Implementation and Analysis of SS7 Signalling Firewall Research Laboratory

Ahmet Mithat Demirkol
Department of Cybersecurity, ASELSAN, Ankara, Türkiye

amithatdemirkol@gmail.com

Abstract: This report details the creation of a high-fidelity, isolated laboratory environment designed to practically analyze Signalling System 7 (SS7) protocol vulnerabilities and the defensive capabilities of modern signalling firewalls. The primary objective of the project is to transform telecommunications infrastructure security testing into an accessible and repeatable methodology using entirely open-source tools. The laboratory architecture is based on a classic Man-in-the-Middle (MITM) scenario involving an attacker (SigPloit), a target (Osmocom STP), and a defense mechanism (P1sec SigFW). This configuration allows all signalling traffic between the attacker and the target to pass through the interposed firewall, enabling a systematic evaluation of the firewall's effectiveness. The report provides a step-by-step guide for setting up the laboratory, configuring the components, executing the test scenarios, and analyzing the results obtained. The empirical findings indicate that while open-source firewalls offer viable mitigation capabilities against standard rule-based threats, they exhibit critical structural limitations when exposed to stateful tracking anomalies and advanced evasion techniques targeting protocol parsers. This work serves as both a theoretical foundation and a practical guide for professionals and researchers working in the field of telecommunications security.

Keywords: SS7 security, Signalling firewall, Telecom security, SigPloit, Osmocom, MITM attack

1. Introduction: Architecture of the SS7 Security Research Laboratory

1.1 Project Vision and Objective

The strategic objective of this report is to create a high-fidelity, isolated laboratory environment for the practical analysis of SS7 protocol vulnerabilities and the defensive capabilities of modern signalling firewalls. This work aims to transform complex and often inaccessible telecommunications infrastructure security testing into an accessible and repeatable methodology using entirely open-source tools.

1.2 Laboratory Components

The laboratory is built on three core open-source tools, each serving a specific role within the ecosystem:

- Sigploit (Attack Tool): An offensive toolkit that simulates an external attacker with access to the SS7 network. It is used to carry out common SS7-based attacks such as location tracking and SMS interception.
- P1sec SigFW (Defense Mechanism): A signalling firewall positioned to monitor, filter, and log signalling traffic. Java-based, SigFW offers filtering capabilities for SS7 and Diameter protocols. Designed as a research project, this tool is ideal for the laboratory environment. (P1 Security, 2024).
- Osmocom STP (Target Network): The Signalling Transfer Point (STP), part of the Osmocom project, represents a legitimate network element (e.g., HLR or MSC) targeted by simulated attacks.

1.3 Strategic Goal: Man-in-the-Middle (MITM) Configuration

The project's primary technical goal is to configure these components to create a classic Man-in-the-Middle (MITM) scenario. This architecture ensures that all signalling traffic between the attacker (SigPloit) and the target (Osmocom STP) passes through the firewall (SigFW) placed in between. This setup allows us to systematically evaluate the accuracy of the firewall's filtering rules, their impact on performance, and their resilience against advanced evasion techniques. (White, 2025; Secret Double Octopus, n.d.)

2. Fundamentals of SS7 and SIGTRAN Security

2.1 SS7 Architecture and Security Model

SS7 was designed in the 1970s for a world where telecommunications networks were state-owned and mutual trust between operators was assumed. This “walled garden” paradigm is the main reason why fundamental security mechanisms such as authentication or encryption are absent from the protocol's design. The SS7 protocol stack consists of layers such as MTP3, which is responsible for addressing and routing, SCCP, which provides end-to-end routing, and TCAP, which carries application layer messages.

2.2 SIGTRAN and the Transition to IP

As the telecommunications world evolved towards IP-based infrastructures, the SIGTRAN (Signalling Transport) protocol suite was developed to carry SS7 traffic over IP networks. SIGTRAN uses SCTP as its transport layer and enables the upper layers of SS7 to be carried over IP through adaptation layers such as M3UA. However, this modernization failed to transfer SS7's underlying “trust model” to the new environment. Combined with commercial applications such as Global Title (GT) leasing, it opened access to the SS7 network to numerous different organizations, directly creating today's threat environment.

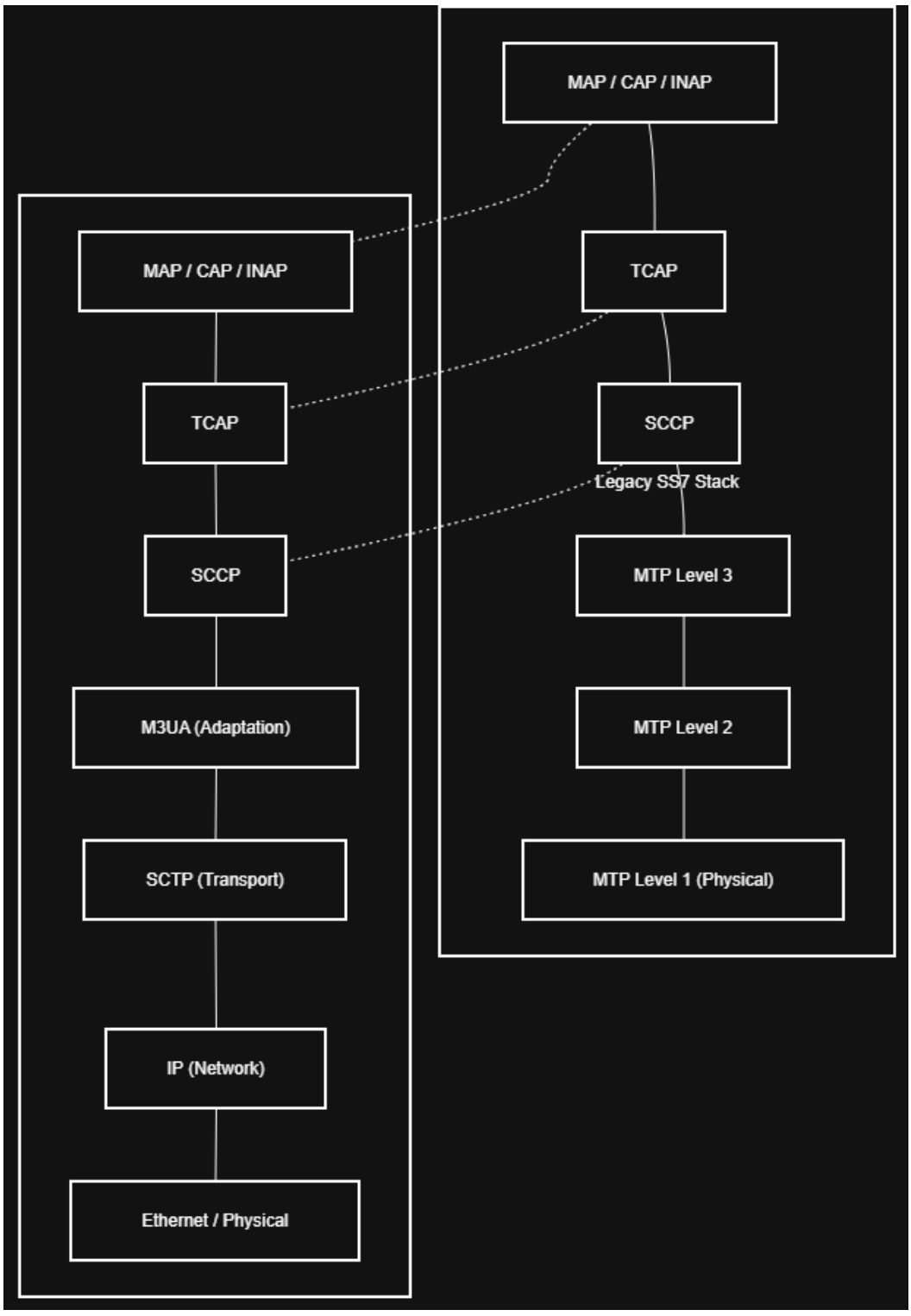

Figure 1: Protocol layer mapping between the IP-based SIGTRAN suite and the legacy SS7 signalling stack, demonstrating how higher-level transactional elements are encapsulated over IP transport

3. Setting Up Laboratory Tools

3.1 The Attacker's Arsenal: SigPloit

Purpose: SigPloit is a Python-based framework designed to simulate common SS7 attacks such as location tracking and SMS interception.

Installation: The repository is cloned and Python 2 dependencies are installed:

```
git clone https://github.com/SigPloiter/SigPloit
cd SigPloit
sudo pip2 install -r requirements.txt
```

Code 1: SigPolit Installation

Initial Configuration: The local IP addresses required for the vehicle to operate are added to the loopback interface.

3.2 Network Core Simulator: Osmocom

Purpose: Osmocom's OsmoSTP component will act as the Signalling Transfer Point to simulate the “victim” network.

Installation: Osmocom packages are compiled from source to allow for flexibility in making changes. Necessary prerequisites such as *“libosmo-sigtran”* are installed natively.

3.3 Monitor: P1sec SigFW

Purpose: SigFW is an open-source SS7 and Diameter firewall written in Java. This tool is positioned as a “reference application and research project.”

Compilation and Execution: Maven commands are used to compile and execute the different components of SigFW.

Security Hardening: Steps such as changing the default API credentials and generating new cryptographic keys must be taken before deployment.

4. Laboratory Construction and Network Configuration

This section details the practical steps required to assemble the isolated virtualized infrastructure, map the addressing schemes, and bind the transport links to achieve transparent inline enforcement.

4.1 Virtualization and Network Architecture

The laboratory environment is built on three virtual machines (VMs) to provide an isolated and repeatable structure. Each VM has a dedicated network interface on a shared internal network (192.168.56.0/24).

Table 1: Laboratory Environment IP Addressing and Port Assignment Plan

VM Role	Machine Name	Static IP Adress	SCTP Listening Port	Purpose
SigPloit Attacker	sigploit-attacker	192.168.56.10	-	The SS7 client that initiates attack traffic.
SigFW Firewall	sigfw-firewall	192.168.56.20	8016 (Inbound)	Listens for connections from SigPloit (Input).
SigFW Firewall	sigfw-firewall	192.168.56.20	- (Outbound)	Initiates the connection to the Osmocom STP (Output).
Osmocom STP	osmocom-stp	192.168.56.30	8015	The target server that accepts connections from SigFW.

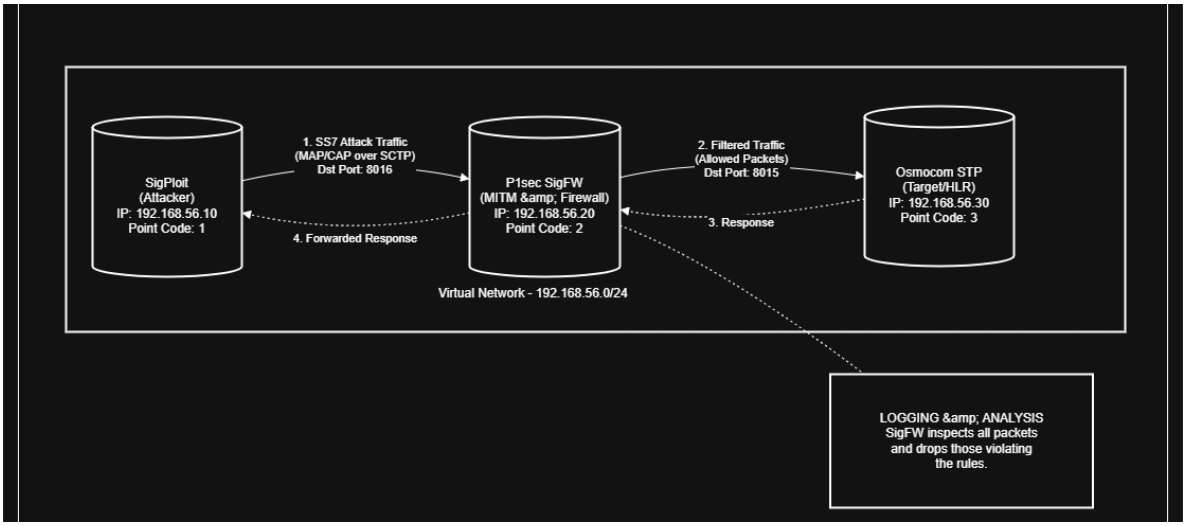

Figure 2: Architectural diagram of the high-fidelity Man-in-the-Middle (MITM) lab topology, detailing packet capture points and host routing configurations across the isolated virtual switch

4.2 Configuring SigFW as the Middle Man

The central file controlling SigFW's behavior is *ss7fw.json*. Correct configuration of this file is critical to the success of the MITM scenario.

- **sctp_server:** Defines the IP address (192.168.56.20) and port (8016) where SigFW will listen for incoming SCTP connections.
- **sctp_server_association:** Specifies the expected source IP (192.168.56.10) of the client (SigPloit) connecting to the firewall.
- **sctp_association:** Defines the outgoing connection from the firewall to the final destination server (Osmocom STP) (192.168.56.30:8015).

Routing in the SS7 network is performed using Point Codes. A simple addressing scheme is defined for the lab: SigPloit (PC=1), SigFW (PC=2), and Osmocom STP (PC=3). These codes must be configured consistently across all devices.

4.3 Establishing and Verifying Connections

Services must be started in the correct order:

1. Osmocom STP,
2. SigFW,
3. SigPloit. To verify whether the connection is successful, the *sigfw.log* file, the management API (**https://192.168.56.20:8443/ss7fw_api/1.0/get_status**), and live packet analysis via Wireshark are monitored.

5. Offensive Operations and Firewall Verification

This section establishes the technical baseline for evaluating the firewall's detection posture by implementing standardized attack playbooks and advanced structural evasion methods.

5.1 Establishing and Verifying Connections

The GSMA FS.11 document, the industry standard guide for SS7 firewalls, categorizes threats into three main categories:

- **Category 1 (Prohibited Packets):** Messages that should never cross operator boundaries and are strictly internal.
- **Category 2 (Unauthorized Packets):** Messages that are contextually incorrect, such as cross-boundary requests for home network tracking data originating from unvouched sources.
- **Category 3 (Suspicious Location Packets):** Situations where legitimate message types are abused to execute tracking anomalies, such as geographically impossible velocities between consecutive location update requests.

Table 2: GSMA FS.11 Threat Categories and Corresponding MAP/CAMEL Messages

GSMA Category	Threat Description	Sample MAP/CAMEL Operation	Typical Filtering Logic
Category 1	Leakage of Internal Network Messages	ActivateTraceMode, PurgeMS	Block in all cases where the source network is not the local network.
Category 2	Location Spoofing/Unauthorized Information Request	UpdateLocation, ProvideSubscriberInfo (PSI)	Block if there is an inconsistency between the source GT address and the country to which the IMSI belongs.
Category 3	Impossible Speed Location Change	UpdateLocation	Block if the time and geographical distance between consecutive UpdateLocation messages indicate a physically impossible speed (Velocity Check).

The mathematical model underpinning the stateful Velocity Check tracking control is defined by a temporal and co-ordinate differential ratio: L1

$$V = \frac{\Delta d}{\Delta t}$$

Figure 3: Velocity check formula

Where Δd represents the spatial distance between the baseline location L_1 and the newly requested location $L2$ calculated over the earth's geodetic surface ($\Delta d = Distance(L1, L2)$), and Δt represents the temporal delta between the two inbound transactional timestamps $\Delta t = T2 - T1$). If $V > Vmax$, the packet is classified as an impossible travel anomaly and dropped.

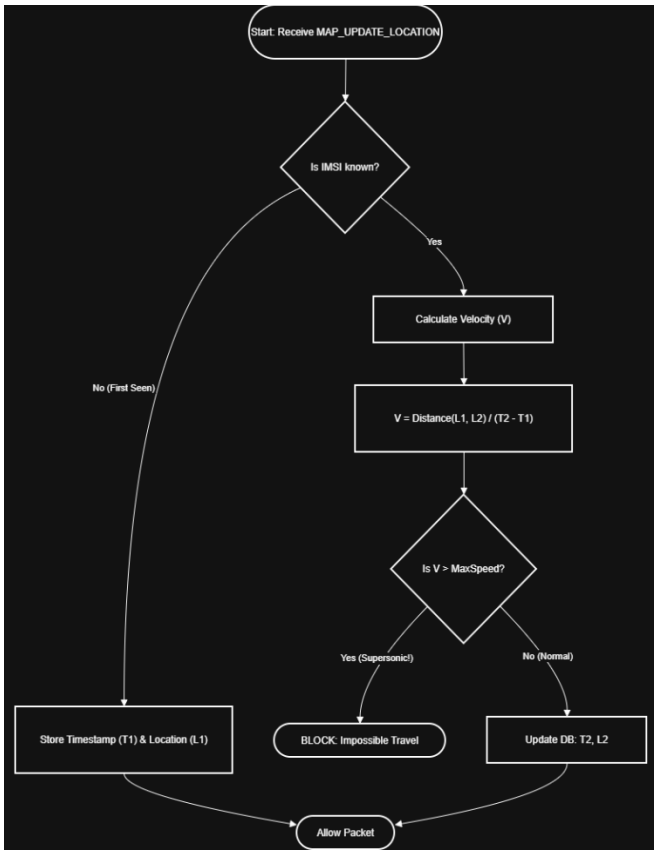

Figure 4: Logical flowchart illustrating the stateful Velocity Check verification algorithm applied to inbound Category 3 location update management

5.2 Execution of Standard Attacks and Advanced Evasion Techniques

To rigorously evaluate the framework, two distinct testing methodologies were executed: a firewall-enforced verification flight and a baseline validation flight where the firewall's filtering configurations were entirely bypassed to confirm that the attack tool (*SigPloit*) could achieve 100% exploit delivery directly to *Osmocom STP*.

During the firewall-active trials, a location tracking scenario was conducted by injecting a Category 2 ProvideSubscriberInfo (PSI) request (Operation Code 70) via SigPloit. The firewall rules were configured to validate incoming Calling Party Addresses (CgPA) at the SCCP layer against the targeted International Mobile Subscriber Identity (IMSI) blocks. Latency data was gathered by extracting the frame timestamp differentials between the firewall's inbound network interface and its outbound transit interface using automated PCAP post-analysis.

Following standard validation, parser difference attacks targeting structural decoding layers were introduced. This technique exploits intentional ambiguities in the Abstract Syntax Notation One - Basic Encoding Rules (ASN.1 BER) utilized by TCAP to establish transactions. A malformed packet was generated by writing a custom Python script to override standard length framing, explicitly utilizing an Indefinite Length descriptor for the IMSI parameter components inside the TCAP component portion.

The evaluation criteria defined a "Blocked" outcome as the firewall successfully generating an explicit rule drop log entry or issuing an SCTP Abort packet, whereas an "Allowed" outcome was marked if the packet was successfully replicated and pushed to the destination port of the target core server.

6. Firewall Activity Analysis and Reporting

This section compiles the empirical data gathered during testing, contrasting the behavioral results of rule-based enforcement against advanced protocol evasion.

6.1 Methodology for Security Assessment

A structured framework is used to present the findings of laboratory experiments. The key assessment criteria are as follows:

- Security Effectiveness: The ability to achieve True Positives (blocking malicious traffic) and True Negatives (forwarding valid packets).
- Performance Impact: The localized transactional delay (latency in milliseconds) introduced by deep packet inspection.
- Robustness: The systemic stability of the software stack when saturated with syntactically malformed streams.

6.2 Presentation of Findings: Test Matrix

The holistic behavior of the open-source firewall across the executed threat spectrum is documented below.

Table 3: Firewall Activity Test Matrix

Test ID	Description	Expected Result	Actual Result	Performance Impact (Latency - ms)	Evidence Reference (Log/PCAP)	Result (Pass/Fail)
TC-01	GSMA Layer 2: PSI Location Tracking	Blocked	Blocked	2 ms	sigfw.log:14:32:11, psi_blocked.pcap	Pass
TC-02	GSMA Layer 3: Velocity Check	Blocked	Allowed	1 ms	sigfw.log:14:38:20, velocity_allowed.pcap	Fail
TC-03	Legitimate Location Update (Check)	Allowed	Allowed	1 ms	legit_ul_allowed.pcap	Pass
TC-04	TCAP Avoidance: Malformed IMSI	Blocked	Allowed	2 ms	tcap_bypass.pcap	Fail
TC-05	Robustness: TCAP Fuzzing (10k packets)	No Crash	No Crash	5-10 ms	fuzzing_test.log	Pass

7. Conclusion

This laboratory study demonstrates the feasibility and technical value of constructing a high-fidelity SS7 signalling security research infrastructure using open-source utilities. The empirical data gathered from the test matrix indicates that while P1sec SigFW successfully thwarts basic, signature-reliant Category 2 tracking vectors (*TC-01*), it presents notable defensive shortfalls when challenged with complex threat models.

Specifically, the engine failed to enforce Category 3 rules due to the lack of an integrated state tracking database necessary for stateful velocity calculation (*TC-02*). More critically, the system proved susceptible to Parser Differential vulnerabilities (*TC-04*). When exposed to non-standard ASN.1 BER indefinite length structures, the firewall's Java-based decoder threw a parsing exception and defaulted to an insecure "Allow" state, while the target node successfully parsed and executed the nested payload.

These findings emphasize that securing legacy telecommunications infrastructure requires capabilities far beyond static signature matching. Future work will focus on expanding open-source testing tooling by engineering native TCAP/MAP dissection extensions for packet manipulation libraries and building specialized, open-source fuzzer frameworks to systematically validate the decoder robustness of core telecom switches.

Acknowledgements

The author wishes to express his deepest gratitude to his family and Elif Nurten for their unwavering support, continuous encouragement, and patience throughout the duration of this research project.

Ethics Declaration: This research did not require ethical approval as it involved the use of isolated laboratory environments and simulated network traffic using open-source tools, with no human participants or real-world subscriber data involved.

AI Declaration: Generative Artificial Intelligence (AI) tools have been used to assist with the translation and linguistic corrections of this article. The fundamental research, data analysis, and scientific findings are the author's original work.

References

3GPP (2024) *Technical Specification TS 29.002: Mobile Application Part (MAP) specification*. Available at: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1585 3rd Generation Partnership Project.

Brian (2017) *Cellular Lab Setup*. Security-Bits.de, 29 December. Available at: https://security-bits.de/research/cellular/lab_setup_new/ (Accessed: 5 December 2025).

GSMA (2017) *FS.07 – SS7 and SIGTRAN Network Security*. Version 4.0. Available at: https://www.gsma.com/solutions-and-impact/technologies/security/gsma_resources/fs-07-ss7-and-sigtran-network-security-v4-0/ (Accessed: 12 December 2025).

GSMA (2025) SS7: Securing a Legacy Protocol in a Modern Threat Landscape, and How Information Sharing Can Help to Mitigate', *GSMA T-ISAC Blog*, 28 July. Available at: https://www.gsma.com/solutions-and-impact/technologies/security/t-isac-blog/ss7-securing-a-legacy-protocol-in-a-modern-threat-landscape-and-how-information-sharing-can-help-to-mitigate/ (Accessed: 8 January 2026).

ITU-T (1996) *Recommendation Q.711: Functional description of the signalling connection control part*. Available at: https://www.itu.int/rec/T-REC-Q.711-200103-I/en International Telecommunication Union.

Kacer, M. and Langlois, P. (2017) *SS7 Attacker Heaven turns into Riot: How to make Nation-State and Intelligence Attackers' lives much harder on mobile networks*. Black Hat USA 2017. Available at: https://blackhat.com/docs/us-17/wednesday/us-17-Kacer-SS7-Attacker-Heaven-Turns-Into-Riot-How-To-Make-Nation-State-And-Intelligence-Attackers-Lives-Much-Harder-On-Mobile-Networks.pdf (Accessed: 20 December 2025).

Nick (2020) 'GSM with Osmocom Part 8: The Mobile Switching Center', *Nick vs Networking*, 10 May. Available at: https://nickvsnetworking.com/gsm-with-osmocom-part-8-ss7-sigtran/ (Accessed: 15 December 2025).

P1 Security (2024) *SigFW: Open Source Signaling Firewall*. GitHub Repository. Available at: https://github.com/P1sec/SigFW (Accessed: 10 January 2026).