Security Now! #732 - 09-17-19

SIM Jacking

This week on Security Now!

This week we continue following the DoH story, which we begin discussing two weeks from now, in the future, as a result of a rip in the space-time continuum. We also look at recent changes to Chrome 77 and the forthcoming Chrome 78, the already compromised though not-yet-released iOS 13.0, Mozilla Firefox's new browser VPN offering, and a look back at last Tuesday's Patch Tuesday. We take note of Chrome's Remote Desktop feature, cover another serious Exim email server problem, handle a bit of miscellany... and conclude with an examination of a serious vulnerability affecting essentially ALL smartphone users known as "SIM Jacking."

Security News

Chrome Follows Mozilla to DoH with a Twist
Google has announced that they, too, will soon be performing a trial of DNS-over-HTTPS (DoH) in the Chrome beta 78 releasing this Thursday, September 19th. (We're currently at release 77).

What's interesting is that, rather than having Chrome pre-configured with a default DoH server like their own, Google will, instead, attempt to PRESERVE whatever DNS the user already has chosen. I LOVE this idea, and it would be =VERY= cool if they were to probe the user's currently-selected DNS for DoH support, test it locally, then switch to it. But apparently that's a bit too aggressive, at least initially. So they will be doing this ONLY if their user has already configured their DNS to one of six providers:

Cleanbrowsing	Cloudflare	DNS.SB
Google	OpenDNS	Quad9

So, initially, for a small group of users running Chrome 78 beta. Google will be running an experiment that checks if their DNS provider is part of that short list of known DoH-compatible providers. If a user's DNS provider is part of the list, Chrome will automatically upgrade to that provider's DoH server to perform DNS resolution. And if they are not already using one of those servers, nothing will change. This will be happening on all Chrome platforms other than Linux and iOS. And on Android 9 and later, if a user has already configured a DNS-over-TLS provider, Chrome will use that instead and only use the ones from their list if there's an error.

By cleverly leaving the DNS provider as-is and only upgrading to the provider’s equivalent DoH service, the user experience should remain the same. For instance, any malware site protections or parental control features offered by the DNS provider would continue to work. If DoH fails, Chrome will revert to the provider’s regular DNS service. And any of these early-adopter users will be able to opting out of the experiment by disabling the flag at chrome://flags/#dns-over-https.

On the Mozilla side, and thanks to the time machine we used last Saturday to record podcast #734, I happen to know that two weeks from now, on our "Joy of Sync" podcast #734, we'll be discussing Mozilla's own move to begin experimenting with enabling DoH-by-Default for their users. But it turns out that as news of Mozilla's plans spread, which, by the way, I was unaware of two weeks from now due to a temporal paradox, Mozilla will have since received some pushback from Linux distro maintainers and some network admins. In an example quoted by BleepingComputer, OpenBSD developer Peter Hessler tweeted that OpenBSD has disabled DoH in their Firefox package in the current and future releases as "sending all DNS traffic to Cloudflare by default is not a good idea."

Kristian Köhntopp, a senior scalability engineer, stated that Mozilla is about to "break DNS" because Cloudflare will be used for DNS resolution over what was assigned by system administrator. This will leak the names of all the websites you visit in a corporate environment to Cloudflare.

Want to enable DoH in Chrome right now?
You can, right now, if you wish.

Chrome currently lacks any user-interface for configuring this. But it dutifully obeys launch-time startup parameters. So, for example, in windows you would modify the startup link to add the command parameters:

```
--enable-features="dns-over-https<DoHTrial"
--force-fieldtrials="DoHTrial/Group1"
--force-fieldtrial-params="DoHTrial.Group1:server/https%3A%2F%2F1.1.1.1%2Fdns-query/method/POST
```

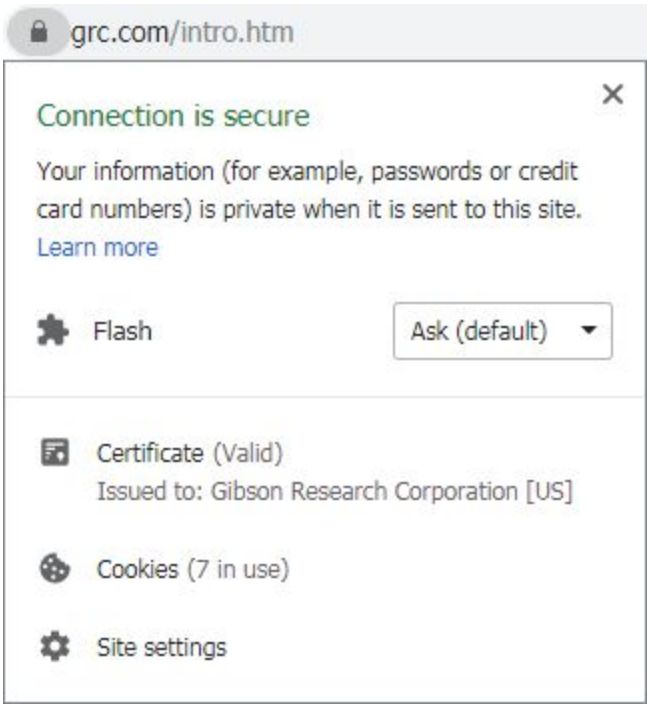
For a current list of DoH servers from the DNS-over-HTTPS page on Github:
https://github.com/curl/curl/wiki/DNS-over-HTTPS#publicly-available-servers

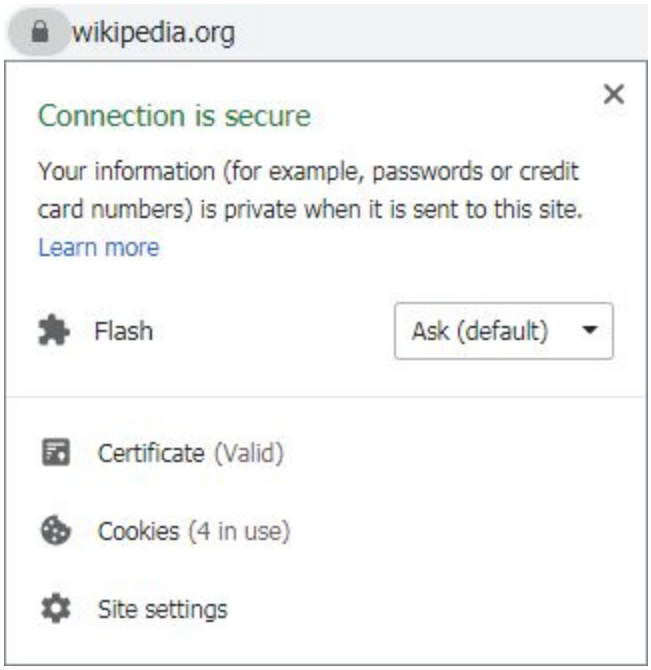
If Chrome is already running, shut it down and relaunch it with the enhanced link.

To verify that DoH support is now working in Chrome, access https://1.1.1.1/help. On the status line "Using DNS over HTTPS (DoH)" the site should display "Yes."

And a tip of the hat to ZDNet for that nifty tip!

Chrome drops all URL bar EV display:
Meanwhile, there's no waiting to experience Chrome's deprecation of all display of Extended Validation certs:





Until this update, GRC's Extended Validated company name would have been proudly displayed in the URL bar. But no longer. Note, also, that the "www's" in front of BOTH domain names has been (controversially) suppressed.

Also, for the record, this release of Chrome 77 fixed a total of 36 security vulnerabilities; 1 Critical, 8 High, 17 Medium & 10 hardly-worth-mentioning.

Here comes iOS "Lucky" 13!

We're supposed to get it this Thursday the 19th. And I'm excited for one feature in particular: I LOVE "swipey" phone keyboards, but I've never been happy with the 3rd-party iOS add-on keyboards... And I've tried living with them all. Gboard, Swype, SwiftKey, and so on. They each misbehave in various ways... Most often by failing to deploy on-screen when they're needed. So when I heard that iOS 13 would **finally** have that built-in I was finally excited about a new iOS release.

But it turns out that iOS 13 will be jumping to 13.1 very soon after release 13.0.0, since a headline grabbing Lock Screen Bypass bug is already known to exist, and to still exist in the "Gold Master" version of iOS 13 that has already been loaded into the many hundreds of thousands of iPhones in shipping containers out there on the high seas.

So... iOS 13 will contain a vulnerability that allows anyone to bypass the lockscreen protection to access sensitive information. Spanish security researcher Jose Rodriguez has revealed that he discovered a lockscreen bypass bug in iOS 13 that allowed him to access the full list of Contacts on his iPhone—and every piece of information saved on them. Jose discovered the newly introduced lockscreen bypass bug on his own iPhone running iOS 13 beta version and reported it to Apple exactly two months ago on July 17. However, that was apparently too late for Apple to do anything about it... so the bypass remains working in the Gold Master (GM) version of iOS 13... Which we'll all be able to obtain this coming Thursday, September 19th.

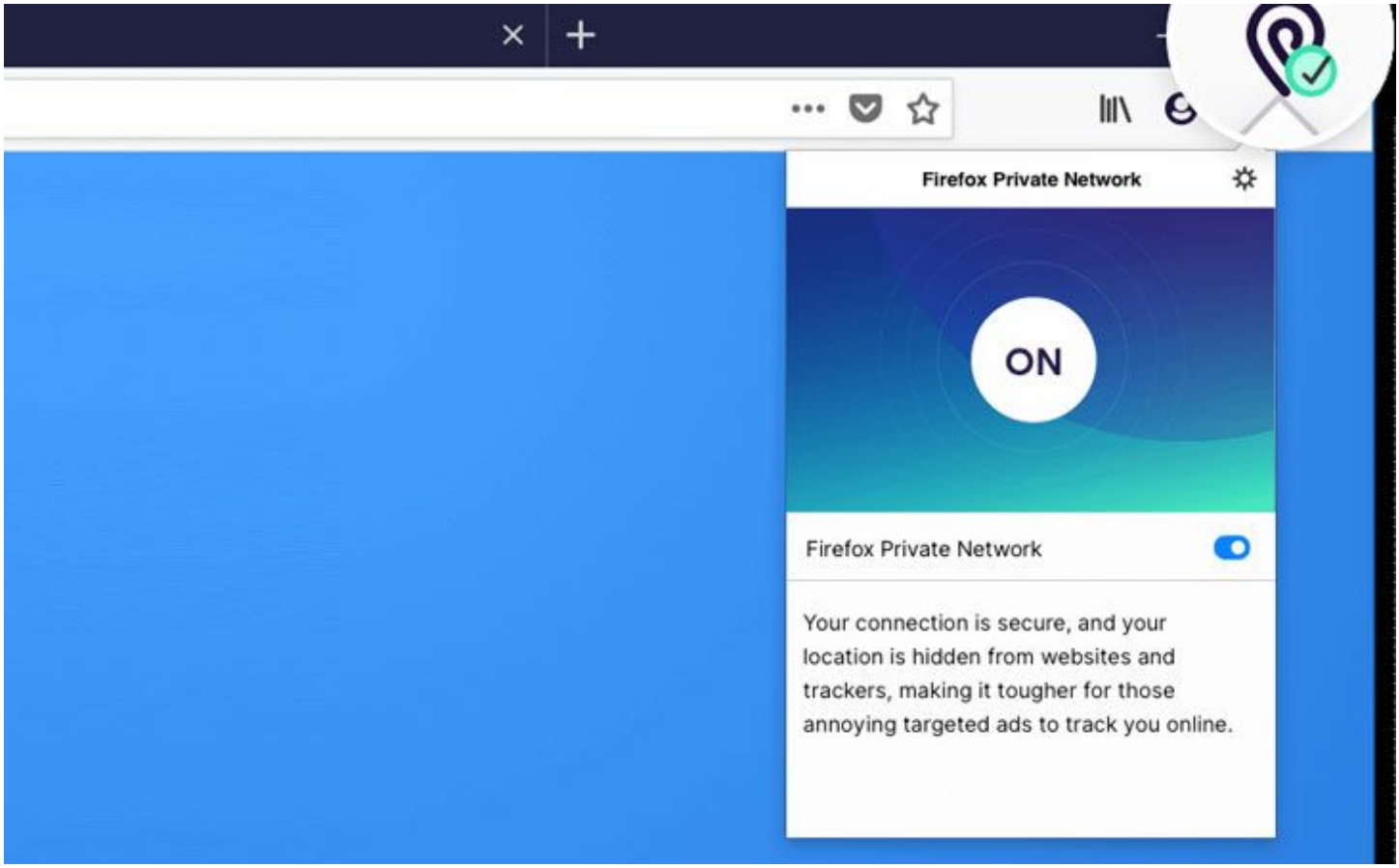
The Lockscreen bug is like those we've seen before, where someone having physical access to a target iPhone is able to trick the phone into granting them access to the full list of stored Contacts, as well as detailed information for each individual contact including their names, phone numbers, and emails—using a FaceTime call.

This is also similar to the one Jose discovered last year in iOS 12.1, just a few hours after Apple released iOS 12.1. It allowed anyone to bypass the phone's lockscreen using the built-in VoiceOver feature.

The bug requires activating a FaceTime call on a target's iPhone and then accessing Siri's voiceover support feature to obtain access to the contact list—and all of the information saved there.

This problem won't exist for long, since it's expected to be patched in iOS 13.1 which is expected to begin trickling out to the public eleven days later on Monday September 30th. So, if this really worries you, you could disable automatic updates until October 1st and jump right over to v13.1.

Mozilla Launches 'Firefox Private Network' VPN Service as a Browser Extension
So, Leo... In another of those time-travel paradoxes... In the future, two weeks from now, during episode #734, I have it on very good authority that I will mention that Cloudflare is launching a mobile device oriented VPN. Whereupon you inform me of the just-breaking news from the past, that Mozilla is also launching a privacy-focused VPN service. That takes me by surprise since, once again, we've been messing around with the space-time continuum for the benefit of our faithful listeners. We'll do anything for our listeners. So, due to the time warp, today, two weeks earlier than then, I am *now* fully up to speed on Mozilla's announcement, even though I'll know nothing about it two weeks from now ago. In any event...



Mozilla has, indeed, officially launched a new privacy-focused VPN service, called Firefox Private Network. It runs as a browser extension to encrypt all of Firefox users' online activity and limiting what websites and advertisers know about Firefox users.

The Firefox Private Network service is currently undergoing beta testing and is available only to desktop users in the United States as part of Mozilla's recently reborn "Firefox Test Pilot" program that lets users try out new experimental features before they're officially released. The Firefox Test Pilot program was initially launched by Mozilla three years ago but was shut down in January this year. Mozilla has decided to bring an updated and changed program back.

Marissa Wood, the vice president of product at Mozilla said: "The difference with the newly relaunched Test Pilot program is that these products and services may be outside the Firefox browser, and will be far more polished, and just one step shy of general public release." So this newly announced "Firefox Private Network" is the relaunched Test Pilot program's first new project.

As we would expect of any Virtual Private Network service, the Firefox Private Network masks its users' IP address from third-party online trackers and protects sensitive information, like the website you visit and your financial information, when using public Wi-Fi. It's important to note, however, that all by itself it's not offering anti-tracking protection since these days that's primarily done from within the browser and is not dependent upon IP addresses which change as mobile users switch hotspots, cellular regions, home and office.

Which is not to suggest that it's not super-useful as a VPN. Mozilla says its Firefox Private Network "provides a secure, encrypted path to the web to protect your connection and your personal information anywhere, and everywhere you use your Firefox browser." So a built-in facility it would bring easy-to-use and useful VPN services to many people who might not otherwise go to all of the trouble to setup a VPN when browsing.

The Firefox Private Network VPN encrypts and funnels Internet browsing activity (but ONLY Internet browsing activity) through a collection of remote proxy servers, thereby masking its user's actual location and blocking third parties, including government and your ISP, from snooping on browser traffic... at least until it emerges from the other end of the VPN connection.

Interestingly, the proxy servers used by the Firefox Private Network extension are provided by Cloudflare. Mozilla and Cloudflare have agreed to provide "strong privacy controls" to limit what data Cloudflare may collect and for how long it may store any data. We have often talked about VPN services and one of the many issues is whether they log and, if so, what log retention policies they follow.

Cloudflare has stated: "Cloudflare only observes a limited amount of data about the HTTP/HTTPS requests that are sent to the Cloudflare proxy via browsers with an active Mozilla extension. When requests are sent to the Cloudflare proxy, Cloudflare will observe your IP address, the IP address for the Internet property you are accessing, source port, destination port, timestamp and a token provided by Mozilla that indicates that you are a Firefox Private Network user. We call this "Proxy Data." All Proxy Data will be deleted within 24 hours."

How To Sign Up For Firefox VPN Service

Firefox Private Network currently works only on desktop Firefox, but it is slated to be available for mobile Firefox users as well, once the VPN exits beta. Although the Firefox Private Network service is currently free, Mozilla hinted that the company is exploring the possibility of adding value commercial service pricing options for the service in the future to make the service self-sustaining.

Anyone with a Firefox account residing in the United States may experiment with the Firefox VPN service for free by signing up on the Firefox Private Network website. Once installed on the desktop, the Firefox Private Network extension will add a toggle on Firefox's toolbar to allow it to be easily turned on or off at any time.

https://private-network.firefox.com/

Patch Tuesday Redux

Last Tuesday was September's Patch Tuesday... and it did not disappoint, providing fixes for a whopping 79 vulnerabilities, 17 of which were considered to be critical.

Among the many, we received a further fix for last month's very worrisome CTF flaws discovered and explored in excruciating detail by Google's Tavis Ormandy. As we'll recall, Tavis discovered how unprivileged attackers could launch their attack code with elevated privileges. And we've long since been disabused of the notion that elevation of privilege is nothing to worry about.

During the previous Patch Tuesday in August Microsoft dealt with one of the related vulnerabilities (CVE-2019-1162), but at the time indicated that there was still more to come. So as part of this month's offerings Microsoft has released another fix for this domain of flaws titled "CVE-2019-1235 | Windows Text Service Framework Elevation of Privilege Vulnerability".

Quoting Microsoft:

> An elevation of privilege vulnerability exists in Windows Text Service Framework (TSF) when the TSF server process does not validate the source of input or commands it receives. An attacker who successfully exploited this vulnerability could inject commands or read input sent through a malicious Input Method Editor (IME). This only affects systems that have installed an IME.
>
> To exploit this vulnerability, an attacker would first have to log on to the system. An attacker could then run a specially crafted application that could exploit the vulnerability and take control of an affected system.
>
> The security update addresses this vulnerability by correcting how the TSF server and client validate input from each other.

As we know, the other continuously-troubled area of Windows has been Remote Desktop. So in this month's patch batch we get FOUR more fixes to the RDP's client side which fix more of those "not really such a big problem" discoveries that Microsoft has apparently been making now that they have put renewed focus upon RDP recently. All four were remote code execution vulnerabilities, but they did require a user to connect to a malicious RDP server.

Again, in Microsoft's words:

> Remote code execution vulnerabilities exist in the Windows Remote Desktop Client when a user connects to a malicious server. An attacker who successfully exploited this vulnerability could execute arbitrary code on the computer of the connecting client. An attacker could then install programs; view, change, or delete data; or create new accounts with full user rights.
>
> To exploit this vulnerability, an attacker would need to have control of a server and then convince a user to connect to it. An attacker would have no way of forcing a user to connect to the malicious server, they would need to trick the user into connecting via social engineering, DNS poisoning or using a Man in the Middle (MITM) technique. An attacker could also compromise a legitimate server, host malicious code on it, and wait for the user to connect.

We should also note that Microsoft has said that three of the four vulnerabilities have been publicly disclosed and that two of them have known exploits.

One of those 17 critical vulnerabilities fixed this month is CVE-2019-1208, a VBScript Remote Code Execution Vulnerability. So it's just as well that Microsoft will be throwing in the towel on JBScript in favor of the now-industry-wide standard ECMAScript. Still, as we all too well know, legacy usage and dependency will continue forever.

So, Microsoft wrote:

> A remote code execution vulnerability exists in the way that the VBScript engine handles objects in memory. The vulnerability could corrupt memory in such a way that an attacker could execute arbitrary code in the context of the current user. An attacker who successfully exploited the vulnerability could gain the same user rights as the current user. If the current user is logged on with administrative user rights, an attacker who successfully exploited the vulnerability could take control of an affected system. An attacker could then install programs; view, change, or delete data; or create new accounts with full user rights.

Chrome Remote Desktop
https://remotedesktop.google.com/home
https://remotedesktop.google.com/support

"Your desktop anywhere - Securely access your computer from your phone, tablet or another computer. It's fast, simple and free."

"Give & get support - Get remote support for your computer, or give remote support to someone else."

There are also iOS and Android clients:
Securely access your computer from your iOS device. It's fast, simple and free.

- Download the Chrome Remote Desktop app from the Chrome WebStore on the computer you want to access remotely.
- Install Chrome Remote Desktop software and follow the instructions to complete setup.
- On your iOS device, open the app and tap on any of your online computers to connect.

Wikipedia:
https://en.wikipedia.org/wiki/Chrome_Remote_Desktop

Chrome Remote Desktop is a remote desktop software tool developed by Google that allows a user to remotely control another computer through a proprietary protocol developed by Google unofficially called "Chromoting". It transmits the keyboard and mouse events from one computer to another, relaying the graphical screen updates back in the other direction, over a network. This feature therefore consists of a server component, for the host computer, and a client component on the computer accessing the remote computer.

The Chrome Remote Desktop client was originally a Chrome extension from the Chrome Web Store requiring Google Chrome; the extension is deprecated, and a web “portal” is available at

https://remotedesktop.google.com. The browser must support WebRTC and other unspecified "modern web platform features". The client software is also available on Android and iOS.

If the computer is to host remote access, such as for remote support, a server package is downloaded and Chrome must be used. This is available for Microsoft Windows, macOS, Linux and Chrome OS.

The Chrome Remote Desktop remote assistance mode has a variation, allowing a permanent, pre-authorised connection to a remote computer, designed to allow a user to connect to another one of their own machines remotely. In contrast, Remote Assistance is designed for short-lived remote connections, and requires an operator on the remote computer to participate in authentication, as remote assistance login is via PIN passwords generated by the remote host human operator. This method of connection will also periodically block out the control from the connecting user, requiring the person on the host machine to click a button to "Continue sharing" with the connected client.

The protocol uses VP8 video encoding to display the remote computer's desktop to the user with high performance over low bandwidth connections. Under Windows, it supports copy-paste and real-time audio feed as well, but lacks an option to disable sharing and transmission of the audio stream. The software is limited to 100 clients. Attempting to add further PCs after reaching 100 will result in a "failed to register computer" error.

EXIM eMail servers are in trouble again

Recall that three months ago, the Exim maintainers patched a severe remote command execution vulnerability (CVE-2019-10149), that was being actively exploited in the wild to compromise vulnerable servers. That was the weird "takes a week to do" data trickle exploit that placed shell commands into the reply to address, which the server would execute once it finally got tired of waiting and timed-out.

This one is way worse: Exim maintainers have released Exim version 4.92.2 after publishing an early warning two days before hand to give sys ops an early warning heads-up on its upcoming security patches which affect ALL versions of the email server software up to and including the then-latest 4.92.1.

Just to remind everyone why this matters, Exim is a widely used, open source mail transfer agent for our Unix-like operating systems (Linux, Mac OSX, Solaris, etc.) which it behind nearly 60% of the Internet's email servers today.

The new vulnerability is CVE-2019-15846 and it affects Exim servers that accept TLS connections. Of course, that's now considered best practice. But the flaw that was discovered allows attackers to obtain root-level access to the system "by sending an SNI (Server Name Indication) ending in a backslash-null sequence during the initial TLS handshake." Whoopsie!

As we know, SNI is an extension of the TLS protocol which allows servers to host multiple TLS certificates on a single IP. It allows a connecting client to tell the server, in the first TLS packet, which server certificate it wants to use. And, according to the Exim team, since the vulnerability does not depend on the specific TLS library being used by the server, both GnuTLS and OpenSSL

are affected.

And though the "drop-in" default configuration of the Exim mail server software does not have TLS enabled -- since TLS certs need to be supplied and configured -- some operating systems DO bundle Exim with the vulnerable feature enabled by default.

So, just a heads up to anyone listening here who is responsible for any Exim installations. You'll definitely want to update to v4.92.2 immediately.

Miscellany

Firefox: browser.tabs.unloadOnLowMemory / TRUE == a WIN!!
Upon enabling the switch Firefox's memory consumption *immediately* dropped

Enabling Ransomware Prevention in Windows 10:
We've been talking a lot about Ransomware recently because... Ransomware.

So I wanted to note something that someone tweeted in my direction, which was that Windows 10's Windows Defender includes a "Ransomware Protection" which enables various protections against ransomware. Presumably because it might bite someone by falsely triggering, this feature is always disabled by default. But these days, with ransomware attacks increasing, I would recommend that Win10 users flip this switch "ON."

Windows 10's Ransomware Protection involves Controlled Folder Access and Ransomware Data Recovery.

Controlled Folder Access will -- DEFINITELY -- cause you some annoyance while it's being trained... Which is why it's definitely OFF by default. By default it blocks many of Microsoft's own apps, like IE and Edge. But once it has been trained it will definitely be useful (until and unless it, too, is bypassed) to keep unknown baddies from touching your user-data-filled directories.

The second component is Ransomware Data Recovery, which automatically syncs those same common user-data directories with your Microsoft OneDrive to backup those files. Ransomware victims with this feature enabled can then use OneDrive to recover their files if they ever become encrypted by ransomware.

Note that the use of any 3rd-party A/V software will disable Windows Defender's real-time protection and so, too, the real-time Ransomware Protection.

Assuming that you're only using Windows Defender, in the Control Panel select "Update & Security" then "Windows Security."

- From "Windows Security", click on Virus & Threat Protection option.
- Scroll down and locate Ransomware Protection and click on the Manage ransomware protection option.

- On the next page is a brief description of Controlled Folder Access and a toggle to enable it. Turn it on.
- To enable Ransomware Protection with Controlled Folder Access enabled, login to OneDrive.
- Controlled Folder Access can be configured to monitor and block any chosen directories from malicious programs.

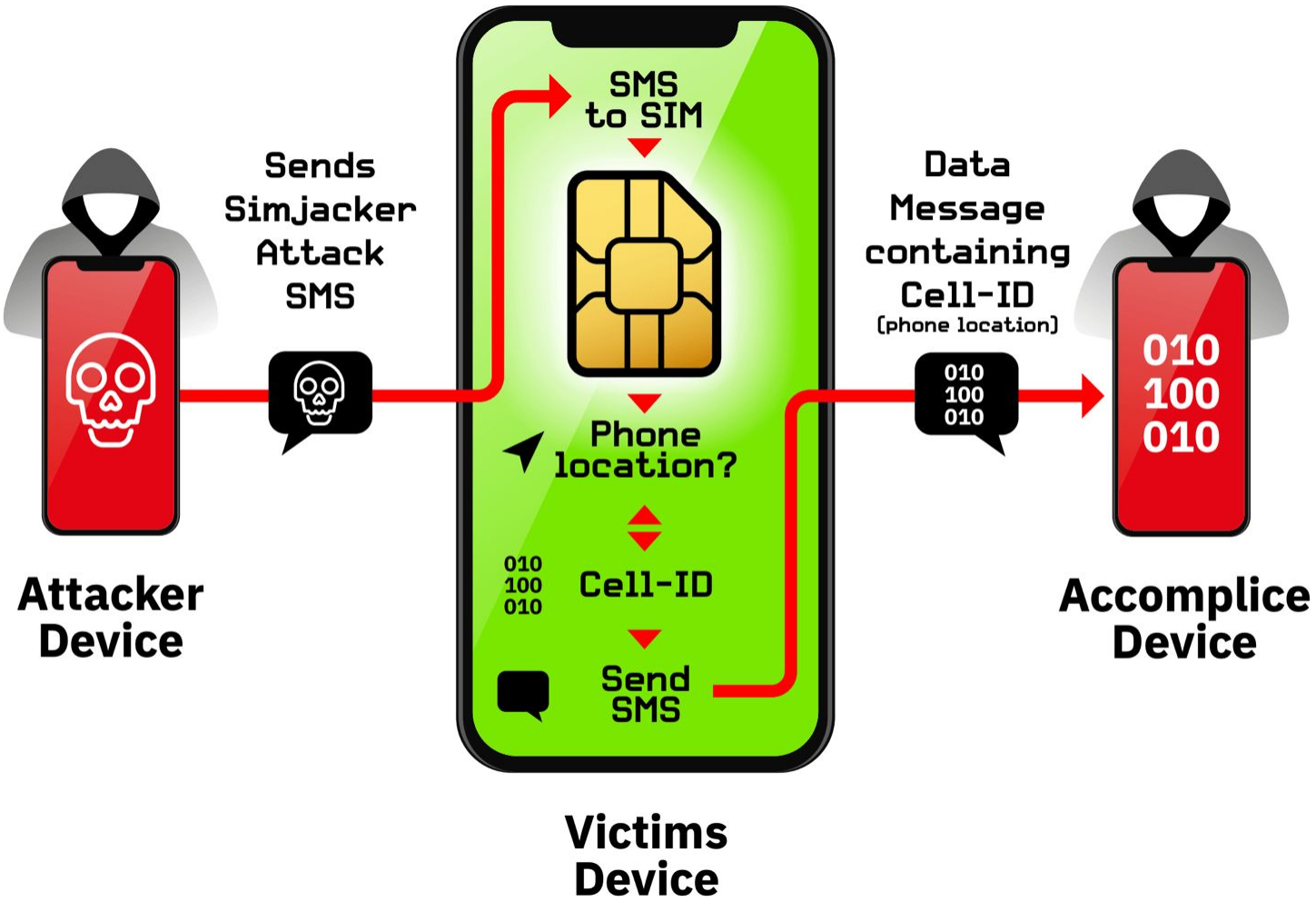
Just be aware that CFA will be =EXTREMELY= noisy until it has been trained about all of the programs which you WANT to give access to. I have extensive experience with it since my Windows SQRL client's self-installer really gets its attention. And I was amazed that Microsoft didn't even pre-load their own browsers into the permitted list.

Simjacker

The SimJacker icon above is animated & wonderful. Watch & it'll repeat: https://simjacker.com/

https://simjacker.com/downloads/briefingpapers/AdaptiveMobile-Security_Simjacker-Briefing-Paper.pdf

Introducing: "SIMjacker" … a new SIM card flaw -- discovered being actively exploited in the wild, which allows attackers to hijack any phone just by sending it an SMS message…



Overview:

AdaptiveMobile Security have uncovered a new and previously undetected vulnerability and associated exploits, called Simjacker. This vulnerability is currently being actively exploited by a specific private company that works with governments to monitor individuals.

"Simjacker" and its associated exploits is a huge jump in complexity and sophistication compared to attacks previously seen over mobile core networks. The main Simjacker attack involves an SMS containing a specific type of spyware-like code being sent to a mobile phone, which then instructs the SIM Card within the phone to 'take over' the mobile phone to retrieve and perform sensitive commands. The location information of thousands of devices was obtained over time without the knowledge or consent of the targeted mobile phone users. During the attack, the user is completely unaware that they received the attack, that information was retrieved, and that it was successfully exfiltrated. However the Simjacker attack can, and has been extended further to perform additional types of attacks.

Simjacker has been further exploited to perform many other types of attacks against individuals and mobile operators such as fraud, scam calls, information leakage, denial of service and espionage. AdaptiveMobile Security Threat Intelligence analysts observed the hackers vary their attacks, testing many of these further exploits. In theory, all makes and models of mobile phone are open to attack as the vulnerability is linked to a technology embedded on SIM cards. The Simjacker vulnerability could extend to over 1 billion mobile phone users globally, potentially impacting countries in the Americas, West Africa, Europe, Middle East and indeed any region of the world where this SIM card technology is in use.

We are quite confident that this exploit has been developed by a specific private company that works with governments to monitor individuals. AdaptiveMobile Security has been working closely with their customers and the wider industry; including both mobile network operators and SIM card manufacturers to protect mobile phone subscribers. We have blocked attacks and are committed to using our global threat intelligence to build defences against these new sophisticated attacks that are circumventing current security measures.

Background

AdaptiveMobile Security's industry leading TIU team has detected unusual activity over messaging and signalling bearers in specific customers, occurring over a long period of time. Specific, targeted subscribers were receiving SMS messages that were causing them to send another SMS with location/terminal information, without any notification or knowledge. Subsequent deeper investigation revealed a vulnerability that allowed almost every single mobile device in affected operators to be open to manipulation. We believe this vulnerability has been exploited for at least the last two years by a highly sophisticated attacker group.

So, how does this work?

The main Simjacker attack involves a SMS containing a specific type of spyware-like code being sent to a mobile phone, which then instructs the SIM Card within the phone to 'take over' the mobile phone to retrieve and perform sensitive commands. The attacks exploit the ability to send SIM Toolkit Messages and the presence of the S@T Browser on the SIM card of vulnerable subscribers. (The S@T Browser is normally used for browsing through the SIM card.) The Attack messages use the S@T Browser functionality to trigger proactive commands that are sent to the

handset. The responses to these commands are sent back from the handset to the SIM card and stored there temporarily. Once the relevant information is retrieved from the handset, another proactive command is sent to the handset to send an SMS out with the information.

//////////////

What the Hell??? An "S@T Browser" of some kind on SIM cards? So, at this point I jump over to Wikipedia to get a bit more background… and that just makes it worse! Wikipedia says:

> **SIM Application Toolkit** (**STK**) is a standard of the GSM system which enables the subscriber identity module (SIM card) to initiate actions which can be used for various value-added services. Similar standards exist for other network and card systems, with the **USIM Application Toolkit** (USAT) for USIMs used by newer-generation networks being an example. A more general name for this class of Java Card-based applications running on UICC cards is the **Card Application Toolkit** (CAT).
>
> The SIM Application Toolkit consists of a set of commands programmed into the SIM which define how the SIM should interact directly with the outside world and initiates commands independently of the handset and the network. This enables the SIM to build up an interactive exchange between a network application and the end user and access, or control access to, the network. The SIM also gives commands to the handset such as displaying menus and/or asking for user input.
>
> STK has been deployed by many mobile operators around the world for many applications, often where a menu-based approach is required, such as Mobile Banking and content browsing. Designed as a single application environment, the STK can be started during the initial power up of the SIM card and is especially suited to low level applications with simple user interfaces.
>
> In GSM networks, the SIM Application Toolkit is defined by the GSM 11.14 standard released in 2001. From release 4 onwards, GSM 11.14 was replaced by 3GPP TS 31.111 which also includes the specifications of the USIM Application Toolkit for 3/4G networks.

AdaptiveMobile Security explained that their global threat analytics system, allowed them to correlate the Simjacker sources with known malicious threat actors. As a result, they can state with a high degree of certainty, that the source is a large professional surveillance company, with very sophisticated abilities in both signalling and handsets. These types of companies exploit the fact that some mobile operators may incorrectly regard core network security as solved if they deploy a standard GSMA 'compliant' firewall.

So they have revealed the existence of the vulnerability and associated exploits that they call Simjacker. They believe this vulnerability has been exploited for at least the last 2 years by a highly sophisticated threat actor in multiple countries, primarily for the purposes of surveillance. Other than the impact on its victims, from their analysis, Simjacker and its associated exploits is a huge jump in complexity and sophistication compared to attacks previously seen over mobile core networks. It represents a considerable escalation in the skill set and abilities of attackers seeking to exploit mobile networks.

Here's an overview of Simjacker, how it works and who is potentially exploiting it, as well as why it is such a significant new type of attack.

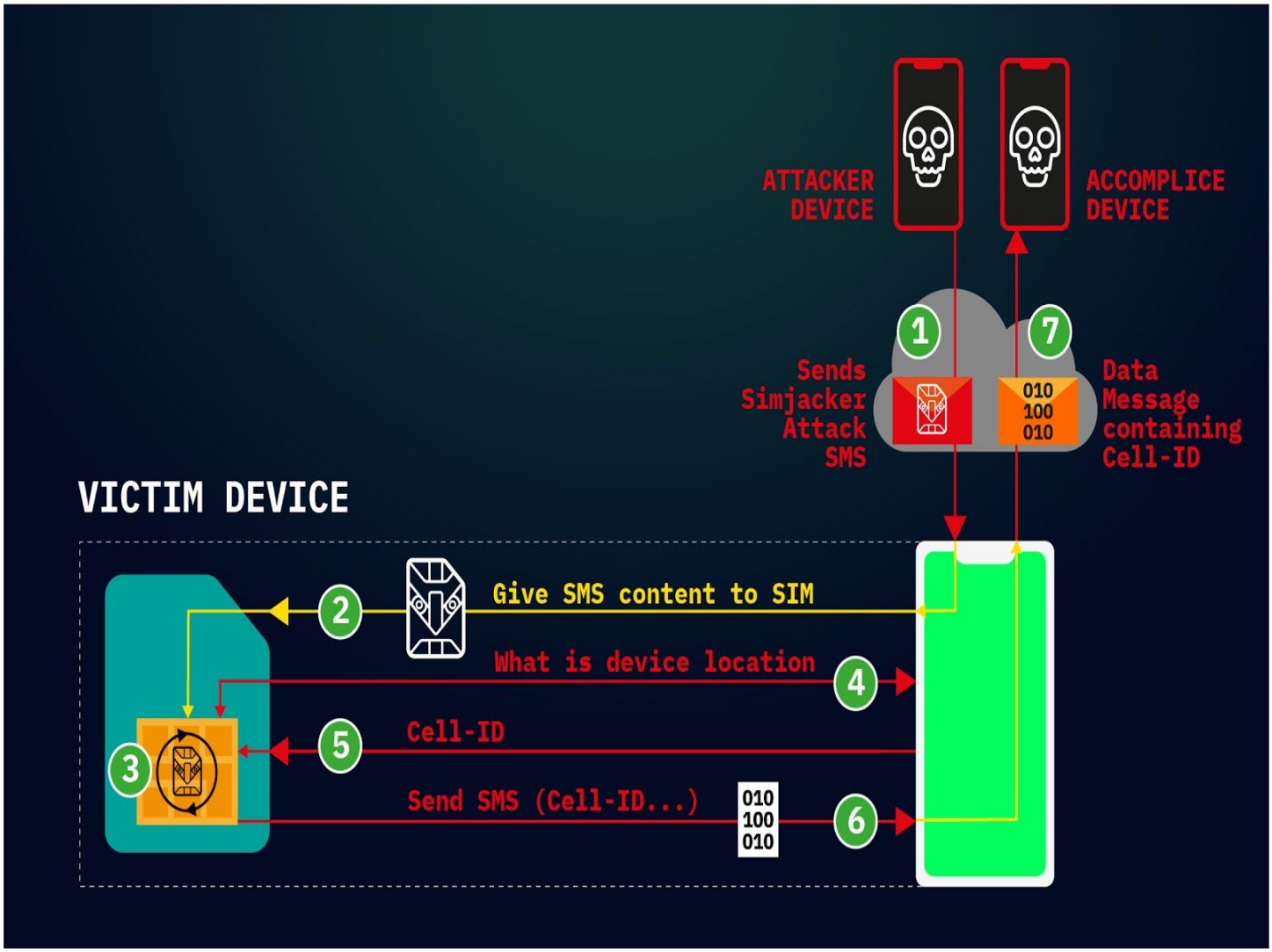
How it Works
At its simplest, the main Simjacker attack involves a SMS containing a specific type of spyware-like code being sent to a mobile phone, which then instructs the UICC (SIM Card) within the phone to 'take over' the mobile phone , in order to retrieve and perform sensitive commands.

The attack begins when a SMS - that we term the Simjacker 'Attack Message' - is sent to the targeted handset. This Simjacker Attack Message, sent from another handset, a GSM Modem or a SMS sending account (in other words, any source) contains a series of SIM Toolkit (STK) instructions, and is specifically crafted to be passed on to the device's SIM Card. For these instructions to work the attack exploits the presence of a particular piece of software, called the S@T Browser located on the SIM Card. Once the Simjacker Attack Message is received by the UICC, it uses the S@T Browser library as an execution environment on the SIM. From there it's able to trigger logic on the handset.

For the main attack observed, the Simjacker code running on the SIM requests location and specific device information (the IMEI) from the handset. Once this information is retrieved, the Simjacker code collates it and sends the combined information to a recipient number via another SMS (we call this the 'Data Message'), by again triggering logic on the handset. This Data Message is the method by which the location and IMEI information can be exfiltrated to a remote phone controlled by the attacker.

During the attack, the user is completely unaware that they received the SMS with the Simjacker Attack message, that information was retrieved, and that it was sent outwards in the Data Message SMS - there is no indication in any SMS inbox or outbox.

What makes this Attack work and why is it Special?
The attack relies both on these specific SMS messages being allowed, and the S@T Browser software being present on the UICC in the targeted phone. Specific SMS messages targeting UICC cards have been demonstrated before on how they could be exploited for malicious purposes. The Simjacker attack takes a different approach, and greatly simplifies and expands the attack by relying on the S@T Browser software as an execution environment. The S@T (pronounced sat) Browser – or SIMalliance Toolbox Browser to give it its full name – is an application specified by the SIMalliance, and can be installed on a variety of UICC (SIM cards), including eSIMs. This S@T Browser software is not well known, is quite old, and its initial purpose was to enable services such as getting your account balance through the SIM card. Globally, its function has been mostly superseded by other technologies, and its specification has not been updated since 2009, however, like many legacy technologies it is still been used while remaining in the background. In this case we have observed the S@T protocol being used by mobile operators in at least 30 countries whose cumulative population adds up to over a billion people, so a sizable amount of people are potentially affected. It is also highly likely that additional countries have mobile operators that continue to use the technology on specific SIM cards.



The Simjacker Attack Message could carry a complete malware/spyware payload because it contains a list of instructions that the SIM card is to execute. As software is essentially a list of instructions, and malware is 'bad' software, then this could make the Simjacker exploit the first real-life instance of spying malware sent within a SMS. Previous malware sent by SMS has been limited to sending links to malware, not the malware itself within a complete message.

Beyond Location

But the novelty and potential of Simjacker does not stop there. Retrieving a person's location is one thing, but by using the same technique, and by modifying the attack message, the attacker can instruct the SIM to execute a wide range of attacks. This is thanks to the fact that the attacker has similar access to the complete STK command set. Some example STK commands are:

- PLAY TONE
- SEND SHORT MESSAGE
- SET UP CALL
- SEND USSD
- SEND SS
- PROVIDE LOCAL INFORMATION
    - *Location Information, IMEI, Battery, Network, Language, etc*

- POWER OFF CARD
- RUN 'AT' COMMAND
- SEND DTMF COMMAND
- LAUNCH BROWSER
- OPEN CHANNEL
    - *CS BEARER, DATA SERVICE BEARER, LOCAL BEARER, UICC SERVER MODE, etc*
- SEND DATA
- GET SERVICE INFORMATION
- SUBMIT MULTIMEDIA MESSAGE
- GEOGRAPHICAL LOCATION REQUEST

By using these commands in our own tests, we were able to make targeted handsets open up web browsers, ring other phones, send text messages and so on. These attacks could be used to fulfil such purposes as

- Mis-information (e.g. by sending SMS/MMS messages with attacker controlled content)
- Fraud (e.g. by dialling premium rate numbers),
- Espionage (as well as the location retrieving attack an attacked device it could function as a listening device, by ringing a number),
- Malware spreading (by forcing a browser to open a web page with malware located on it)
- Denial of service (e.g by disabling the SIM card)
- Information retrieval (retrieve other information like language, radio type, battery level etc.)

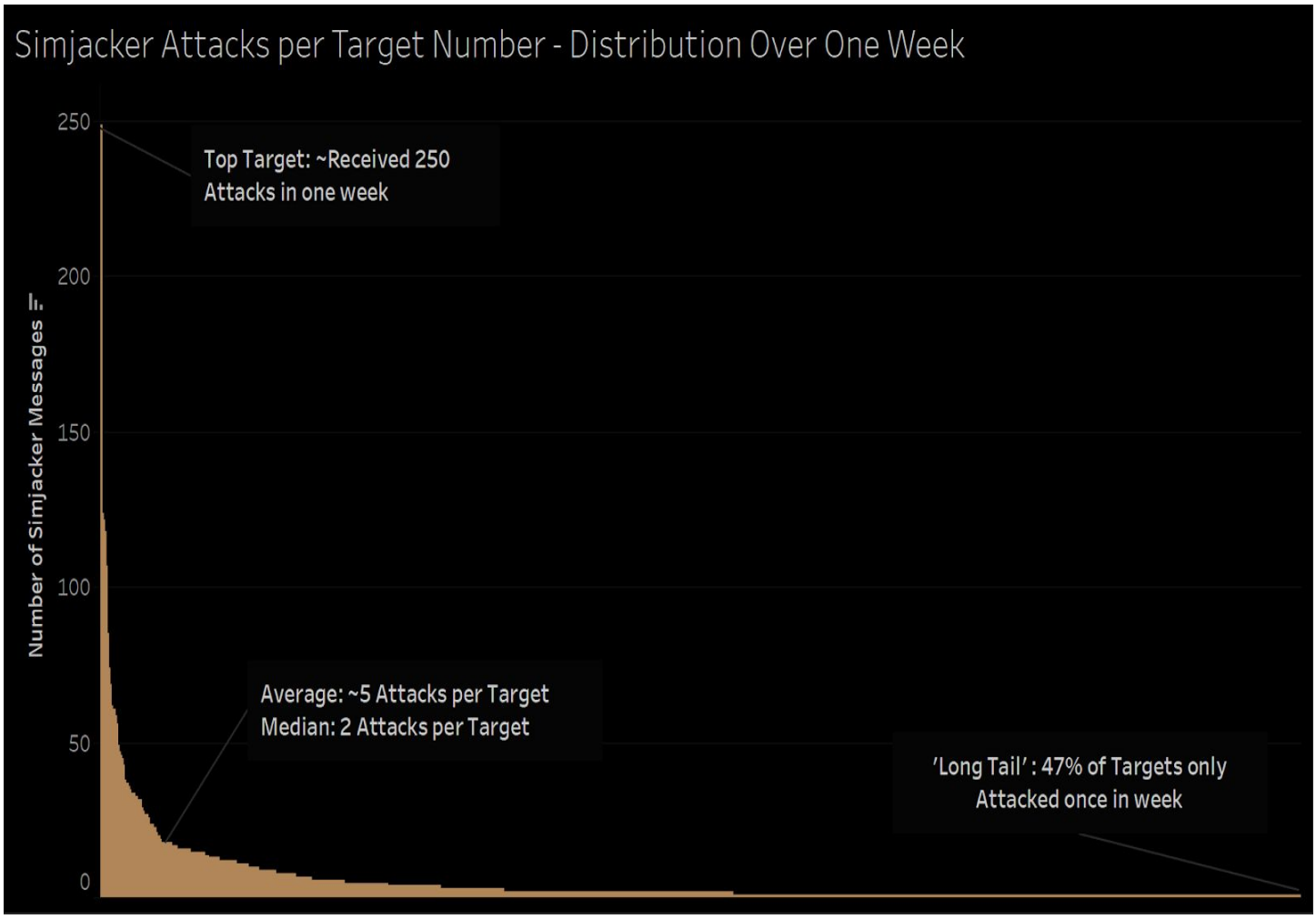
The AdaptiveMobile Security people explained that it might be possible to go even further - depending on handset type - which they will discuss in our Virus Bulletin 2019 presentation on October 3rd. And they note that, worryingly, they are not the only people to think of these additional attacks. They said that over the last few weeks and months that had observed the attackers themselves experimenting with these different capabilities.

Finally, another benefit of Simjacker from the attacker's perspective is that many of its attacks appear to be fully handset make and model independent, thanks to the fact that the vulnerability is due to a very low-in-the-communications-stack SIM standard that all handsets must incorporate. They have observed devices from nearly every manufacturer being successfully targeted to retrieve location: Apple, ZTE, Motorola, Samsung, Google, Huawei, and even IoT devices with SIM cards. It's worth noting that some specific attacks and handsets do matter. Some, such as setting up a call, require user interaction to confirm, but this is not guaranteed and older phones or devices with no keypad or screens (such as IoT device) may not even ask for this.

Who is Doing this

They wrote: "The next question then is who is exploiting this, and why? We are quite confident that this exploit has been developed by a specific private company that works with governments to monitor individuals. As well as producing this spyware, this same company also have extensive access to the SS7 and Diameter core network, as we have seen some of the same Simjacker victims being targeted using attacks over the SS7 network as well, with SS7 attack methods being used as a fall-back method when Simjacker attacks do not succeed. So far, we have seen phone numbers from several countries being targeted by these attacks and we are

very certain that individuals in other countries have also been targeted via Simjacker attacks. Using our collection of Signalling Intelligence we were able to correlate this Simjacker-related SS7 activity with a group we have already detected attempting to attack targets via SS7 means around the world.



In one country we are seeing roughly 100-150 specific individual phone numbers being targeted per day via Simjacker attacks, although we have witnessed bursts of up to 300 phone numbers attempting to be tracked in a day, the distribution of tracking attempts varies. A few phone numbers, presumably high-value, were attempted to be tracked several hundred times over a 7-day period, but most had much smaller volumes. A similar pattern was seen looking at per-day activity, many phone numbers were targeted repeatedly over several days, weeks or months at a time, while others were targeted as a once-off attack. These patterns and the number of tracking indicates it is not a mass surveillance operation, but one designed to track a large number of individuals for a variety of purposes, with targets and priorities shifting over time. The 'first use' of the Simjacker method makes sense from this viewpoint, as doing this kind of large volume tracking using SS7 or Diameter methods can potentially expose these sources to detection, so it makes more sense to preserve those methods for escalations or when difficulties are encountered.

Blocking the Attacks and Thinking Long-term

To deal with this vulnerability, we and the mobile industry have been taking a number of steps.

1. We have been working with our own mobile operator customers to block these attacks, and we are grateful for their assistance in helping detect this activity.
2. We also communicated to the GSM Association – about the existence of this vulnerability. This vulnerability has been managed through the GSMA CVD program, allowing information to be shared throughout the mobile community.
3. As part of this, information was also shared to the SIM alliance, a trade body representing the main SIM Card/UICC manufacturers and they have made new security recommendations for the S@T Browser technology.

In general, our recommendations for the mobile community to deal with the immediate threat is for mobile operators to analyse and block suspicious messages that contain S@T Browser commands. Mobile Operators could also try to change the security settings of UICCs in the field remotely, or even uninstall and stop using the S@T Browser technology completely, but this may be slower and considerably more difficult to do. However, this is very much only a first step, due to the greater implications of the Simjacker attacks.

The existence of Simjacker at all means that we need to radically alter our mindset when it comes to the security of mobile core networks. We believe that the Simjacker attack evolved as a direct replacement for the abilities that were lost to mobile network attackers when operators started to secure their SS7 and Diameter infrastructure. But whereas successful SS7 attacks required specific SS7 knowledge (and access), the Simjacker Attack Message require a much broader range of specific SMS , SIM Card, Handset, Sim Toolkit , S@T Browser and SS7 knowledge to craft. This investment has clearly paid off for the attackers, as they ended up with a method to control any mobile phone in a certain country, all with only a $10 GSM Modem and a target phone number. In short, the advent of Simjacker means that attackers of mobile operators have invested heavily in new attack techniques, and this new investment and skill set means we should expect more of these kinds of complex attacks.

As a consequence, this means that we, in the mobile security community also need to improve our capabilities. For mobile operators, this also means that relying on existing recommendations will not be sufficient to protect themselves, as attackers like these will always evolve to try to evade what is put in place. Instead mobile operators will need to constantly investigate suspicious and malicious activity to discover 'hidden' attacks. We can and should expect other vulnerabilities and attacks that also evade existing defences to be discovered and abused. As the attackers have expanded their abilities beyond simply exploiting unsecured networks, to now cover a very complex mix of protocols, execution environments and technologies to launch attacks with, Operators will also need to increase their own abilities and investment in detecting and blocking these attacks.

The Simjacker exploit represent a huge, nearly Stuxnet-like, leap in complexity from previous SMS or SS7/Diameter attacks, and show us that the range and possibility of attacks on core networks are more complex than we could have imagined in the past. Now is the time to make sure that we stay ahead of these attacks in the future.