SS7 Exploitation

Andrew Yarbrough, Tanner Bonds
asy0003@auburn.edu, tjb0057@auburn.edu
Auburn University

Abstract

We use phones to communicate with friends, family, relatives, businesses, and many other entities every day. Our calls, or other forms of communication, can travel around the globe and allow us to reach anyone from such a small device. The way we can accomplish this feat is through the Signaling System No. 7 protocol set. Signaling System No. 7 (SS7) is a set of telephony signaling protocols developed in 1975, which is used to set up and tear down telephone calls in most parts of the world-wide public switched telephone network (PSTN) [1]. In this project, we explore the contents of SS7 and how it works in the wild. Then, we simulate our own SS7 network using the tool SigPloit and explore the different kinds of available attacks due to the lack of security inside SS7. Next, we detail how to create the same lab space we use to demonstrate one of our attacks on the SS7 network. Finally, we outline possible countermeasures any individual can do on the client side of SS7 to protect themselves from these vulnerabilities and draw our conclusions of SS7.

1 Introduction

After Alexander Graham Bell invented the telephone in 1876[2], the first telephones network started being developed in 1877. As more and more telephones were being added to this network, the need for a switching mechanism to connect all these phones was needed. Thus, the Public Switching Telephone Network was created. The Public Switching Telephone Network (PSTN) is the operation/system created with individuals called operators to connect callers to their desired contact by using a switchboard where the operator would unplug an aux cord from one port and plug it into another port, thus connecting the two ends of the line. As time went on, the PSTN grew so big that operators were not enough to manage this system alone. Then, in 1975, SS7 was created to replace these operators and automate all these connections. Today, SS7 makes up a vast majority of the network used to connect anyone worldwide for VoIP, SMS, banking, and others. In this paper, we plan to go over several different topics surrounding SS7.

The main goal we wish to accomplish with our project is to simulate, demonstrate, and document how SS7 works, how attacks in SS7 work, demonstrate how those attacks work, and leave behind detailed information on how to replicate the network and attacks in a safe environment. Throughout this project, any testing we do is on a version of SS7 that is separate from the main components of SS7 in the wild. We do not condone or recommend to anyone to try any attacks demonstrated in this project on the real-world SS7 network as the network's infrastructure can be easily damaged, and the user will face hash penalties.

This paper follows our approach to simulating, demonstrating, and documenting the lab we create for testing SS7. We start by presenting related works done on SS7 and attacks done in SS7. Then, we discuss in detail what SS7 is and how it works in the wild. SS7 is made of many nodes and links that bring individuals together over long distances. We will discuss what each of these nodes and links does and how they all work together to bring the set of protocols used to create SS7 and bring people together over the phone. Next, we start explaining the components of our lab and how each component is created and used. We start by listing the virtual environment we use to house our SS7 network. Then, we list the tool SigPloit[3] that simulates and houses our attack demonstrations. From there, we will explain how the different attacks in SigPloit work in the SS7 network and demonstrate one of those attacks. After we have detailed all this information for others to use in the future, we will list possible countermeasures to the attacks found in SS7 and SS7 as a whole for everyday users to employ and understand. Finally, we will wrap up the paper with our conclusion of SS7 and any remarks we have about

SS7 from before, now, and into the future.

2 Related Work

In "Securing SS7 Telecommunications Networks"[4], authors Lorenz, Moore, etc., discuss the SS7 Network's architecture, comparing the backbone of telephone networks being SS7 to TCP/IP being the backbone of the Internet. They further describe the features of PTSN that SS7 controls, such as call set up, tear down, billing, and busy signals, to name a few. SS7 also provides more advanced features such as call forwarding, caller ID, and three-way calling.

However, the SS7 network also contains some significant vulnerabilities. Specifically, due to the nature of the creation of SS7, in that it was meant for a closed telecommunications community, the network possesses limited authentication capabilities. The authors discuss that deregulation and convergence of the PTSN with Internet and wireless networks significantly increase the number of potential vulnerabilities. These factors can allow attackers to perpetrate fraud, interception, and interruption on a massive scale. Attacks on the SS7 network can take place in many forms.

In "SS7 Vulnerabilities—A Survey and Implementation of Machine Learning vs Rule Based Filtering for Detection of SS7 Network Attacks" [5], authors Ullah, Rashid, Afzal, etc., also discuss the issues with the SS7 Network. They provide a focus on SS7 exploits that allow attackers to intercept messages, track the location of a subscriber, tape and redirect calls, send billions of spam messages, and many more. The authors also discuss a potential machine-learning based framework for detecting anomalies within the SS7 Network, which is compared with rule-based filtering; they then provide a conceptual model for such a framework.

3 The SS7 Network

SS7 is a set of protocols used in the everyday building and destruction of telecommunications between individuals or entities. The set of protocols used in SS7 have many components that all work together to provide the service people use every day to make phone calls, text, pay bills, etc. In this section of the paper, we will explore two different avenues regarding SS7. First, we will look at the links and nodes that make up SS7's architecture and detailed information about all the different components. After that, we will explore how SS7 works in the wild and is used from day-to-day communications worldwide.

3.1 SS7 Architecture

The main components that make up the architecture of the SS7 system are links and nodes. In total, there are six different links lettered A to F and three different nodes: Signaling Switching Point (SSP), Signaling Transfer Point (STP), and Signaling Control Point (SCP). Figure 1 below shows how each of these nodes and links forms together to create the SS7 architecture and network. We will refer to this diagram throughout this section when discussing elements of the SS7 architecture.

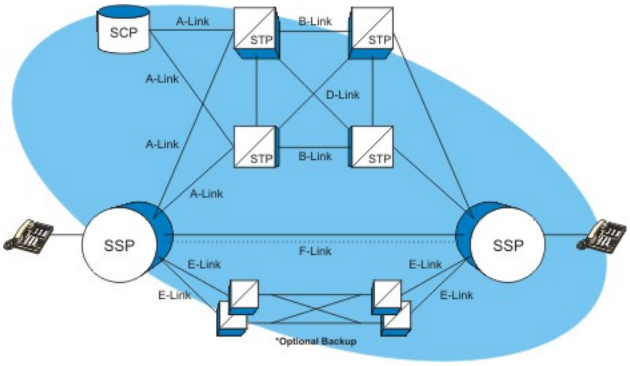

Figure 1: Diagram of SS7 Network Architecture.

The three nodes in SS7 (SSP, STP, and SCP) all serve a different purpose in the building and breaking down of calls between individuals or entities. SSP, or Signaling Switching Point, is the node that handles the originating, terminating, and switching of calls. In the old PSTN system, system operators were in charge of originating, terminating, and switching calls between individuals or entities. They performed these actions by pulling and plugging in aux cords into a switching device that connected and disconnected both ends of the call. With SS7, these actions became automated within the software components of SS7 and made the need for operators obsolete. As shown in Figure 1, the SSP nodes are what phones connect to first. The SSP nodes will gather the information about who is calling and whom the caller is trying to reach and then pass this information off to the STP nodes.

STPs, or Signaling Transfer Points, are the nodes that handle all the receiving and routing of incoming signals towards the proper destinations[6]. Since STP nodes handle all the routing, these nodes also handle any specialized routing functions needed inside the telecommunication. Figure 1 shows that each of the 4 STP nodes is connected via B and D signaling links across from one another and diagonally. The top left and top right STP nodes are also connected to the bottom left and bottom right STP nodes via C-Links. The reason all these nodes are connected in this way is for failsafe purposes. When the left-hand side SSP nodes send the infor-

mation about the call to the left-hand side STP nodes, it sends both nodes the same information. The redundancy here helps so that if one node fails, the other can still rout the right-hand side STP nodes to the desired person to be contacted—the same works on the right-hand side of the STP nodes. Should either STP node on the left-hand side try to contact the STP nodes on the right-hand side, and one of the right-hand side nodes is down, then the call can still proceed since at least one of the nodes is still active.

As we mentioned earlier, the SS7 system can handle conference calls and other advanced types of calling features. The node that handles all this information necessary for advanced calling capabilities is the SCP node. SCP, or Signaling Control Point, provides the advanced calling functions to the STP nodes when the STP nodes recognize that an advanced type of call is being made. Although not shown in Figure 1, SCP nodes are usually created in pairs for the same reason that STP nodes are deployed in pairs: redundancy. Should an SCP node go offline or become unavailable, the STP nodes can access a second SCP node to acquire the information they need for the advanced calling.

The final component of the SS7 architecture we plan to discuss is the different link types. The SS7 architecture has six different link types lettered A to F. A links, or commonly referred to as signaling beginning links, serve as the links that deliver signaling information to and from signaling endpoints. In Figure 1, A-Links are used between the SSP nodes and STP nodes and between the SCP nodes and STP nodes. C, B, and D links serve as links that interconnect the STP nodes across a network. However, the difference between C links and B and D links is that C links are used for mated STP pairs while B and D links connect to other STP nodes in a different mated pair. In Figure 1, the left-hand side STP nodes are connected using a C link. They are connected using a C link because they are a mated pair that serves as redundancies of the other. We can also see in Figure 1 how the left-hand side mated pairs and connected to the right-hand side mated pairs via B and D links where B links connect straight across, and D links connect diagonally. Next, shown at the bottom of Figure 1, we have some optional backup nodes connected via E links. The E links, or extended links, along with the additional STP nodes, are there as a backup if the primary set of STP nodes are to fail. Finally, we have the F link. Once the signal from one SSP node has traveled across the STP nodes to the desired SSP node, a new link will be created between the two SSP nodes to create an F link. This link is the link used to connect the two callers and begin communicating. In later versions of the SS7 network, this F link is no longer created in the fashion we see in Figure 1 due to security reasons.

3.2 SS7 Structure

Now that we have explored the main components and architecture that makeup SS7, we will explore how SS7 works in the wild. Figure 2 below shows the different components used in a real network that communicate and use SS7. Figure 2 is also an example of SS7 being used across different carrier types like Verizon and AT&T across the globe. We can break the components down into two categories and talk about how each component works within those categories. We will start with the Basestation Subsystem.

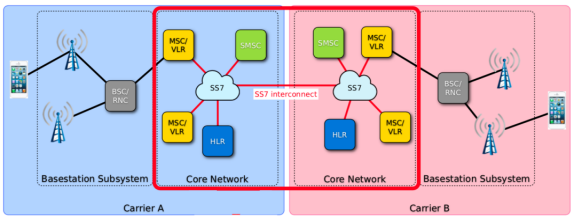

Figure 2: SS7 In The Wild.

The Basestation Subsystem is the phone networking system component that receives signals from a phone and sends them to the main core components to be used for SS7. As shown in Figure 2, our Basestation Subsystem has two cell towers that receive transmissions from a smartphone wirelessly. The signal transmitted from the smartphone is then passed to the BSC/RNC node. The BSC node, or Base Station Controller, is the node that controls and monitors the cell towers as well as serving as the interface element between the cell towers and the MSC node in the Core Network. That covers all the Basestation Subsystem components; so, we will move on to the main category: the Core network.

The Core network is the essential element to the telephony system as it handles all the connecting and addressing between callers. The Core network category has five major components: MSC, VLR, SMSC, HLR, and SS7. In the last section, we already discussed what the SS7 architecture is and how it works; so, we will focus on the other four components here and explain how they all work, and give information to the SS7 component. The first components to start with are the MSC and VLR. The MSC, or Mobile Switching Center, receives the information about the caller and distributes that information to SS7. The MSC component is also used to connect similar subscribers (i.e., two individuals whom both use Verizon) to build the call and communicate. The VLR, or Visitor Location Register, also works with the MSC component. The VLR serves as a database for the MSC component and provides information about the caller within the subscribed network.

The next component of the Core network that

we will discuss is the SMSC component. SMSC, or SMS-Centre, is the main component that handles are SMS texts. This component handles storing, forwarding, converting, and delivering text messages between individuals or entities. How this component works with the SS7 network is that it serves as a storage device. When someone tries to send a text to their friend or family member, the connection between them needs to be created first. So, the Core network will store the contents of the SMS in the SMSC component, and, once the connection is made, the text will be sent through the SS7 connection to the other side (aka the recipient). The messages within the SMSC component are not stored indefinitely. Each message is stored temporarily, with the time being left up to the service provider that set up the network. Once the message is deleted, it can no longer be retrieved to send again. Meaning, should a connection take longer to establish than the allotted time to store the message, the message will get deleted before it can be sent.

The final component we will discuss is the HLR component. HLR, or Home Location Register, is the second database used in the core network. The HLR stores all the information about a subscriber, such as their phone number, services they have available, whether their number has been ported to another network, and more. In the SS7 network, we mentioned that the SSP node would gather the information it needs about the caller to establish the connection between the two ends of the connection. The location that the SSP gets most of this information from is the HLR component. Also, stored in the HLR component is the subscriber's last known location. We explore this type of attack in this lab and show how someone can get this information from HLR illicitly in our methodology.

4 Methodology

In this section, we will outline our planned methodology to conduct a simulated attack on an SS7 Network. We will first discuss the tools we used to conduct this attack, and then our setup and attack execution approach.

4.1 Tools

Our simulated SS7 network and attack were created, ran, and observed inside a virtual machine. A virtual machine running Kali Linux 2020.3 was instantiated and ran using VirtualBox (version 6.1.12). The exact versions of this software are not critical to creating the network, so long as they are relatively up to date to avoid issues with deprecation.

In order to simulate an SS7 network, a tool called SigPloit[3] was used. SigPloit is a signaling security testing framework to exploit and penetration test vulnerabilities used in signaling protocols. In order to test SS7 attacks using SigPloit in the real world, valid permission and access to run these tests on the SS7 network by a telecom provider must be provided. Fortunately, SigPloit provides server-side files that allow us to create a virtual lab with a simulated SS7 Network, allowing us to test these attacks with no risk to real-world systems.

At first glance, it is revealed that SigPloit is currently written in Python2. Python2 is a deprecated and obsolete version of the popular coding language Python. As a result, we must convert all Python2 code in SigPloit to Python3. 2to3[7] is a tool in the Python Standard Library that converts the Python2 code to up-to-date Python3 code. This tool is easily called and utilized on the Python2 files using the command-line interface in Linux.

Upon first boot into the virtual machine, Some packages will be needed for SigPloit to run correctly; these packages are specified in SigPloit's GitHub repository. The two most notable packages are Ipy and pysctp. Ipy[8] is a package that provides a class and other tools for the handling of IPv4 and IPv6 addresses and networks. The package pysctp[9] provides a socket API implementation for the SCTP protocol stack and library. This allows SCTP sockets to be used in scenarios that would typically require TCP or UDP sockets while also preserving the protocol's characteristics.

4.2 Attacks

SigPloit provides simulation servers and attacks for a multitude of protocols and scenarios. These protocols include SS7, GTP (3G), or Diameter (4G). In this instance, we are only concerned with SS7 protocols and attacks. SigPloit contains four attacks that can be launched on our simulated SS7 network, SMS interception, tracking, fraud, and DOS.

With SMS Interception, once an attacker has gained access to the SS7 network, an attacker can target a subscriber on the network while convincing the network that the attacking device is an MLS/VLR node. An attacker will register a victim's MSISDN (mobile phone number) on a fake MSC for the attacker's setup. The real MLS will update the location of the victim's MSISDN. Now, when the victim is sent an SMS, the MSC will transfer the SMS to the SMS-C, in which the SMS-C will then ask the HLR for the victim's location. The HLR will reply with the MSC address that is controlled by the attacker. With this, the SMS-C will transfer the SMS to the attacker's MSC. This attack allows attackers to access sensitive information, such as a two-factor authentication code sent by a bank over SMS to a user needing

verification.

Several methods within the SS7 network allow for attackers to alter information in transit throughout the network, which can result in many instances of fraud. Fraudulent attacks can have many outcomes, such as the redirection of termination of outgoing calls, USSD request manipulation, which can result in the illegitimate transfer of funds, SMS Message Manipulation, which can result in a victim receiving an SMS message from an attacker that appears to be legitimate, and Subscriber Profile Changing, which can allow attackers to bypass being billed for calls for SMS messages sent within the SS7 network[10].

Attackers can conduct Denial-of-Service attacks on users by exploiting vulnerabilities within methods of the protocol's architecture. The service's target can not only be voice and SMS services but internet service as well. The disruption of service is typically conducted against an individual or specific targets. However, if an attacker can gain access to an IMSI (International Mobile Subscriber Identifier) database or even brute-force IMSIs, an attacker will have the ability to conduct a massive service denial across the network. In some cases, attackers can change a subscriber's profile within the database, causing service failure even after a subscriber reboots the specific device. If the VLR address where a subscriber is registered is removed from the HLR, terminating calls cannot be routed to the subscriber's VLR/MSC[10].

One of the most well-known attacks conducted on the SS7 network is Location Tracking. At all times, the network needs to know which base station (cell tower) is closest to the subscriber to receive service. If an attacker can determine the base station's ID, then its geographical coordinates can be found in databases. If an attacker has the cell tower's location, they also have an approximation of a subscriber's location. There is such a high density of cell towers in a large city that an attacker could approximate a subscriber's location to their current street is possible. This attack is the primary attack we selected to conduct on our simulated network.

One of the most well-known attacks conducted on the SS7 network is Location Tracking. At all times, the network needs to know which base station (cell tower) is closest to the subscriber to receive service. If an attacker can determine the base station's ID, its geographical coordinates can be found in databases. If an attacker has the cell tower's location, they can also approximate a subscriber's location. There is such a high density of cell towers in a large city that an attacker could approximate a subscriber's location to their current street is possible. An attacker will query the MSC/VLR directly and ask the HLR for the specific subscriber's IMSI and global title, in which a global title is an address in the SCCP protocol used to route signaling messages[11]. If an attacker can obtain the IMSI and global title of a subscriber, then the MSC/VLR can be asked to query the Cell ID of the subscriber.

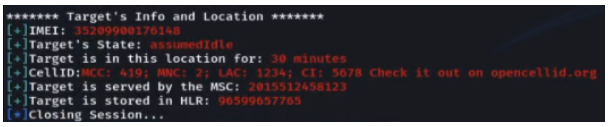

Figure 3: Results of Simulated Location Track.

Figure 3 shows the results of a successful location track attack on our simulated network using SigPloit. The attack will print out the victim's IMEI, the serial number of the cellular device on the network, whereas the IMSI is identified by the SIM card stored in the device. SigPloit will also print out the victim's MSC they were served by and the HLR in which they are stored. However, this information is already known to the attacker. SigPloit will give us a readout of our victim's state and how long they have been within range and connected to this specific base station. SigPloit finally gives us the ID number of the base station, which can be found using a simple lookup. However, as this is a simulated cell tower, no results will be found from a lookup of the given Cell ID.

5 Countermeasures

When a typical consumer sees all these different types of attacks that can be done on a network that everyone around the world uses daily, it begs the question: What can we do to protect ourselves from these attacks? What can we do to fix this? In regards to fixing SS7, there is no "real" solution. The reason for there being no real solution is because everyone uses SS7. To fix the protocols in SS7 that causes these issues, one would need to change the source coding and structuring of SS7 and then distribute it globally. As many research pieces have shown, a vast majority would not start using the new update until years later. So, instead of fixing SS7 at the protocol leave, we will suggest some practical and impractical countermeasures that people can use or implement independently.

For impractical countermeasures, the easiest one to do is not buy a phone. If a person does not own a phone, then none of the detailed attacks in this paper will affect that individual. However, the trade-off for this countermeasure is quite steep in that a person loses the ability to stay in touch with friends and family across the globe. Another impractical countermeasure would be a One-to-One telephone. These telephones would work similarly to the cup and string phones that kids make

to talk to each other over a short distance. The phones would only be linked to each other and only communicate with each other without using any network like SS7. The drawback of this approach is distance. How far individuals would be able to communicate would depend on the length of the wire(s) between the two phones.

Moving on to the more practical countermeasures, both credited security analysts and governments suggest these. One such countermeasure is User Password Security. User Password Security implements two-factor authentication inside SMS delivering apps. For example, the application Signal would work. The application encrypts any messages being handled between two users based on secret keys that the app builds for the two parties. Once each member has traded keys and built their secure channel, the app will verify the messages and senders' integrity and allow for verified and approved participants to read the message(s). Another countermeasure that moves more towards the subscribing companies like Verizon, AT&T, t-mobile, etc., is to have excellent monitoring and event analysis techniques. These companies can implement techniques that actively monitor their SS7 network and determine if and when a breach has happened, and act accordingly. These techniques should be implemented on both the server-side and client-side to protect both corporate components of the network and individual users. The final suggestion we have for users is Regular updates. SS7 has been around for a long time, and new techniques within subscribing companies are being implemented every day to find better ways to protect from these attacks. Unfortunately, these new techniques are only useful if subscribers download the updates on their devices and use them. Subscribers need to download any security updates they receive to improve the security of their devices and protect themselves from these numerous SS7 attacks.

6 Conclusion

In this article, we have described a network that has been in service since the telecommunications industry became automated. This network has provided a plethora of features to improve the average consumer's telecommunications experience. However, this network's use may be outweighed by its risks. The vulnerabilities within this network create a considerable risk to a user's security and privacy, and an attacker can exploit these vulnerabilities with ease. We have shown that these attacks can be conducted with scripts that can violate a subscriber's privacy and track their location within minutes. We discuss a series of countermeasures, both practical and impractical, to help improve the security of the network.

References

[1] "Signalling System No. 7," 15-Aug-2020. [Online]. Available: https://en.wikipedia.org/wiki/Signalling_System_No._7. [Accessed: 03-Dec-2020].

[2] "Telephone History", Dalbello.comminfo.rutgers.edu, 2000. [Online]. Available: http://dalbello.comminfo.rutgers.edu/FLVA/infrastructure/infoinfra/telephone/index.html. [Accessed: 03- Dec- 2020].

[3] R. D'Alessandro and I. Dal Grande, "SigPloit," GitHub. [Online]. Available: https://github.com/SigPloiter/SigPloit. [Accessed: 02-Dec-2020].

[4] G. Lorenz, T. Moore, G. Manes, J. Hale and S. Shenoi, "Securing SS7 Telecommunications Networks", 2001. [Online]. Available: https://www.researchgate.net/profile/Sujeet_Shenoi/publication/216757769_Securing_SS7_Telecommunications_Networks/links/0046352f8d024af680000000/Securing-SS7-Telecommunications-Networks.pdf. [Accessed: 03-Dec-2020].

[5] K. Ullah, I. Rashid, H. Afzal, M. M. W. Iqbal, Y. A. Bangash and H. Abbas, "SS7 Vulnerabilities—A Survey and Implementation of Machine Learning vs Rule Based Filtering for Detection of SS7 Network Attacks," in IEEE Communications Surveys Tutorials, vol. 22, no. 2, pp. 1337-1371, Secondquarter 2020, doi: 10.1109/COMST.2020.2971757. [Accessed: 03-Dec-2020]

[6] "Signaling System 7 (SS7)", Cs.rutgers.edu, 2020. [Online]. Available: https://www.cs.rutgers.edu/ rmartin/teaching/fall04/cs552/readings/ss7.pdf. [Accessed: 05- Dec- 2020]

[7] "2to3 - Automated Python 2 to 3 code translation — Python 3.9.1rc1 documentation", Docs.python.org. [Online]. Available: https://docs.python.org/3/library/2to3.html. [Accessed: 05- Dec- 2020]

[8] "IPy", PyPI. [Online]. Available: https://pypi.org/project/IPy/. [Accessed: 05- Dec- 2020]

[9] "pysctp", PyPI. [Online]. Available: https://pypi.org/project/pysctp/. [Accessed: 05- Dec- 2020]

[10] "SS7 VULNERABILITIES AND ATTACK EXPOSURE REPORT", Gsma.com, 2018. [Online]. Available: https://www.gsma.com/membership/wp-content/uploads/2018/07/SS7_Vulnerability_2017_A4.ENG_.0003.03.pdf. [Accessed: 05- Dec- 2020]

[11] "Global title", En.wikipedia.org. [Online]. Available: https://en.wikipedia.org/wiki/Global_title. [Accessed: 05- Dec- 2020]