VB2021
localhost

7 - 8 October, 2021 / vblocalhost.com

STK, A-OK? MOBILE MESSAGING ATTACKS ON VULNERABLE SIMS

Cathal Mc Daid
AdaptiveMobile Security, Ireland

cmcdaid@adaptivemobile.com

INTRODUCTION

The Simjacker SMS attack, revealed in 2019 [1], showed how surveillance companies are using binary SMSs to gain access to vulnerable SIM card (UICC) applications on mobile devices for surveillance purposes. However, there has been no in-depth follow-up since the research was revealed on what has changed, nor has there been an analysis of other potentially vulnerable UICC applications.

In this paper, we provide a recap of the principals of the Simjacker attack and how it works. First, we will go into detail on what binary SMSs are and their frequency in mobile networks. We will then outline details of other, previously undiscussed, UICC applications that have characteristics that mean they may also be vulnerable to attacks via UICC-destined binary SMSs, as well as their scale and distribution.

In the second part of the paper we will share new details from our experiences in detecting and blocking UICC-destined SMS attacks that exploit the Simjacker vulnerability – including the impact on the industry and on the attacker of releasing public information. We also cover information on a new attack delivery method used by the Simjacker attacker, as well as the scale of their attacks. This will show how these types of attacks are very much on-going, and the importance of intelligence in stopping them.

At the end, we will explain what the mobile operator community has done since the release of the original Simjacker research, and what needs to be done in the future.

BINARY SMS – WHAT IT IS AND HOW IT IS USED

The Simjacker attack involved a specially formatted SMS – termed binary SMS – targeting an unsecured UICC application – called *S@T Browser* – on the SIM card/UICC. This binary SMS contained a list of (U)SIM toolkit (STK) commands. Binary SMSs are not formally defined within the 3GPP specifications, but within the industry we can say that a binary SMS is considered to be a specific type of SMS which is designed to be interpreted by an application running on the receiving mobile device, and not to be read directly by a human – although the results of many binary SMSs (such as missed called notifications) will be displayed to a person. We can identify two types of binary SMS:

1. A binary SMS designed to be interpreted by an application running on the mobile device. These range from simple commands such as missed call notifications, to messages that send provisioning data or manage mobile equipment remotely.
2. A binary SMS designed to be interpreted by an application running on the UICC. These are normally used to communicate with or manage a UICC remotely to perform functions such as downloading applications or managing files.[1]

While this may define for whom the binary SMS messages are destined, identifying them in the first place is not straightforward. We normally use specific encoding parameters of the SMS to help us define whether it is binary message or not. The three main parameters are:

- TP-PID: Protocol Identifier. Defined in 3GPP TS 23.040 [2]
  - For non-binary messages this is typically set to 0x00. For UICC-destined messages these are often set to 0x7F, but can be other values as well.
- TP-DCS: Data Encoding Scheme. Defined in 3GPP TS 23.038 [3]
  - For non-binary messages this is typically set to GSM 7-bit (0x00) and UCS2 (0x08). For binary messages these are normally set to an encoding scheme other than GSM 7-bit. For UICC-destined messages these are often set to 0xF6, but can be other values as well.
- TP-UDH: User Data Header. Defined in 3GPP TS 23.040
  - Data is stored here in the form of information elements with specific identifiers for each element (IEI). For non-binary messages, the main use of this is to indicate concatenated messages (IEI 0x00), and to a lesser extent, to indicate the presence of EMS content. For UICC-destined messages IEIs 0x70 to 0x7F are used.

Care must be taken with these however, as some SMS messages may be destined for an application and have all the above set to 'standard', human-readable values. Plus, some SMS messages may actually be designed to be read by humans but have some of the above parameters set and so may appear 'binary' (in order to avoid anti-spam filters, for example).

From our analysis, the percentage of binary messaging, from three operators' MT SMS traffic over a typical days' traffic, is as shown in Figure 1.

This is traffic on the delivery leg of the SMS flow, and is representative of the general traffic that is received on mobile handsets. As you can see, the percentage of these binary messages is consistent across all three operators, and the majority are simple commands like missed call notifications. This percentage is inclusive of UICC-destined messages. The actual

[1] A binary SMS destined to a UICC is also called an SMS OTA (over-the-air), although confusingly this term is also used sometimes to refer to any binary SMS.

percentage of UICC-destined messages is considerably lower, and varies more. This is because some operators may continuously send large numbers of messages to UICC applications, whereas others essentially don't use it to any great extent except for periodic bursts (for example, to change roaming preferences and other information on the UICC).

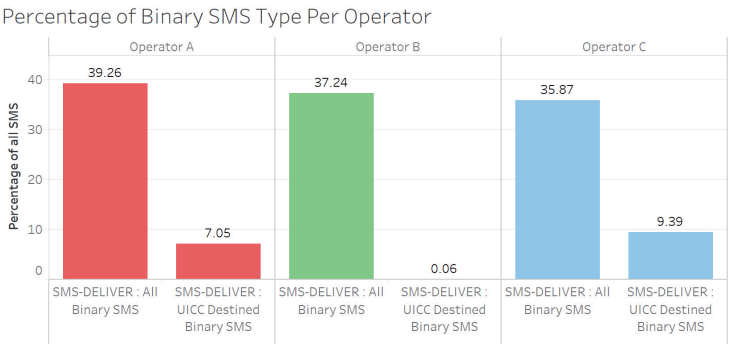

Figure 1: Percentage of binary SMS type per operator.

Past misuses of binary SMSs

Binary SMSs have been used for illegitimate purposes as well. These have ranged from simple nuisance notifications, spam, denial of service effects due to unexpected interaction, all the way to information extraction, location tracking and mobile malware insertion. Table 1 is a list of unique binary SMS-related vulnerabilities reported in the press over the last 20 years.

Name	Date	Target	Confidentiality	Integrity	Availability
WIB Attack [4]	Sep 2019	UICC	X		
Simjacker [5]	Sep 2019	UICC	X		X
OMA CP Phishing [6]	Sep 2019	Device/OS	X	X	
Visual Voicemail -IMAP Attack [7]	Aug 2019	Device/OS		X	
Samsung WAP Push Attack – OMA CP [8]	Jan 2017	Device/OS	X		X
PINGSMS 2.0 [9]	Jul 2015	Device/OS	X		
CoreTelephony Class 0 SMS [10]	Mar 2015	Device/OS			X
iPhone Springboard Class 0 SMS [11]	Jan 2014	Device/OS			X
MONKEYCALENDAR[12]	Dec 2013	UICC	X		
GOPHERSET [12]	Dec 2013	UICC	X		
Rooting SIM Cards [13]	Jul 2013	UICC	X		
Android Nexus Class 0 SMS [14]	Nov 2013	Device/OS			X
SIM Bricking in Android Devices [15]	Nov 2012	Device/OS			X
iOS text spoofing [16]	Aug 2012	Device/OS		X	
Samsung WAP Push Attack [17]	Jul 2012	Device/OS	X		
SIM Toolkit Attack [18]	Nov 2011	UICC			X
SMS of Death [19]	Dec 2010	Device/OS			X
SMS auto-reply[2]	Jun 2010	UICC			X
Fuzzing the Phone in your Phone [20]	Jul 2009	Device/OS			X
TAFT (There's An Attack For That) / Spoofed MMS Notification Message [21]	Jul 2009	Device/OS			X
Windows Mobile 5&6 WAP Push Vulnerability [22]	May 2009	Device/OS	X		
SonyEricsson WAP Push [23]	Jan 2009	Device/OS			X
Curse of Silence [24]	Dec 2008	Device/OS			X
Hijacking Mobile Data Connections [25]	Apr 2008	Device/OS	X	X	
Hide Sender Field - Windows Mobile [26]	Oct 2007	Device/OS		X	
Siemens 45 Long Image Name [27]	May 2003	Device/OS			X
Nokia 6210 Malformed vCard [28]	Feb 2003	Device/OS			X
Nokia 6210/3310/3330 Handset Malformed SMS UDH [29]	Nov 2001	Device/OS			X

Table 1: Unique binary SMS-related vulnerabilities reported in the press over the last 20 years.

[2] Note: Uses SIM Toolkit auto-reply. Not made publicly available at the time, discovered by *Paloma Networks.*

You can see in general that:

1. New binary-message-related attacks have been a consistent feature of mobile security, with on average 1.5 new types of attacks being identified per year.
2. The capabilities of attacks using binary SMS have expanded over time. Initially, binary SMS reported exploits tended to be around simpler availability (denial of service [DoS]) types of attacks but, over time, the capabilities used have become more complex.

Taken together, it is clear that the misuse of binary SMSs has a very long history – almost as long as the history of modern mobile devices, and that new attacks exploiting them are a question of when, not if.

OTHER VULNERABLE SIM CARD/UICC APPLICATIONS

As covered at VB2019 [30], the Simjacker attacks used binary messages that were directed to a specific vulnerable UICC (SIM card) application called *S@T Browser*. A UICC application is a piece of software that runs on the UICC and can receive updates via binary SMS. Binary SMSs destined to UICC applications have a specific value called the TAR (Toolkit Application Reference) encoded in the Command Header binary SMS. This tells the mobile device which UICC application the incoming binary SMS is for.

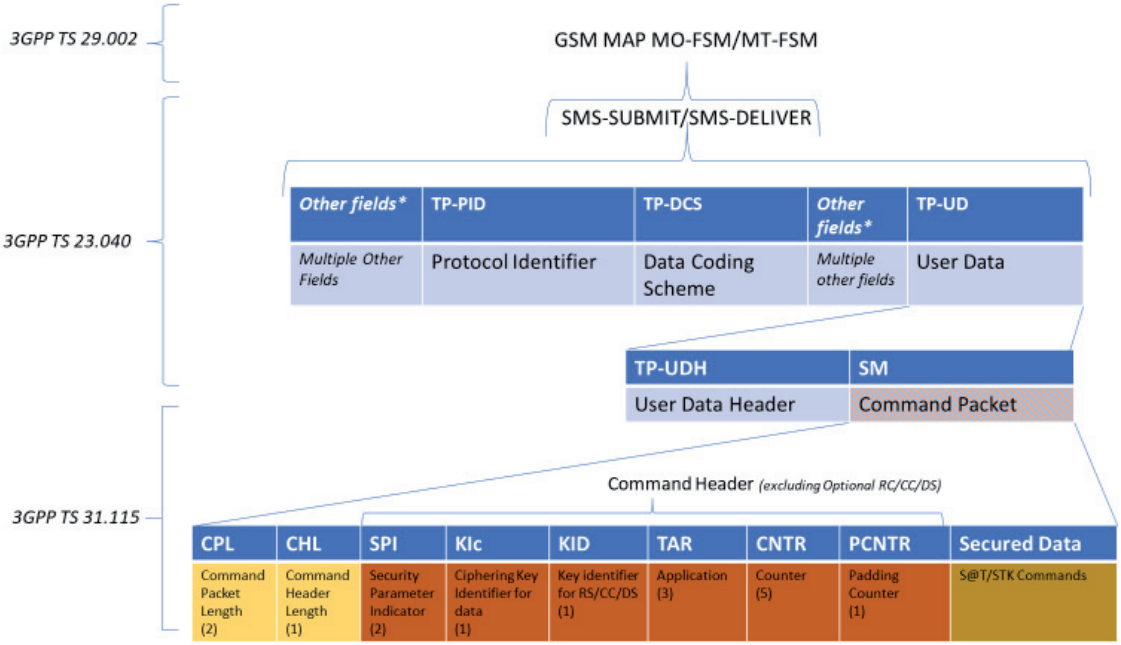

Figure 2: Layers of UICC-destined binary SMS.

The key vulnerability for the *S@T Browser* is that, in the past, this UICC application standard did not restrict who could access it. All messages destined to a UICC application have a specific field in the Command Header – called the Security Parameter Indicator (SPI) – that is used for UICC-destined binary SMSs to indicate what security is in place. If the first five bits of this are set to 0s then this means there is no security in place at this level, and the UICC will accept binary SMSs from any source. This is what led to the Simjacker vulnerability.

In follow-up investigations we also observed other binary SMSs destined to other UICC card applications/TAR values which also had their SPI set to 0s (no security). We were able to get a global view of this by looking at messages to inbound roamers in our customer networks. A relative breakdown of the volumes of these in a customer network in 2020/2021 is shown in Figure 3.

You can see in Figure 3 that, alongside the *S@T Browser*, the main other vulnerable SIM card application we observed is *WIB* (*Wireless Internet Browser*), whose vulnerability was reported shortly after we uncovered the Simjacker vulnerability. However, what is of interest to us here is the ~13% of traffic which goes to *other UICC applications*. In the course of our research in 2020/2021 we identified 30 unique TAR values (UICC applications), active in 50 operators from 39 countries, with zero security set. Table 2 lists those TAR values.

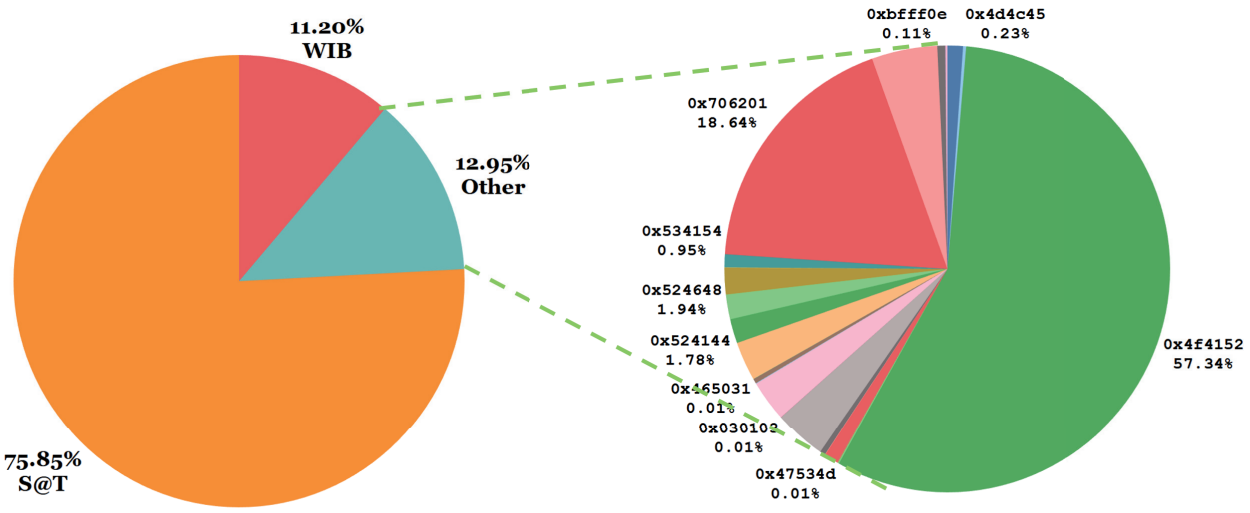

Figure 3: Breakdown of UICC-destined binary SMSs to vulnerable TARs.

UICC TAR with no-security MSL	ASCII value	Number of operators using UICC application	Regions using UICC application, based on subscriber	Prevalence / observed usage
0x012100		2	Europe, Asia	Very low
0x012400		1	Europe	Low
0x030101		1	Europe	Medium
0x030103		1	Europe	Very low
0x030206		1	Europe	Medium, probable encrypted
0x03020b		1	Asia	Very low
0x03020c		1	Europe	Very low, probable encrypted
0x200101		1	Asia	Very low
0x251105		2	Africa, Americas	Low
0x464d4c	FML	1	Americas	Very low
0x465031	FP1	1	Europe	Very low, probable encrypted
0x47534d	GSM	1	Asia	Very low
0x494d45	IME	2	Africa	Low
0x4c5041	LPA	3	Asia, Africa	Low, push engine for events
0x4d4c45	MLE	2	Asia	Low, notifications
0x4f4152	OAR	2	Americas	Common
0x504732	PG2	2	Africa	Medium, contacts exchange and notifications
0x524144	RAD	9	Africa, Asia, Americas	Medium, phonebook backup, balance display
0x524145	RAE	2	Europe, Asia	Medium
0x524648	RFH	1	Europe	Medium
0x533347	S3G	1	Americas	Very low
0x534144	SAD	1	Asia	Very low
0x53414c	SAL	5	Europe	Medium, notifications
0x534154	SAT	1	Europe	Medium, roaming control
0x534c59	SLY	1	Americas	Very low, probable encrypted
0x706201	pb	10	Africa, Europe, Asia	Common in many countries. Contacts exchange & notifications
0xb00001		2	Americas, Europe	Medium, RFM – ADF
0xb00010		4	Americas, Europe, Asia	Low, RFM – SIM file system
0xb00020		1	Americas	Very low, RFM – USIM RFM application
0xbfff0e		2	Europe	Low, propriety toolkit application – SIM events

Table 2: 30 unique TAR values (UICC applications), active in 50 operators from 39 countries, with zero security set.

To give an example of the scale, 'very low' means we observed a handful of IMSIs receiving these messages in a several-month timeframe, whereas 'low' is in the tens of devices, 'medium' is hundreds, and 'common' is higher than this. The volumes themselves are not useful to determine the number of devices vulnerable globally, because they are taken from what we observe in our customer site, but they are useful to give a relative indication between the different TAR values.

A map and table of the global distribution of the vulnerable TARs detected per country/region are below. We have deliberately not indicated what vulnerable TAR is present per location to protect operators.

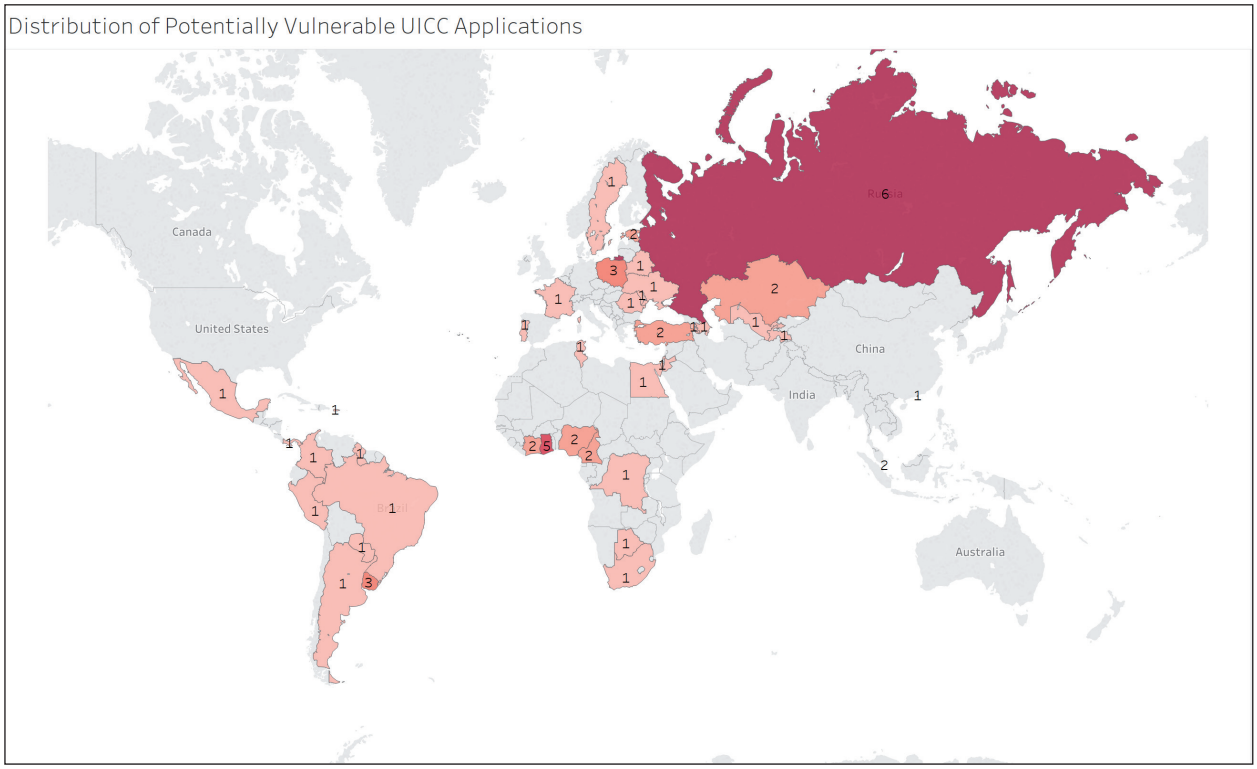

Figure 4: Map of number of potentially vulnerable UICC applications per country/region.

Number of potentially vulnerable UICC applications	Country/region
1	Argentina, Armenia, Azerbaijan, Belarus, Botswana, Brazil, Colombia, Congo Democratic Republic, Egypt, France, Guyana, Hong Kong (China), Israel, Jordan, Mexico, Moldova, Panama, Paraguay, Peru, Portugal, Puerto Rico, Romania, South Africa, Sweden, Tajikistan, Tunisia, Ukraine, Uzbekistan
2	Cameroon, Côte d'Ivoire, Estonia, Kazakhstan, Nigeria, Singapore, Turkey
3	Poland, Uruguay
5	Ghana
6	Russia

Table 3: Number of potentially vulnerable UICC applications per country/region.

In theory, all of these TARs (UICC card applications) are vulnerable to exploitation by attackers. However, it can be difficult to determine:

- The vulnerability of each UICC application – even though the SPI parameter indicates there is no security, the applications may have additional security in place within the UICC card, such as encryption.
- The function of each UICC application – there is no global registry for TAR values, to know which application is which. Therefore, understanding the level of malicious acts possible per application is very difficult.

However, there are ways in which we can make educated guesses about the above.

1. Whether these UICC applications are actually vulnerable

One specific way to understand the level of security in these applications – and thus whether the UICC applications are actually vulnerable – is to determine whether encryption is in place. Even though encryption can be indicated in the SPI

octet, there is nothing stopping *another* form of encryption being in place which is not communicated in the SPI, but is known to the recipient UICC application. We tested for the presence of any encryption – SPI indicated or not – by making observations on the *entropy* of the message. The theory is that binary SMSs with high entropy are more likely to be encrypted than not – thus indicating another layer of security is in place. To test this, we visualized the entropy value per TAR in two ways:

1. The average Shannon entropy (y-axis) per message per TAR
2. The sample entropy per message per TAR (x-axis)

These were calculated per octet in the binary SMS messages.

A high entropy could be due to other factors though, such as compression. To help validate our approach, we also calculated the entropy for a cross-section of UICC-destined binary SMSs to TARs that had a non-zero SPI set, indicating they had encryption in place. Even though we know these TARs are not vulnerable, by including them we would be able to say that any TAR values relatively close to them have the same entropy results as we see for encrypted messages. We also included in the entropy of S@T and WIB – messages types we know have a structure in place – for visual reference. The results are displayed in Figure 4[3].

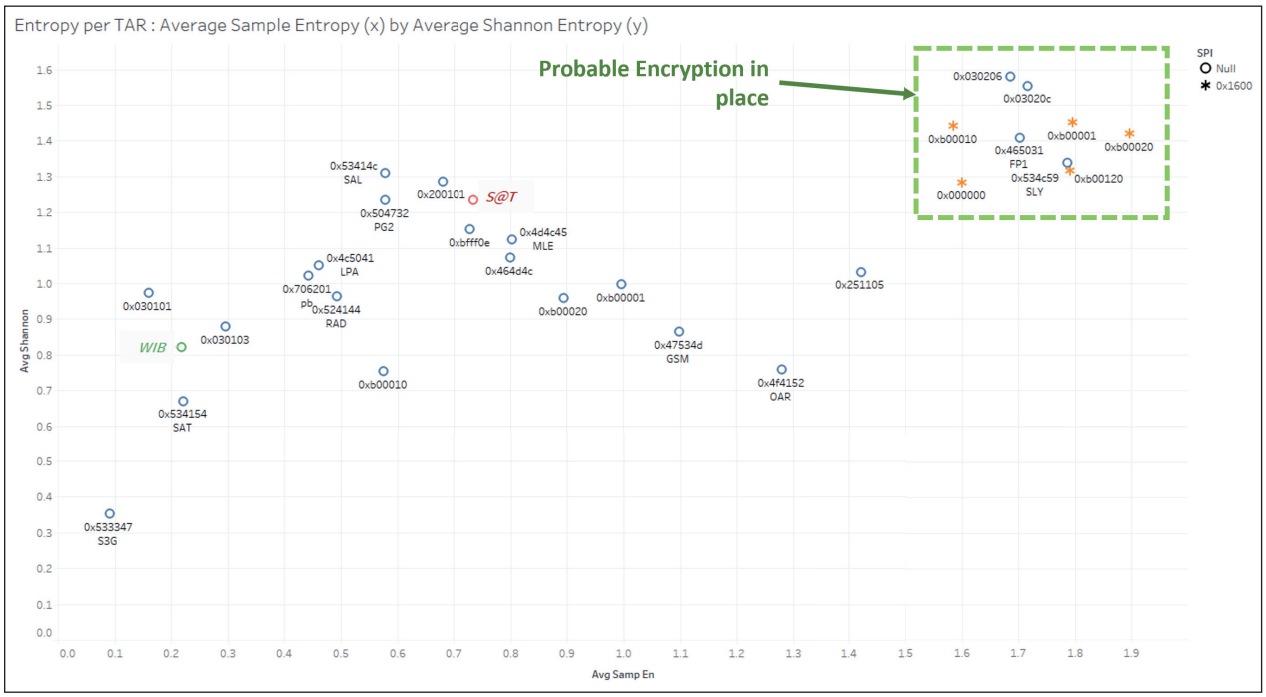

Figure 5: Entropy per UICC application.

These results seemed to confirm our theory, and the use of this method. We believe that, even though the TAR values in the top right corner: 0x3020c, 0x030206, 0x465031 (FP1), 0x534c50 (SLY) have SPI set to 0, it seems they have a form of encryption in place, and thus would be relatively safer than the others. The other TAR values have a less random distribution of octets, indicating there is a structure present which an attacker could reverse engineer. This shows us that the majority of other UICC applications we uncovered with no SPI security set also don't seem to have any other form of encryption in place, and so indeed are potentially vulnerable.

2. What could be achieved if these applications were vulnerable

Given that we know that many of these applications have a recognizable structure, and could be sent binary SMSs, we come to the question of what an attacker could do with them. This is a difficult question to answer. For this research, we did not attempt to execute and prove that malicious activity was possible, as it would have taken many months to reverse engineer up to ~30 different protocols. However, we do not think that a determined attacker – or somebody with insider protocol knowledge – would find this a limiting factor.

Nonetheless, there is a level of partial decoding we can do for some of the applications, by looking for specific octets like ASCII hex values, or be taking in information we found on the web. From this, it seems that many of the applications seem to have been used for notifications. While the impacts of hacking most of these would seem limited, there were several different applications of note for which we can theorize dangerous impacts:

[3] Note: these methods only could be applied for binary SMS messages beyond a certain length.

- 0xb00010 is a TAR that is used to access the SIM file system (Remote File Management). Normally this would always be used with a security level set, but we observed four operators sending messages where this was not set. This application has access to the full DF_{GSM} file set, which has network-related information like IMSI, SIM key (Kc), what operators to connect to [31], and other sensitive information. As an example, in some of these unsecured messages we observed requests to get the contents of 0x6f7e, which is EF_{LOCI} (Location Information) [32].
- 0xb00001 is a TAR that is used for application data file management. Like 0xb00010, this is a sensitive application to get access to as it gives full access to a range of files on the SIM card. We were able to decode some of the unsecured messages and observed they were used to update devices with a list of what networks they were allowed to connect to. In another case we observed that, in one operator in the Americas, it seems to be *S@T Browser* protocol code that was being used. We found that these were refresh commands in order to do a SIM card file update.
- 0x706201 was the most popular TAR application, based on the number of countries we saw it deployed in. This is an application with multiple functions, including interactive menus and notifications for operator offers for credit top-up. It is also used for 'caller exchange'. While the malicious use of it may seem limited, in theory an attacker could use this to suggest contacts to add to a target device.

Ultimately, the full range of attacks possible would only be limited by research, however regardless of the threat, in order to reduce any possible attacks no SIM card application should allow unsecured messages.

Informing the mobile community

As part of this research we informed the GSM Association of these potentially vulnerable UICC applications, who in turn contacted all the mobile operators identified. We grouped the responses we received into different types:

- Some of the applications were already known to be vulnerable by the operator, and they were in the process of migrating and updating the UICC applications to be secure. In the interim, network filtering of binary UICC-destined messaging via SMS firewalls or similar would be required to reduce the overall vulnerability.
- In some other cases the operators were of the belief that, while the UICC applications were vulnerable, the overall system was not vulnerable. Namely, that the UICC applications were designed only to respond to specific source addresses, and/or that SMS firewalls in place would prevent unauthorized attempts to connect to them. We don't consider these to be 100% effective solutions, as the majority of mobile operators do not actively monitor their SMS firewalls for dedicated and sophisticated binary message attacks, and the only way to be certain of security in these cases is for the security of the applications to be improved on the UICC card.
- For a few of these UICC applications, it turned out that the applications were configured correctly, but that the binary UICC-destined messaging was configured incorrectly in certain circumstances, or for certain IMSIs. This misconfiguration was unknown to the mobile operator. The main impact here was not a security risk, but an operational one as these UICC apps were not receiving commands.
- In a few cases, the operator had no information about the UICC application. This mainly occurred for UICC applications in specific countries, and these applications may have been loaded by a third party.

We are unable to tell how many SIM cards are affected globally. The total subscriber numbers of the affected operators come to around 767 million, but we do not believe that operators in these countries have the affected applications on all or many of their SIM cards. On the other hand, there may well be additional operators with these applications present on their UICC cards that we did not observe. Based on prevalence observed, a conservative base estimate is that the total number of affected SIMs will be around 37 million.

As far as we could tell, we did not see any of these applications being exploited. However, this analysis was taken from inbound roamers in one customer network, and is not a view of every message worldwide, so there is no guarantee these UICC applications were not targeted. The only way to be certain that no ill-effects come from these applications is to ensure that all UICC card applications are secured properly, and this is what we recommended to operators.

SIMJACKER (S@T BROWSER) EXPLOITATION

While the 'other' UICC card applications do not seem to be exploited extensively, the same cannot be said for the *S@T Browser* application. To recap, we submitted information on the Simjacker vulnerability to the GSM Association (GSMA) in late June 2019. After internal discussions and conversations with specific known affected operators and SIM card manufacturers, the GSMA issued an internal advisory to all remaining mobile operators in August 2019. *AdaptiveMobile* revealed the existence of the vulnerability publicly on 12 September 2019, and issued technical information on 3 October. The principle behind this staggered release of information to the public was so that mobile operators would have a chance to confirm if they were vulnerable to *S@T Browser* attacks, and put in safeguards if so.

Industry reaction to S@T Browser vulnerability announcement

When we identified the Simjacker vulnerability being exploited by attackers, we were left with a dilemma as to how best to

inform the mobile security industry without revealing information that would allow others to recreate the attacks before operators had a chance to react. Given that there were at least 61 mobile operators affected in at least 29 countries, with potentially up to a billion subscribers affected, we decided to submit the information about the vulnerability through the GSM Association's CVD programme [33], as our company has participated for many years in the various working groups within the GSMA. However, the GSMA suffers from the fact that a large percentage of mobile operators do not actively participate in the working groups, especially smaller, less well-resourced mobile operators. In addition, the Simjacker vulnerability was the first actively exploited vulnerability to be submitted via the GSMA CVD programme.

As a result, even though the technical information had been shared within the GSMA and the mobile community for several months, we were not confident that every affected mobile operator had received the information and taken action on it. Considering these drawbacks, the method we chose to make the Simjacker vulnerability public was to do an initial publicity notification on 12 September, with an icon and a recognizable name in order to increase visibility of the vulnerability, but not to give out any public technical details, prior to our technical presentation four weeks later, and to point concerned operators to the GSMA. We did not comment on social media or respond to many comments in the interim.

Ultimately, this method seems to have been the best one. Despite the information being available in a CVD form, and extensive information sharing with and within the GSMA, after the initial public release we still received multiple private queries from operators who had not known or acted on the information shared up to that point. In many of these cases we also had not known they had the *S@T Browser* technology in place as we had observed no *S@T Browser* on these networks.

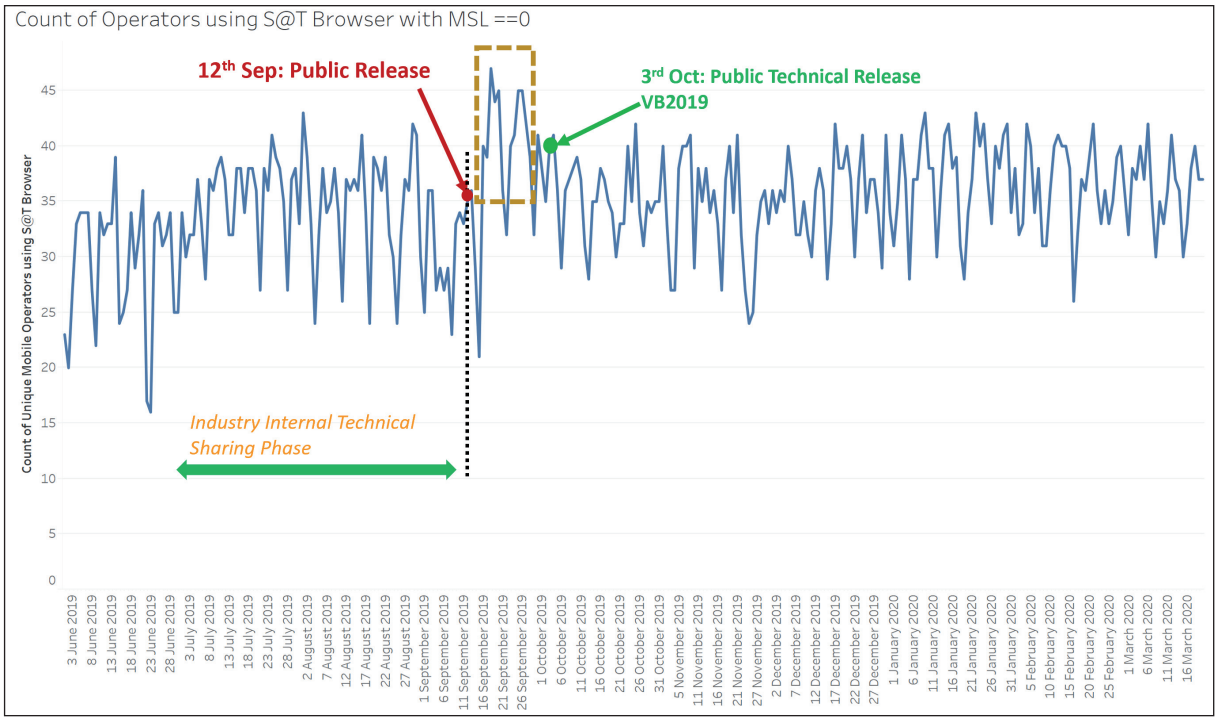

Figure 6: Number of mobile operators using S@T Browser with no security.

We can show the lack of reaction prior to our media release visually. Figure 6 is a graph of the number of operators we observed using the *S@T Browser* technology with no security. You can see that there were two spikes (in the dashed box) a number of days after the public release. This we attribute to some mobile operators testing the presence of *S@T Browser* in their networks, and (we believe) subsequently disabling it. As we did not see this activity before the public release, we believe these operators only heard, or decided to act on the information immediately *after* the public release, and not during the months before when the information was shared within the community. While it may be distasteful to some in the infosec community to use a branded vulnerability as a form of publicity, there is no guarantee – *in this case* – that without it a similar protective effect would have occurred.

This graph is also interesting as we can see that the numbers of operators using the *S@T Browser* has stayed relatively constant after the spikes. We believe that this is because the majority of operators – especially those that make heavily use of the *S@T Browser* – did not disconnect or disable it. Instead, what they did, if they put in protection, was to put in safeguards and filtering in their SMS infrastructure, to filter out malicious or unwanted *S@T Browser*-destined binary messages. This is beneficial, but not 100% effective. It requires constant analysis and investigation, and there are certain ways potentially to bypass it[4].

[4] Note: the above is taken from counts of inbound roamers' mobile operators using the *S@T Browser*, we did not take values after March 2020 due to Covid19. This is because global roaming traffic declined greatly after mid-March 2020, so counts after that date should not be compared against previous values. However, if this had not occurred we believe the trend lines would have stayed the same.

Attacker reaction to S@T Browser vulnerability announcement

If the reaction to the announcement was slow by some mobile operators in the community, the reaction of the attackers was anything but.

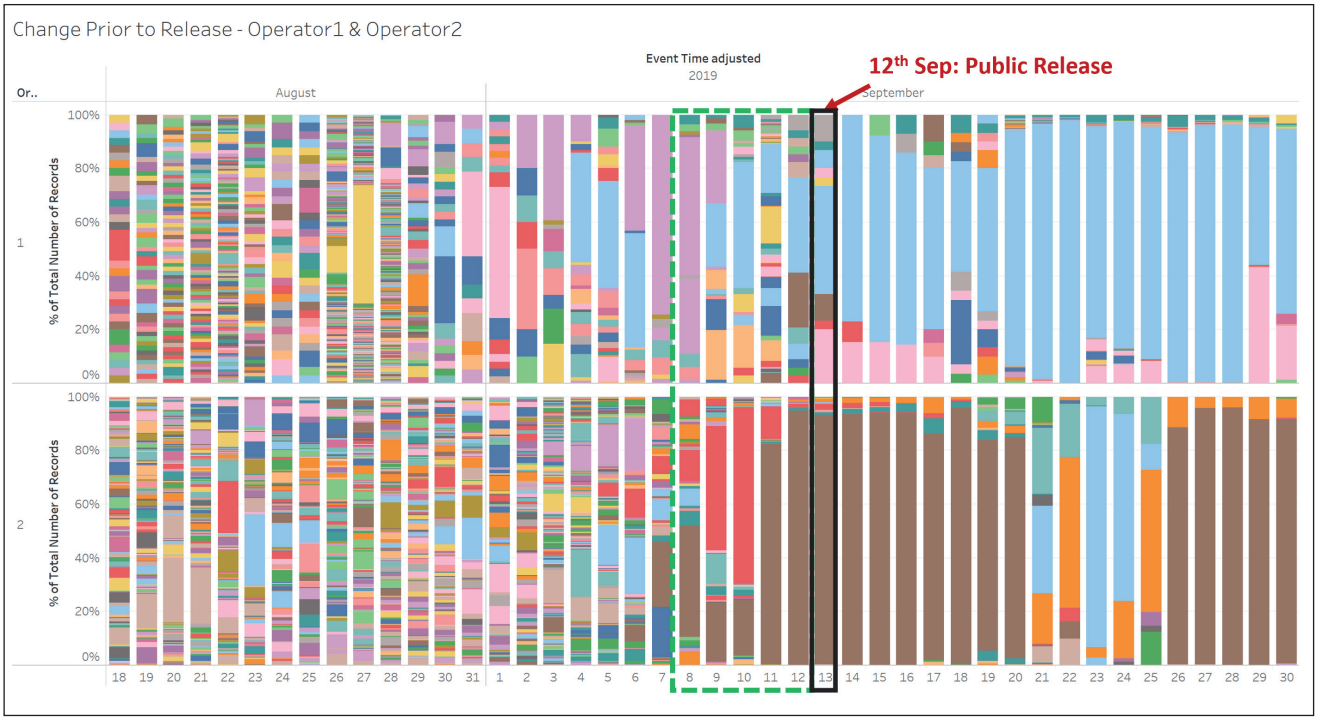

Figure 7: Simjacker attackers' reaction prior to public disclosure.

One interesting thing we observed was that even *before* the public release of information, there was a change in attacker behaviour. The graph in Figure 7 shows the percentage distribution of Simjacker attackers in two operators, where each colour represents an individual target subscriber. We observed that the attackers changed the fundamental nature of their attacks (dashed green box) in the days before the public release (red box), switching from attacks on dozens of subscribers – visually indicated by many small boxes with lots of colours – to sending many times to a very small set of test numbers. As the system did not change (there were no changes in the respective operators in terms of the blocking or detection of these attacks) and it happened at the same time in more than one targeted operator, we attribute these changes to knowledge of the vulnerability in the mobile operator community being made available to the attacker, prior to the general public announcement. This was not unexpected, as inevitably the further that information is disseminated, the greater the chance the attacker has to learn it.

Simjacker over SIP

Other inventive changes we observed indicated that the attackers were trying to bypass the security that was in place. The flow of SMSs within mobile operators can vary depending on network technologies, origin, messaging infrastructure used, and destination. Dozens of possible paths exist, all of which must be investigated and secured in order to stop sophisticated attackers. The danger, of course, is that lesser-known, or unconsidered paths could be used by attackers to slip through attacks.

In our analysis we detected a previously unknown messaging path, where the Simjacker attackers were able to take advantage of the deployment of IMS SIP networks within a customer operator. Specifically, in the case where SMSs were being sent to mobile subscribers who were registered on the VoLTE (IMS) network, SMS delivery was via SIP, and not via the traditional SS7 networks. Attackers were using this path in order to try to bypass defences which were on the SS7 side only. Figure 8 shows a trace of a Simjacker message we detected being sent via a SIP message.

Here, the *S@T Browser* payload is the same, and the message is requesting both IMEI and Cell-Id, with the exfiltrated message being sent out via SMS. But rather than the attack being sent over SMS over SS7, the attack is sent via SIP message.

Once made aware, the mobile operator could plan to put in defences to block any attacks over this path. However, this still meant that other paths could be exploited once this path was defended. To address this, the long-term solution was to help the mobile operator go through all the possible ways for a subscriber to be attacked over any SMS-related messaging interface.

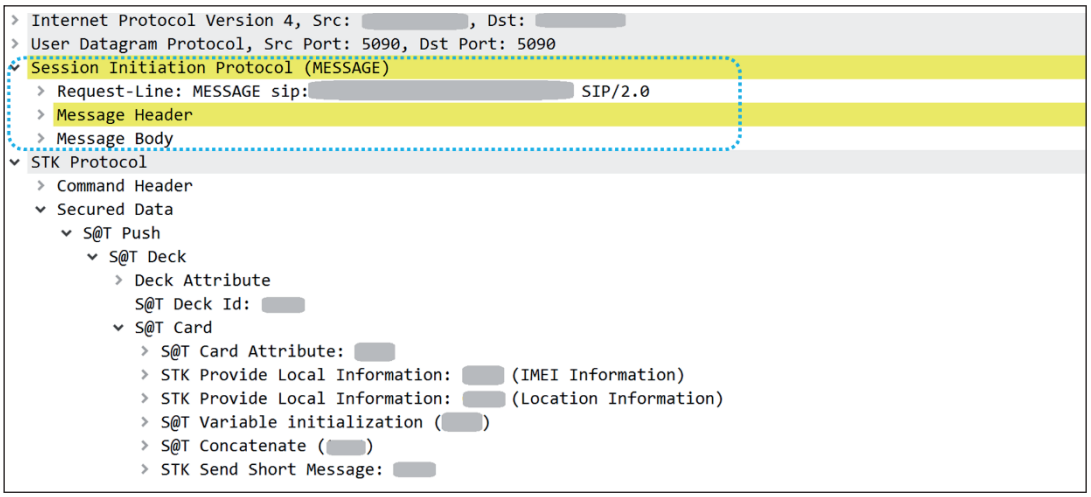

Figure 8: Simjacker message sent over SIP as opposed to SS7.

Analysis of the nature of targets in Mexico

If we do discover the attackers bypassing defences, like the SIP message attack above, this gives us the (brief) opportunity to understand more about the attackers – not only about the attacks themselves, but about what they are trying to extract and the potential scale of their attacks.

From analysing these incidents, we can build up a picture of the devices that are being tracked within Mexico. In 2021, the most tracked device brands were *Apple* phones (30.2%), and the most tracked model is the *iPhone 12 Pro Max* (4.3%). This should not be taken as one device being more vulnerable that others – all devices are vulnerable to the Simjacker attack, as the vulnerability is in the SIM card. Rather, this is simply the distribution of the devices used by the targets.

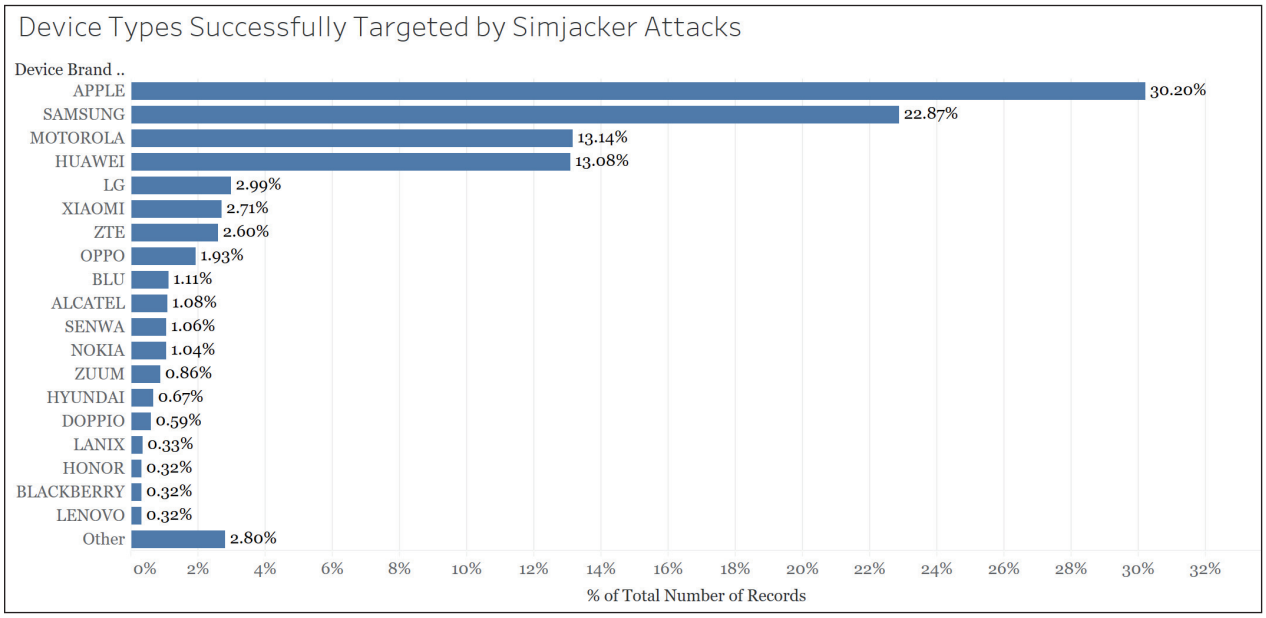

Figure 9: Top 20 device brands successfully targeted by Simjacker attacks.

The vast majority of mobile devices tracked were smartphones (91.7%), but we also noticed a small percentage of feature phones (6.14%). The remaining devices were tablets and modems. Some notable interesting phones we observed were 'secure' devices like the *SilentCircle Blackphone 2* devices successfully being tracked, as well as locator devices used in cars.

In addition, the volumes of these brief incidents allow us to extrapolate the level of attacks using the Simjacker vulnerability that we would expect if we were not present. By taking the volumes of successful attacks we observed for the specific periods and extrapolating to the Mexico subscriber base for a full year, we are also able to estimate the capacity of the attackers. We estimate that the attackers' 'natural' level of surveillance attacks using the Simjacker vulnerability in Mexico is around *259k lookups per year* on around *31k subscribers*. This would be the volume of attacks we would expect in the past (prior to our discovery of the Simjacker vulnerability in 2019), and if all mobile operators in Mexico removed their defences against Simjacker attacks.

INDUSTRY OPERATOR RECOMMENDATIONS AND CONCLUSIONS

Since the Simjacker vulnerability release we have worked closely with the GSM Association in order to cover these attacks and other binary message attacks. Since that time we have helped the GSMA's Fraud and Security Working Group (FASG) to create a version 1 of guidelines for mobile operators to filter binary SMS traffic. This document (*FS.42 Binary SMS Filtering Guidelines*) covers attacks that have happened in the past, and best practices for operators to deal with these attacks. This is a starting point from which to put in place defences to prevent binary SMS attacks in the future. Mobile operators can obtain this through their GSMA membership.

This is only the start. Operators need not only to implement these guidelines, they also need to be prepared to put in the time and resources to monitor these defences, and build up their threat intelligence in this area. Vulnerable UICC applications, and more generally the use of malicious binary SMSs in general, has consistently been shown to be a unique and long-lasting attack vector, yet it is an area that is still rarely discussed or considered. Despite nearly 20 years of different types of binary SMS-related attacks, it took a massive, on-going surveillance campaign using the Simjacker vulnerability for the industry to 'wake up' to the need to properly secure this environment. The ongoing attempts by the attackers using Simjacker to bypass defences, the scale of their attacks, and the recent NSO PegasusProject reports all show the levels of sophistication that mobile phone targeting attackers have reached, and how they use every tool at their disposal, including SMSs. To stop any future messaging attacks using binary SMSs, mobile operators need to make sure their SMS security is more than just OK.

ACKNOWLEDGEMENTS

I wish to thank my colleagues Ryan Dalton and Martin Gallagher, as well as the wider Data Intelligence and Threat Intelligence Unit teams within *AdaptiveMobile Security*, in helping me analyse and interpret this data. I also gratefully acknowledge the assistance of our mobile operator customers in helping us research these threats and thank the GSM Association for helping to distribute recommendations.

REFERENCES

[1] Olson, P. Hackers Use Spyware to Track SIM Cards. The Wall Street Journal. September 2019. https://www.wsj.com/articles/hackers-use-spyware-to-track-sim-cards-11568400758.

[2] 3GPP. 23.040 Technical realization of the Short Message Service (SMS). https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=747.

[3] 3GPP 23.038 Alphabets and language-specific information. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=745.

[4] Lakatos. #WIBattack: Vulnerability in WIB sim-browser can let attackers globally take control of hundreds of millions of the victim mobile phones worldwide to make a phone call, send SMS to any phone numbers, send victim's location, launch WAP browser, etc. Ginno Security Laboratory. September 2019. https://ginnoslab.org/2019/09/21/wibattack-vulnerability-in-wib-sim-browser-can-let-attackers-globally-take-control-of-hundreds-of-millions-of-the-victim-mobile-phones-worldwide-to-make-a-phone-call-send-sms-to-any-phone-numbers/.

[5] Mc Daid, C. Simjacker Technical Paper. AdaptiveMobile.https://www.adaptivemobile.com/downloads/simjacker-technical-paper.

[6] Skrobov, A.; Makkaveev, S. Advanced SMS Phishing Attacks Against Modern Android-based Smartphones. Checkpoint. September 2019. https://research.checkpoint.com/2019/advanced-sms-phishing-attacks-against-modern-android-based-smartphones/.

[7] Silvanovich, N. Look, No Hands! The Remote, Interaction-less Attack Surface of the iPhone. Black Hat. https://i.blackhat.com/USA-19/Wednesday/us-19-Silvanovich-Look-No-Hands-The-Remote-Interactionless-Attack-Surface-Of-The-iPhone.pdf.

[8] Court, T.; Biggs, N. WAP just happened to my Samsung Galaxy. Context /Accenture. January 2017. https://web.archive.org/web/20210226101549/https:/www.contextis.com/en/blog/wap-just-happened-my-samsung-galaxy.

[9] Margaritelli, S. How to use old GSM protocols/encodings to know if a user is Online on the GSM Network AKA PingSMS 2.0. https://www.evilsocket.net/2015/07/27/how-to-use-old-gsm-protocolsencodings-know-if-a-user-is-online-on-the-gsm-network-aka-pingsms-2-0/ (https://web.archive.org/web/20160119181656/http://evilsocket.net/#open).

[10] Apple. About the security content of iOS 8.2: CoreTelephony Class 0 SMS. https://support.apple.com/en-us/HT204423.

[11] Kovacs, E. Flash SMS flaw in iOS can be exploited to make the lock screen unresponsive. Softpedia News. April 2014. https://news.softpedia.com/news/Flash-SMS-Flaw-in-iOS-Can-Be-Exploited-to-Make-the-Lock-Screen-Unresponsive-437566.shtml.

[12] Der Spiegel. The NSA's Spy Catalog. December 2013. https://www.spiegel.de/international/world/a-941262.html.

[13] Nohl, K. Rooting SIM Cards. Black Hat USA 2013. https://media.blackhat.com/us-13/us-13-Nohl-Rooting-SIM-cards-Slides.pdf.

[14] Alecu, B. 0class2DOS. m-sec. http://www.m-sec.net/defcamp2013/0class2DoS.pdf.

[15] Borgaonkar, R. Dirty use of USSD codes in cellular networks. TROOPERS 2012. https://www.troopers.de/wp-content/uploads/2012/12/TROOPERS13-Dirty_use_of_USSD_codes_in_cellular-Ravi_Borgaonkor.pdf.

[16] pod2g's iOS blog. Never trust SMS: iOS text spoofing. August 2012. https://web.archive.org/web/20120825003634/http://www.pod2g.org/2012/08/never-trust-sms-ios-text-spoofing.html.

[17] c0rnholio. [Security Advisory] Samsung leaves it's Android Smartphones with WAP-Push Feature Open to Attacks (one sms to rule them all). Silent Services. July 2012. https://www.silentservices.de/security-advisory-samsung-leaves-its-android-smartphones-with-wap-push-feature-open-to-attacks-one-sms-to-rule-them-all/.

[18] Alecu, B. SIM Toolkit Attack. m-sec. http://blog.m-sec.net/2011/sim-toolkit-attack/.

[19] Mulliner, C.; Golde, N.; Seifert, J.-P. SMS of Death: from analyzing to attacking mobile phones on a large scale. https://www.researchgate.net/publication/228900500_SMS_of_Death_from_analyzing_to_attacking_mobile_phones_on_a_large_scale.

[20] Mulliner, C.; Miller, C. Fuzzing the Phone in your Phone. Black Hat USA 2009. https://www.blackhat.com/presentations/bh-usa-09/MILLER/BHUSA09-Miller-FuzzingPhone-PAPER.pdf.

[21] Lackey, Z.; Miras, L. Attacking SMS.Black Hat USA 2009. https://www.blackhat.com/presentations/bh-usa-09/LACKEY/BHUSA09-Lackey-AttackingSMS-SLIDES.pdf.

[22] Silent Services. [Securitry Advisory] Windows Mobile Security Advisory: Manufacturers leave device open for WAP-Push based attacks. http://www.silentservices.de/adv01-2008.html (https://web.archive.org/web/20100905064934/http://www.silentservices.de/adv01-2008.html).

[23] Mobile Security Lab. MSL-2008-001 SonyEricsson WAP Push Denial of Service. https://www.mseclab.com/?page_id=123.

[24] Engel, T. Vulnerability Advisory. Remote SMS/MMS Denial of Service – "Curse Of Silence" for Nokia S60 phones. https://berlin.ccc.de/~tobias/cursesms.txt.

[25] Mune, C.; Gassirà, R.; Piccirillo, R. Hijacking Mobile Data Connections. Black Hat 2009. https://www.blackhat.com/presentations/bh-europe-09/Gassira_Piccirillo/BlackHat-Europe-2009-Gassira-Piccirillo-Hijacking-Mobile-Data-Connections-whitepaper.pdf.

[26] Whitehouse, O. SMS Handler Issue With Regard to Malformed WAP Push Messages Hiding Source. BugTraq. https://seclists.org/bugtraq/2007/Oct/271.

[27] r2subj3ct dwclan org; Siemens Mobile Phone – Buffer Overflow. https://bugtraq.securityfocus.com/detail/20030506072810.12619.qmail.

[28] Whitehouse, O. Nokia 6210 DoS SMS Issue. @stake, Inc. https://web.archive.org/web/20030301185207/http://www.atstake.com/research/advisories/2003/a022503-1.txt.

[29] de Haas, J. Mobile security: SMS and WAP. Black Hat Europe 2001. http://www.blackhat.com/presentations/bh-europe-01/job-de-haas/bh-europe-01-dehaas.ppt.

[30] Mc Daid, C. Simjacker – the next frontier in mobile espionage. Virus Bulletin. 2019. https://www.virusbulletin.com/conference/vb2019/abstracts/simjacker-next-frontier-mobile-espionage.

[31] 3GPP. TS 51.011 Specification of the Subscriber Identity Module – Mobile Equipment (SIM-ME) interface. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2793.

[32] 3GPP. TS 31.102 Characteristics of the Universal Subscriber Identity Module (USIM) application. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1803.

[33] GSMA. GSMA Coordinated Vulnerability Disclosure (CVD) Programme. https://www.gsma.com/security/gsma-coordinated-vulnerability-disclosure-programme/.