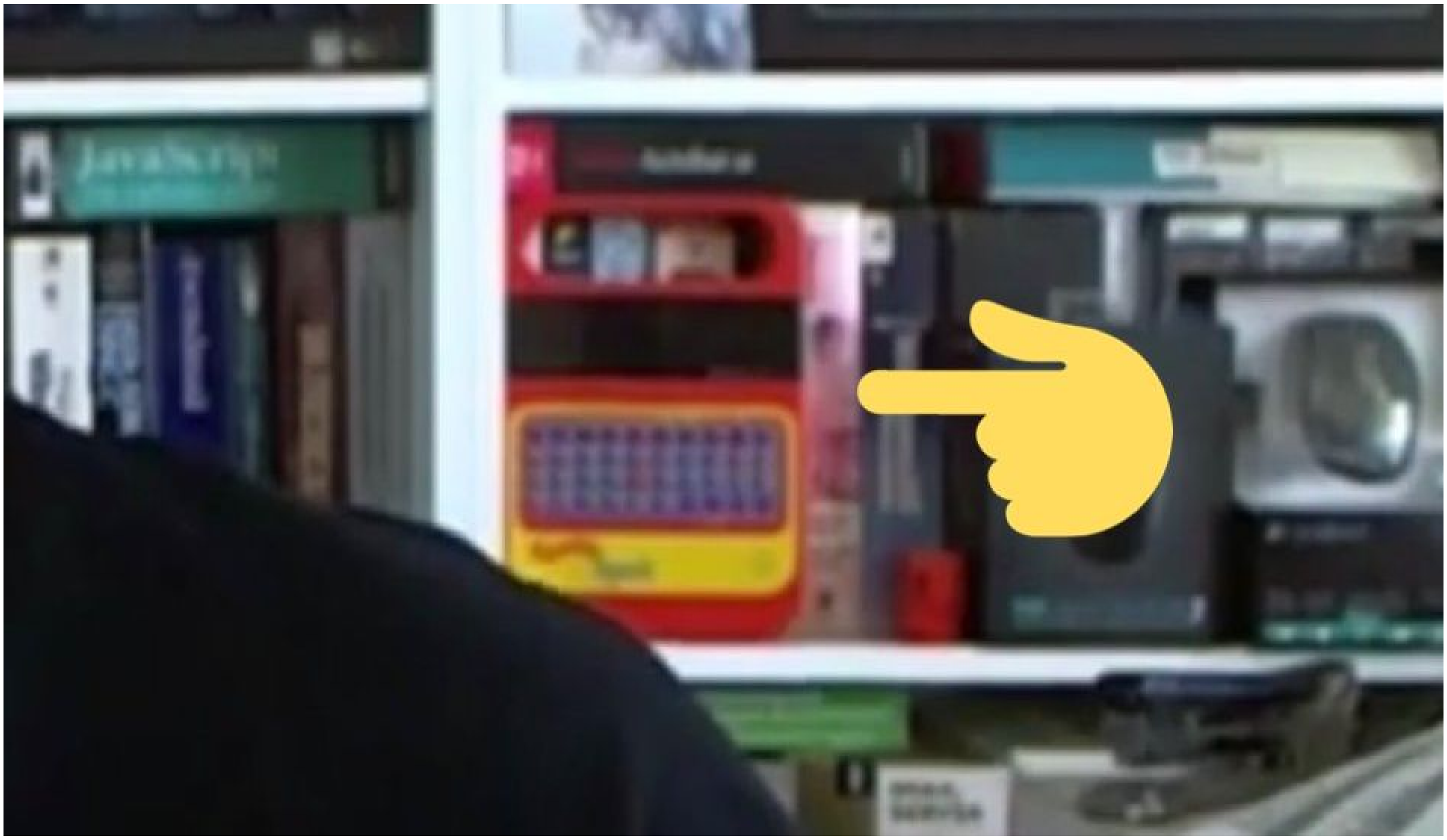
Closing The Loop

Igor Lima @igorlimatweets
Replying to @SGgrc
Loved the WiFi history Steve! Really appreciate the detail you provided, especially MiMo beam forming and collision detection. Please continue proving such historical context in future episodes!

Brian Helman @BrianHelman
I listened to the latest Security Now today. Three comments:. I didn't know there was an 802.11 wireless implementation - I always thought 802.11a was first. Doesn't that make 11ax v7 though (11, 11a, 11b, 11g, 11n, 11ac, 11ax)? 2nd, is I'd have loved to hear why they picked the names the way they did instead of 11, 11a, 11b, 11c... Lastly, you left out the biggest advancement with 11ax-unless I missed it (that we don't use because IoT manufacturers lag so far behind):. OFDMA (Orthogonal Frequency Division Multiple Access) allowing sub-channelization to reduce data rates to sub-2 Mbps. This is HUGE! It keeps low-bandwidth devices from hogging full channels freeing up space for devices that need the bandwidth.

Liron Amitzi / @amitzil
@SGgrc listening to you talking about DOH and wondering: if ISPs start having their own DOH enabled DNS, wouldn't they still be able to monitor us? Even if the transport is encrypted, they own the DNS servers. Am I missing something?

Classy Gay INFJ / @gayinfj
@SGgrc Hi Steve, can you tell me what this device is? It looks very familiar. Thanks!...



Stuart Donaldson @stuartdonaldson
Hey Steve! You messed me up. You turned me on to Peter F. Hamilton. I just finished up the audiobook Naked God, the last book in the Nights Dawn Trilogy. Now I have 6 weeks of Security Now to catch up on. Stay safe!

[My recommendation for your next read, Stuart, is Fallen Dragon. It's fabulous and it's a rare Hamilton stand-alone novel. Then you MUST read Pandora's Star and Judas Unchained.]

Miscellany

"Bosch" (on Prime)
Six seasons of really terrific, pitch-perfect, hardened police detective writing.

"Perry Mason" coming to HBO June 22nd.
Matthew Rhys as "Perry Mason"
Tatiana Maslany as "Sister Alice"

Contact Tracing Apps R.I.P.

Software-based contact tracing is doomed
Why? Academics who have modeled the system have determined that to be effective, **80%** of all smartphone users would need to voluntarily opt-in to using the app... and that's never going to happen.

As we saw, the instant the Apple/Google initiative was announced, both the non-technical and, sadly, the technical press went berserk over the privacy implications. Even highly technical individuals, who should have known better, spoke out with errant, frightening and unfounded warnings before they understood how the system worked. This podcast looked at the system's technology carefully and understood exactly what and why the Apple/Google team had designed it as they did. We found that the API itself absolutely protects the user's privacy.

But, in practice, that doesn't matter at all. For one thing, as we've discussed since, health officials really do have a need to collect real-time geographical location data as part of a workable system. Adding "where you were when it happened" would go a long way toward making up for a lack of pervasive use of an application. For example, if only a few people in a large gathering were app-enabled, and it was determined from the app that that was the most likely infection event, then a call could be put out for other non-app-enabled people who were also present at that event to take the necessary precautions.

As we noted previously, the importance of also knowing **where**, which Apple and Google scrupulously avoided using, has already occurred to the state of Utah, who has created a much more useful solution which is also -- necessarily -- much more invasive, even though it was thoughtfully designed with things like immediate user-deletion of all data and short-term self-expiring location data.

The simple truth is, a short-term sacrifice of privacy **is** required for spreading events to be located and managed. Even fully human-mediated contact tracing is, by definition, a short-term sacrifice of privacy. Someone whom you've never met and don't know anything about, needs to interview you to determine everything you're willing to share about where you've been, what you've done, and who you've been in contact with for the previous two weeks. Tell me that's not a massive imposition on one's privacy. Of course it is. But that's what's necessary.

Google says that people are constantly clearing their web browser histories. And that's just cyber. Many people apparently **really** don't want anyone else to know where they've been and what they've been up to. At least when interviewed by a human contact tracer, someone can choose to "elide" anything they're embarrassed to share. But you can't do that with an app. So how many people are going to voluntarily install what amounts to spyware?

As we well know, many people have an inherent mistrust of the government and its motives. We've already seen people worrying that this might just be the start of more pervasive monitoring with statements like "If they are allowed to do this, they'll always want more and why would they ever want to stop?" and so forth.

So, no. It was a noble idea. I loved the cleverness of the technology. But it's clear that as a voluntary initiative it's never going to get off the ground.