

March 10, 2021

Communications Security, Reliability, and Interoperability Council VII

**REPORT ON**

**Session Initiation Protocol SECURITY CHALLENGES AND MITIGATION**

*Working Group 6: SIP Security Vulnerabilities*

Table of Contents

1 Results in Brief 4

1.1 Executive Summary 4

2 Introduction 4

2.1 CSRIC Structure 5

2.2 Working Group 6 Team Members 6

2.3 Subject Matter Expert Contributions 7

2.4 Acknowledgements 7

3 Objective, Scope, and Methodology 7

3.1 Objective (as provided by the FCC) 7

3.2 Scope 8

3.3 Methodology 8

4 Background 8

4.1 SIP Protocol 8

4.2 SIP Architecture 10

4.3 SIP Deployment 16

4.3.1 SIP Interoperability: SIP Protocol Interworking 17

4.3.2 SIP interoperability: Codecs and Transcoding 18

4.3.3 SIP interoperability: NAT Traversal 19

4.4 SIP Service Enablers 20

4.5 SIP Security 21

4.6 Related Activities 22

5 Analysis of Known SIP Issues and Vulnerabilities 22

5.1 Spoofing 23

5.1.1 Caller ID Spoofing 23

5.1.2 Route Hijacking 25

5.2 Tampering 25

5.2.1 SDP Tampering 25

5.2.2 SIP Header Injection 26

5.2.3 SDP Injection 27

5.2.4 RTP / RTCP Injection 28

5.2.5 RTP Bleed 28

5.3 Repudiation 29

5.3.1 Toll Avoidance 29

5.4 Information Disclosure 30

5.4.1 Disclosure of Valid Extensions or Usernames 30

5.4.2 Fingerprinting Internal Infrastructure from Via Headers 31

5.4.3 Fingerprinting SIP Servers via Banner 31

5.4.4 Offline Brute Force Attack Using Captured Handshake 33

5.4.5 Online Brute Force Attack Against Authentication 33

5.4.6 Disclosure of Geolocation 34

5.5 Injection 35

5.5.1 SIP MITM 35

5.5.2 RTP MITM 35

5.5.3 SIP Text 36

5.6 Denial of Service 37

5.6.1 Denial of Service 38

5.6.2 Denial of Service Using SIP Based Numbers against Critical Infrastructure 39

5.6.3 SIP REGISTER / INVITE Flooding 40

5.6.4 Silent INVITE Denial of Service 40

5.6.5 SIP UDP: Amplification Attacks and Spoofing 41

5.7 Elevation of Privilege 41

5.7.1 SIP Registration Hijacking 42

5.8 Multi-Impact Attacks 42

5.8.1 SIP Header and SDP Tunneling 42

5.8.2 SIP Route Steering 43

5.8.3 SIP Registering vs Non-Registering User Agents 43

5.8.4 SIP Conferencing Information Disclosure 43

6 Findings, Gap Analysis, and Recommendations 44

6.1 Root Cause Findings 44

6.2 Gap Analysis 46

6.3 Emergency Services / 9-1-1 Considerations 48

6.4 Recommendations 49

6.4.1 Operators and vendors should adopt well-established security frameworks 49

6.4.2 STIR/SHAKEN and End-to-End Security 51

6.4.3 Further Study of SIP Primitives that Inhibit Better Security 51

7 Conclusions 52

8 Appendix 53

8.1 Acronyms and Definitions 53

# Results in Brief

## Executive Summary

Session Initiation Protocol (SIP) is an application-layer signaling protocol for creating, modifying, and terminating sessions with one or more participants. These sessions include Internet telephone calls, messaging, and multimedia conferences. Because SIP is used to initiate voice sessions, it is also important for the 911 service.

As SIP-enabled services and devices continue to become more widely used, there is an increasing need for the industry to prioritize safeguarding SIP assets against undesirable exploitations and attacks. Due to its open architecture, SIP-enabled services are vulnerable to a variety of different types of security threats, similar to security threats against Hypertext Transfer Protocol (HTTP) or any publicly available service on the Internet. In a network architecture where data and voice reside together, the architecture is vulnerable to multi-layered security threats including Distributed Denial of Service (DDoS), vulnerabilities in the Transmission Control Protocol (TCP) stacks of servers, and application layer vulnerabilities specific to SIP applications.

The Federal Communications Commission (FCC) directed CSRIC VII, via Working Group 6 (WG6), to review the security vulnerabilities affecting SIP, including the provisioning and operation of communication services. This report focuses on the cybersecurity risk inherent in any SIP-based network or system, with a particular focus on the threat surface and potential attack vectors related to SIP. This report is designed to outline how industry is addressing these vulnerabilities, identify gaps in industry action, provide updates on existing best practices relevant to SIP, and recommend additional best practices that, if implemented, would address such vulnerabilities and mitigate their associated risks.

This report identified four key actions that can be taken to considerably enhance the security of SIP-based systems. While all the mitigations listed in this document are important to achieve very high reliability and security of SIP-based services, these four stand out:

1. Use TCP transport protected by TLS exclusively, with a PKI-based authentication scheme. This requires upgrades to many existing systems.
2. Keep components up to date with security patches. Many systems are unable to be patched rapidly or at all. Those systems should be replaced.
3. Deploy STIR/SHAKEN more widely (e.g. non-carrier and international).
4. For systems where massive TDoS would cause severe repercussions (e.g., emergency services), deploy volumetric DDoS mitigation services that include call processing as well as packet processing mitigations.

# Introduction

SIP is a signaling protocol primarily used for setting up, modifying, and tearing down real-time multimedia sessions, such as voice and video calls over the internet. SIP/2.0 was originally defined in Request For Comments (RFC) 2543 by the Internet Engineering Task Force (IETF), and subsequently updated by [RFC 3261](https://tools.ietf.org/html/rfc3261). SIP/2.0 is the current version of the protocol used by hundreds of millions of devices and users worldwide.

The recent global pandemic highlights the importance and reliance the world has on internet applications and services to entertain, educate, inform, transact business, and communicate. However, these applications and services are vulnerable to fraud, identity theft, and denial‑of‑service attacks. Discussing vulnerabilities in SIP and best practices for mitigating them is relevant and important.

SIP is a text-based client-server protocol similar to HTTP; likewise, SIP messages present many opportunities for threats like spoofing, hijacking, and message tampering. SIP can use well-known protocols for transport such as TCP and User Datagram Protocol (UDP) during a session, in which SIP-enabled services directly inherit the vulnerabilities of those protocols.

SIP is a reasonably complex protocol that often relies on other protocols such as Session Description Protocol (SDP) and the Real-Time Protocol (RTP). It has moreover been updated or expanded by numerous extensions to headers, message types, and the like over the course of decades. As these protocols become more complex and depend or interact with other protocols, the attack surface expands accordingly.

In order to taxonomize the various vulnerabilities that SIP-enabled services may face, CSRIC VII elected to structure them into a model of threats developed by Microsoft known as STRIDE. It provides a mnemonic for security threats in six categories: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege (STRIDE). By grouping the various attack classes, common root causes and mitigation strategies can be more easily identified.

This document is designed to educate and highlight vulnerabilities and security concerns pertaining to SIP, and to present corresponding best practices for mitigation tactics and strategies. Further, this document highlights cases where additional work is required to address unsolved vulnerabilities. Securing SIP, or any reasonably complex protocol or system, is not a destination; it is a journey across a landscape of layered defenses to provide a defense in depth approach.

## CSRIC Structure

CSRIC VII was established at the direction of the Chairman of the FCC in accordance with the provisions of the Federal Advisory Committee Act, 5 U.S.C. App. 2. The purpose of CSRIC VII is to provide recommendations to the FCC regarding ways the FCC can strive for security, reliability, and interoperability of communications systems. CSRIC VII’s recommendations will focus on a range of public safety and homeland security-related communications matters. The FCC created informal subcommittees under CSRIC VII, known as working groups, to address specific tasks. These working groups must report their activities and recommendations to the Council as a whole, and the Council may only report these recommendations, as modified or ratified, as a whole, to the Chairman of the FCC.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Communications Security, Reliability, and Interoperability Council VII** | | | | | |
| **Working Groups** | | | | | |
| Working Group 1: Alert Originator Standard Operating Procedures & Duplicate NWS Alert | Working Group 2: Managing Security Risk in the Transition to 5G | Working Group 3: Managing Security Risk in Emerging 5G Implementations | Working Group 4: 911 Security Vulnerabilities during the IP Transition | Working Group 5: Improving Broadcast Resiliency | Working Group 6: SIP Security Vulnerabilities |
| Chair:  Craig Fugate, APTS/  CoChairs:  Michelle Mainelli, NWS  Terri Brooks, T-Mobile | Chair:  Kathy Whitbeck, Nsight | Chair:  Farrokh Khatibi, Qualcomm | Chair:  Mary Boyd, West Safety Services | Chair:  Pat Roberts, Florida Association of Broadcasters | Chair:  Danny McPherson, Verisign |
| FCC Liaisons:  James Wiley/  David Munson | FCC Liaison:  Kurian Jacob | FCC Liaison: Steven Carpenter | FCC Liaison: Rasoul Safavian | FCC Liaison: Robert “Beau” Finley | FCC Liaison: Ahmed Lahjouji |

Table - Working Group Structure

## Working Group 6 Team Members

Working Group 6 consists of the members listed below.

|  |  |
| --- | --- |
| **Name** | **Company** |
| Danny McPherson – Chair | Verisign |
| Jamal Boudhaouia | CenturyLink |
| Pierce Gorman | T-Mobile |
| Mark Hess | Comcast |
| Zeeshan Jahangir | T-Mobile |
| Susan M. Miller | ATIS |
| Thomas B. Nachbar | SGE |
| Richard E. Perlotto II | The Shadowserver Foundation |
| Jon Peterson | Neustar |
| Krisztina Pusok | American Consumer Institute |
| Evans Roberts Jr. | AT&T |
| Brian Rosen | NENA |
| Dorothy Spears-Dean | NASNA |
| John Totura | Comtech |
| Brian Trosper | Verizon |
| Steve Watkins | Cox Communications |
| Vladimir Wolstencroft | Twilio |
| Ahmed Lahjouji - Liaison | FCC |

Table - List of Working Group Members

Working Group 6 Alternates are listed below.

|  |  |
| --- | --- |
| **Name** | **Company** |
| Steve Barclay | ATIS |
| Ramone Torres | ATIS |
| Chris Wendt | Comcast |
| Damien Whaley | Cox |
| Shaun Slatton | Cox |
| Yong Kim | Verisign |
| Matthew Thomas | Verisign |
| Eric W. Kroymann | Verizon |

Table - List of Working Group Alternates

## Subject Matter Expert Contributions

|  |  |
| --- | --- |
| **Name** | **Company** |
| Henning Schulzrinne | Columbia University |
| Jonas Borgesson | Twilio |
| Tony Friar | Velona Systems LTD |
| David Maxwell | GSMA |
| Sean Turner | sn3rd LLC |
| Christopher Desmond | Verizon |
| Jackie Wohlgemuth | ATIS |

Table - List of Subject Matter Experts

## Acknowledgements

CSRIC VII, Working Group 6, would like to acknowledge the significant contributions of each of its members, alternates, member organizations, and subject matter experts that positively influenced this work product. Without their expertise, participation, analysis, and contributions throughout the process, the report findings, conclusions, and recommendations contained herein would not have been possible. Member insight, focus, and leadership throughout the Working Group 6 process, particularly in a time our nation is battling a pandemic, is evidenced by the quality of this report’s analysis, recommendations, and conclusions.

In particular, CSRIC VII, Working Group 6, would like to thank all the subject matter experts and other volunteers. This work would not have been possible without their collaboration and contributions, particularly in an unprecedented time of crisis. We would also like to explicitly acknowledge Matthew Thomas of Verisign for serving as editor of the report. Finally, CSRIC VII, Working Group 6, would also like to acknowledge and thank ATIS and Verisign for the supporting infrastructure that facilitated the work progress.

This CSRIC VII Report is the product of nearly two years of thoughtful and detailed recurring conference calls and subcommittee work. This report benefited from cooperative consideration and a strong motivation to improve the resiliency of SIP and emergency communications capabilities.

# Objective, Scope, and Methodology

## Objective (as provided by the FCC)

SIP is an application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants.[[1]](#footnote-1) These sessions include Internet telephone calls, multimedia distribution, and multimedia conferences. Because SIP is used to initiate voice sessions, it is also important for 911 service. The FCC directs CSRIC VII to review the security vulnerabilities affecting SIP that affect the provision of communications service. CSRIC VII should outline how industry is addressing these vulnerabilities, identify any gaps in industry action, update any existing best practices relevant to SIP, and develop additional ones that, if implemented, would address such vulnerabilities and mitigate their associated risks, including the promotion of end-to-end security.

## Scope

The purpose of this document is to describe known SIP vulnerabilities, attacks, and security considerations that currently have been detected in the wild, theorized, or otherwise proposed. While interdependencies on underlying infrastructure, network environments, operating systems, and SIP service enablers are discussed in this document, comprehensive exploration of these externalities is beyond the scope of this document.

## Methodology

This document aims to enumerate and describe existing security threats that SIP-based application service providers and vendors should be cognizant of from an operational perspective. While a multitude of security mechanisms have been proposed for SIP-based infrastructures, there are still a wide array of vulnerabilities and threat models that affect this architecture. These types of vulnerabilities range from resource exhaustion attacks, spoofing, tampering, information disclosure, etc.

The rest of this document is organized as follows:

* Section Four provides a background of how SIP-based infrastructures are designed, commonly deployed, and how their components interact.
* Section Five codifies known SIP issues and vulnerabilities into a threat model divided into the STRIDE mnemonic for security threats into various categories. Explanations of each vulnerability are subsequently expanded upon with details regarding the vulnerability, its impact, and potential mitigation options.
* Section Six summarizes CSRIC VII’s findings and presents a Gap Analysis of on-going work efforts in various standardization groups to address SIP security issues.
* Section Seven concludes this CSRIC VII report.

# Background

This section provides an overview of SIP deployment and architecture as they relate to SIP vulnerabilities and best practices. This section will further expand on network and service design, testing, maintenance, assumptions, and requirements.

## SIP Protocol

As established in RFC 3261, SIP is an application layer signaling protocol designed for creating, modifying, and terminating multimedia sessions over the internet. It is a text-based message protocol that draws heavily on the Simple Mail Transfer Protocol (SMTP) and HTTP protocols. SIP messages, which include header fields and a message body, are either a request or a response, where responses largely follow HTTP-like formats. Further refinements and extensions of the original text-based SIP protocol are described in numerous RFCs. A complete list of related RFCs can be found here: <https://www.packetizer.com/ipmc/sip/standards.html>

SIP protocol capabilities include:

* Location of endpoint(s)
* Media capabilities of endpoint(s)
* Availability of endpoint(s)
* Session establishment and mid-call changes
* Session transfer and termination

Often, discussion of communications protocols refers to a model developed by the International Standards Organization (ISO) called the Open Systems Interconnection (OSI) model. Significant content is available on this model and it won’t be reproduced here, but briefly, the model has 7 layers. Some of the layers are rarely referred to, but layers 2 through 4 and layer 7 are referred to often. The table below outlines the names of the layers and provides examples of protocols that are said to be representative of a given layer. The Transport Layer Security (TLS)[[2]](#footnote-2) protocol is a cryptographic protocol designed to provide communications security over a computer network. The TLS protocol, which is important to SIP deployments, nominally is implemented at the transport layer; however, in practice it spans multiple layers of the OSI model.

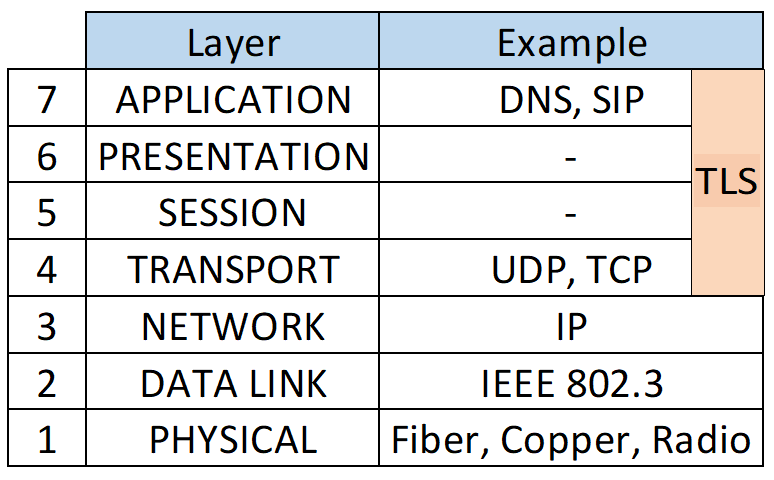


Figure - OSI Model

One reason it is valuable to reference the OSI model is to illustrate a common principle of network security, which is that security designs are typically layered. The focus of this report is on SIP, ostensibly an application layer protocol, but it is noteworthy that good security design should be inclusive of consideration for the substrate protocols as well, particularly transport, network, and data link layers, and the associated protocols utilized in those substrate layers.

TLS is a good example of how protocol layering models are helpful conceptualizations, but sometimes real-world implementations deviate from clear conceptual model layering. In this case, it is arguable which layer(s) TLS would reside in, but in the end it is irrelevant. The thing to remember is that TLS is a very helpful protocol for securing SIP, which is often transported using TLS, especially in “line side” (also sometimes called “access”) connection applications.

SIP implementations (or “stacks”) supporting SIP defined in RFC 3261 are required to support either TCP or UDP transport protocols to carry the SIP protocol packets and can either use the default port 5060 for either TCP or UDP, or a custom port number defined for/by the SIP stack.

It is worth noting that one of the main challenges of SIP deployment is reliable interworking between varying implementations of SIP protocol. A common example of this challenge is the lack of SIP implementations that support both TCP and UDP transport. UDP is commonly supported; however, it is frequently constrained by IPv4’s minimum supported packet size of 576 octets, which results in any SIP message larger than 512 octets (576 less header and feasible options) potentially becoming fragmented across multiple UDP packets[[3]](#footnote-3). High rates of UDP fragmentation would otherwise require reassembly by the receiver, a task which has proven to be a significant challenge for some SIP implementations. While larger packet buffers and mechanisms such as path maximum transfer unit (PMTU) discovery[[4]](#footnote-4) have since been introduced that enable confident use of larger packet sizes without requiring fragmentation along the path or reassembly by the receiver, there remain places where packets with payloads larger than 512 octets are simply not feasible. Another problem of UDP-based SIP is its inability to utilize security mechanisms such as TLS. Similarly, because of the amount of state required to maintain connection-oriented transport connections utilizing TCP, implementations of SIP can be susceptible to load, and potentially vulnerable to attacks aimed at exploiting or exhausting resources required for state management, as is discussed in more detail in later sections.

## SIP Architecture

SIP architecture design is a combination of both data network design and service network design. Data network design is concerned with data network elements and the interconnection of those elements. Common data network elements include end nodes, which are often computer servers or computer clients of servers, Ethernet switches, routers, and firewalls. While there are many variations and hybrids of these elements, they are the most common components of a SIP architecture.

The following SIP components are commonly seen throughout various SIP architectures and are depicted in the classic “SIP trapezoid” figure below:

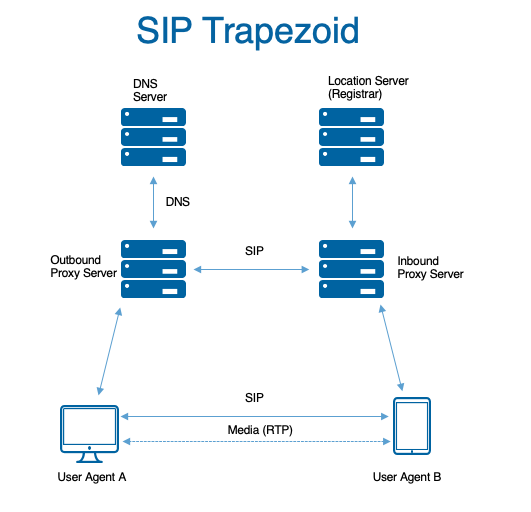


Figure - SIP Trapezoid

* User Agent Clients (UACs): an application that initiates a SIP request to a UAS
* User Agent Servers (UASs): an application that receives SIP requests from a UAC and returns responses to those requests.
  + SIP Redirect Servers (a special-purpose UAS that issues 3xx class responses): enables users to temporarily or permanently change locations and still be contactable through the same SIP identity. Also used by telephone number translation services.
* SIP Registrar: a network service where UAs can register to receive SIP requests for a particular address-of-record (AoR)
* SIP Proxy Servers: a network service responsible for relaying or routing requests between user agents.
  + Non-standard intermediaries such as back-to-back User Agents (B2BUAs), most commonly in application-layer firewall implementations known as “Session Border Controllers” (SBCs).

A SIP SBC is a special, application-layer firewall. An SBC provides three main functions:

1. Network security boundary for policy enforcement
2. Topology hiding using Network Address and Port Translation (NAPT)
3. SIP protocol “normalization”

The SBC typically consists of a B2BUA that rewrites SIP messages in both directions to accomplish network boundary policy enforcement and protocol normalization, making itself the endpoint for signaling in both directions. It also typically “anchors” media – making itself the endpoint for media in both directions, relaying media packets from inside to outside. It does this by rewriting the SDP portion of the message in both directions. In doing so, it can provide NAPT traversal functions for the caller.

Below is a diagram of a high-level network design using an SBC for the function of NAPT.

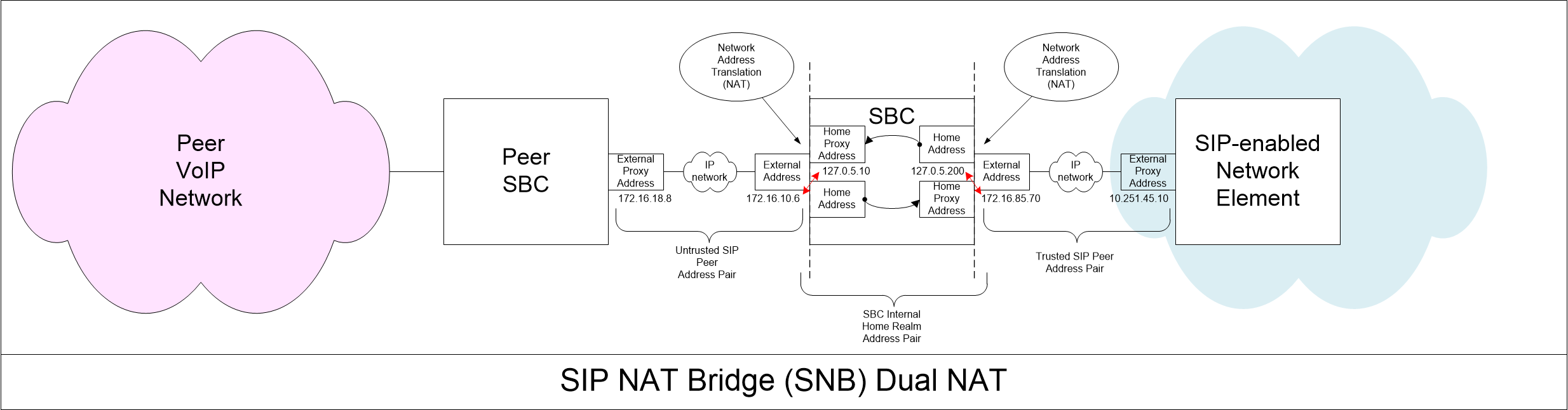


Figure - Network Design Using an SBC for NAPT

Some NAPT functions only translate the IP address in the IP header of a packet. Other NAPT functions also translate the transport layer port number. A “SIP NAT” is a specialized NAT for the IP addresses of the packet carrying the SIP message, as well as rewriting IP addresses carried in the SIP message itself. In the previous diagram the IP address of a packet carrying a SIP message is translated twice while passing through the virtual core of the SBC. The reason for the dual translation is not to improve security but to provide an internal network layer binding mechanism to support use of overlapping and potentially duplicate IP address space on external (often virtual) network interfaces. This enables interconnection of multiple networks whose address space is identical.

Layer 7 application protocols which embed IP addresses within the payload of the application protocol present a special use case issue for NAT. SIP is an example of a special use case for NAT/NAPT in that IP addresses are embedded within the headers and body of a SIP message as well as the network layer IP packet headers. IP addresses can be found in SIP headers such as the Request-URI and/or Contact headers of a SIP INVITE, as well as within the SDP message body. IP addresses for the RTP media descriptions within SDP message bodies in SIP are very commonly not the same IP addresses as are used for SIP signaling.

An important reason for using different IP addresses for SIP signaling versus RTP media is because the computer processing work required for media packets is significantly different than the processing work required for signaling packets. It also enables the deployment architecture mentioned later in Section 4.3. SIP call processing is delay tolerant compared to RTP media. RTP media is sensitive to delay, variations in delay (jitter), and most especially packet loss. Sometimes RTP media is managed through a “media gateway” separate from SIP signaling, and SIP signaling is managed through a SIP proxy or sometimes a B2BUA.

There is an important operational consideration for using different IP addresses for RTP media as compared to SIP signaling. Testing and troubleshooting call flows is done using packet capture and protocol analysis tools. Because RTP media uses significantly more bits per second compared to SIP signaling, packet capture buffers fill very quickly when listening for both SIP and RTP packets. Using different IP addresses (and VLANs) for signaling versus media packets improves efficiency of network monitoring packet filters and troubleshooting operations.

It is worth emphasizing that conventional firewalls are not often used to protect service provider SIP networks. Conversely, SBCs are not often used to provide the network security functions offered by conventional firewalls such as support/protection specific for the Domain Name System (DNS)[[5]](#footnote-5), Hypertext Transfer Protocol (HTTP)/Hypertext Transfer Protocol over TLS (HTTPS)[[6]](#footnote-6), and Simple Mail Transport Protocol (SMTP)[[7]](#footnote-7). Instead, both SBCs and firewalls are used to support/protect the specific network functions for which they were designed. SBCs are often used as Application Level Gateways (ALGs). The protocol requirements for DNS, HTTP, and SMTP are significantly different than the requirements for SIP, SDP, and RTP. The evolution of firewalls and SBCs reflect these differences.

It should be useful to illustrate and compare network architectural choices for firewalls as compared to SBCs. For larger more complex non-SIP networks, it is common to put web servers and other systems which communicate with client applications over the internet into a layered boundary architecture with firewalls on an external boundary facing the internet, and an internal boundary facing the more trustworthy part of the communications network.

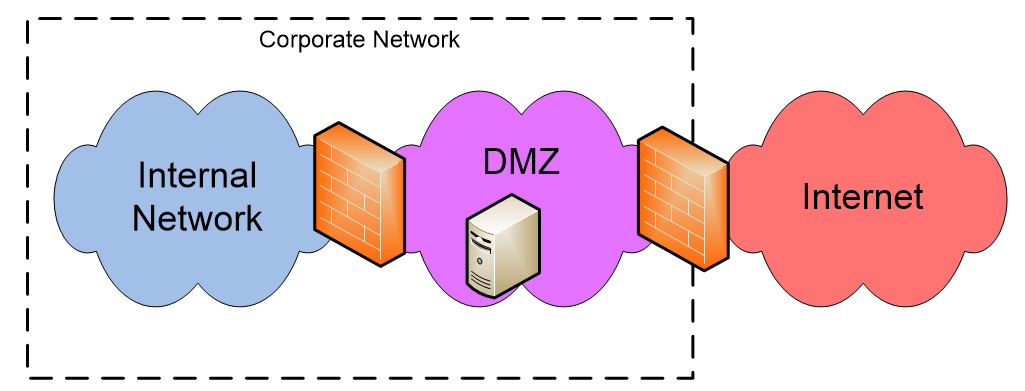


Figure - Typical Corporate Network

The trusted, semi-trusted, untrusted network architecture is also used in service provider SIP networks where the firewall between the semi-trusted and untrusted networks is an SBC and the elements in the DMZ semi-trusted network are voice and messaging services elements such as IP Multimedia Subsystem (IMS) control functions and voice switching elements. An illustration of the trusted, semi-trusted, and untrusted network architecture showing examples of both SBCs and firewalls is depicted in the “walled garden” figure below.

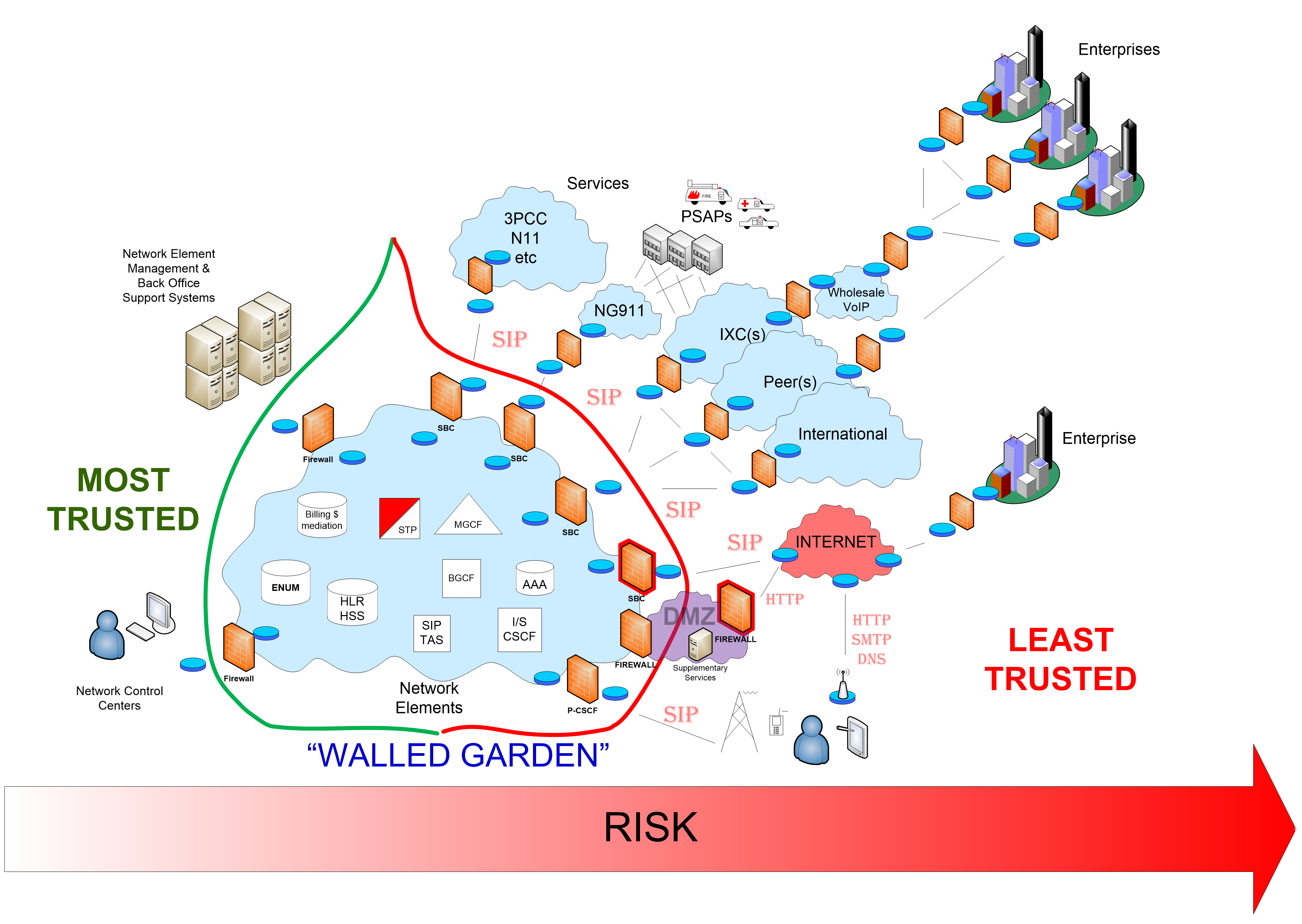


Figure - Network Architecture Layout Based on Trust and Risk

Because network elements such as routers, firewalls, SBCs, and SIP proxies are essentially specialized computer platforms (or sometimes are network virtual functions operating on general purpose computing hardware), they will at times require hardware or software upgrades or modifications, which require the element to be temporarily removed from service. Because the voice or video calls or other sessions signaled with SIP may be mission critical (e.g., 911 calls), it is important for service providers to design resiliency into their SIP architecture. A robust architecture will take into account requirements for load distribution, efficient capacity management, and failure recovery for planned and unplanned outages.

Figure 6 depicts an example of a physical Voice of IP (VoIP) network inter-connection infrastructure that might be found in a large enterprise or service provider network. The remaining text in this section explains the rationale for the various elements and their interconnection in this highly resilient design.

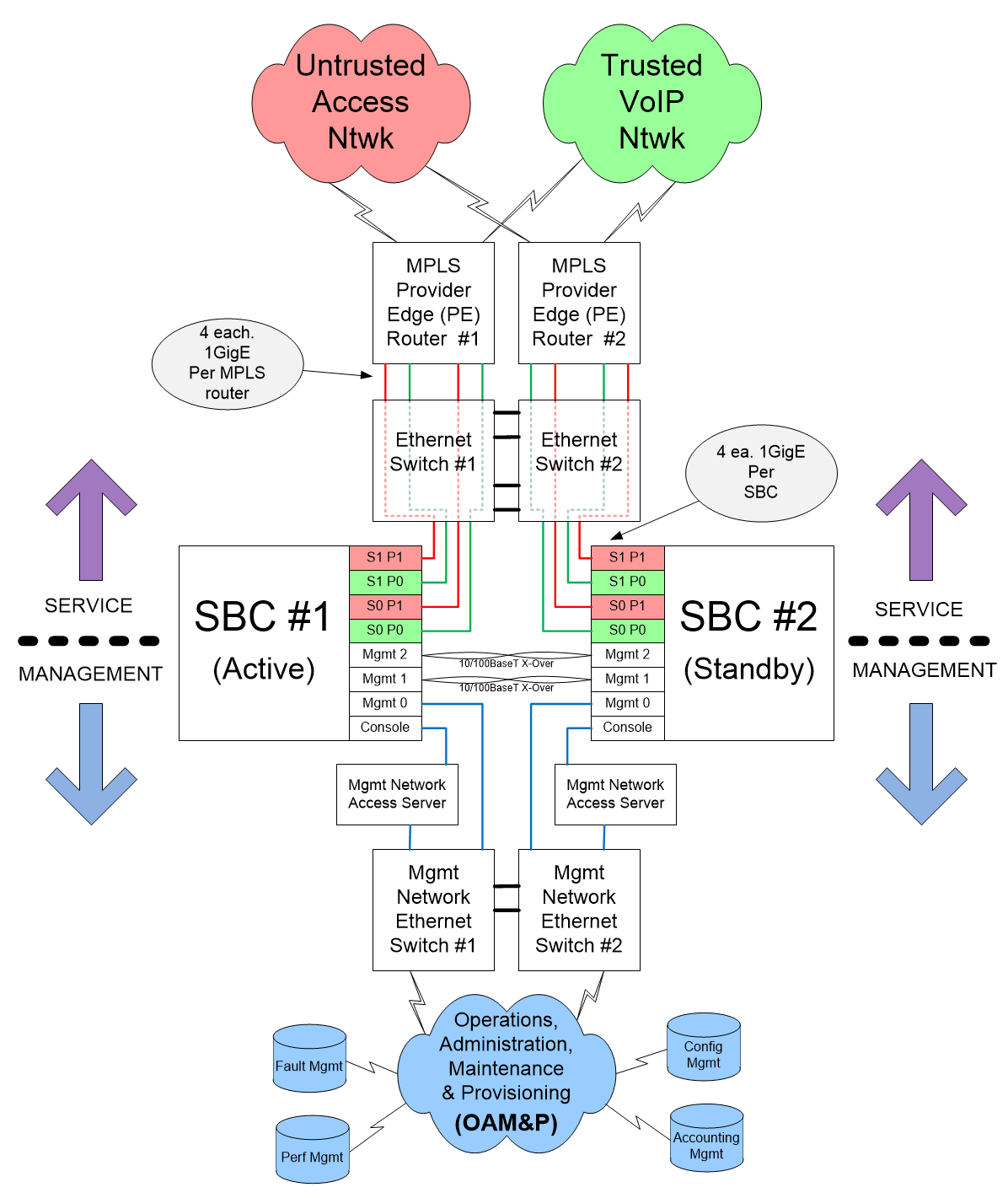


Figure - Example VoIP Network Inter-Connection Infrastructure Design

At the top of Figure 6 is a redundant pair of interconnection routers with diverse connections to the wide-area network in the red and green clouds, with multiple connections to a redundant pair of Ethernet switches, which are themselves interconnected, and connected to a pair of SBCs whose management interfaces also connect to a redundant pair of interconnected Ethernet switches (and likely served by another pair of internal redundant routers in the blue cloud). This highly fault-tolerant design protects against element and link-level failures.

The pair of SBCs are interconnected with two redundant cables (intentionally to eliminate reliance on a single Ethernet switch). These interconnections act as an inter-SBC “heartbeat” and are used to checkpoint session state information and to detect when a failover from the “Active” SBC to the “Standby” SBC is required. Ideally, the “heartbeat” connections permit failover in less than 50 milliseconds which implies a minimal loss of up to 3 RTP packets (assuming 20 msec packetization rates).

The SBCs also have Ethernet and serial interconnections to an internal Operations, Administration, Maintenance, & Provisioning (OAM&P) network used for managing network elements. The SBC management Ethernet connection is used for communicating with fault, configuration, performance, and accounting application support systems. The console port is an RS232 serial connection for “dumb terminal” access to the SBC using an application such as Telnet through the Access Server. The console port provides limited remote access to the SBC hardware but is not a replacement for the Ethernet management interface.

The “service” interfaces in the SBCs are in two diverse slots, each with two Ethernet ports. In the example, the cards in the two slots are both active and are not redundant to each other. One card has a port connected to the trusted network, and the other port connected to the untrusted network so that a failure of a card may degrade service but it does not result in a complete failure to pass traffic between the trusted and untrusted networks, which would be the case had the design engineer chosen to connect both ports of a single card to only the trusted, or untrusted networks. The design choice illustrates both load distribution and resiliency to failure.

An active/standby high-availability (HA) design and that enables switching, which element is the active system only when required for planned maintenance or unplanned outages, illustrates both architectural and operational best practices. In addition to an HA pair of SBCs, the example VoIP infrastructure includes a redundant router and Ethernet switch design which provides a fast failover HA capability.

## SIP Deployment

An ideal SIP architecture deployment would segregate open public internet accessible services from closed non-public, private, non-internet accessible services. This means segregating IP routing infrastructure into two zones, one that advertises IP address route prefixes to the internet and one that does not, as depicted in Figure 7. The simplest form of this is a private physical circuit to a service provider.

Technologies exist that allow this separation virtually over the same physical medium, e.g. MultiProtocol Label Switching (MPLS)[[8]](#footnote-8). While Internet Protocol Security (IPsec)[[9]](#footnote-9) can be used to secure SIP at the network layer, it is not adequate if data is transported over the public internet as the IPsec endpoints are still exposed to denial of service (DoS) risks. By segmenting the architecture, the most critical SIP infrastructure is prevented from being exposed to all of the inherent risks associated with public internet accessibility.

An additional benefit to segregation is the elimination of “shared fate” scenarios. If the public IP facing infrastructure is impacted, it does not impact the private non-internet facing services. An example of critical infrastructure that should not be internet-exposed are carrier SIP interconnects. The primary NG9-1-1 SIP interconnects should also not be exposed, but secondary or tertiary connections using Internet routable addresses may be needed to achieve availability goals at the expense of additional DoS exposure.



Figure - SIP IP Address Segregation

### SIP Interoperability: SIP Protocol Interworking

The “SIP Protocol normalization” function mentioned in the discussion on SBCs above is one of the key differentiating factors in describing the difference between a conventional firewall, which operates mostly in OSI layers 3 and 4, and an SBC which performs significant protocol interworking functions in OSI layer 7.

Perhaps the most common example of the “fix-up” interworking functionality on SBCs is called Header Manipulation Rules (HMRs). These are rules written using a lightweight scripting language, such as Regular Expressions (RegEx), which perform simple operations on some portion of a SIP message, most commonly the SIP INVITE method.

A very common example of an HMR is to strip information such as a “From” header display name on calls inbound to a service provider as untrusted content, ostensibly to be replaced by a SIP Proxy within the service provider’s network which has access to authoritative information in a Calling Name (CNAM) database. Other common HMRs will make modifications to the host portion of “Request-URI” or “Contact” headers in order to normalize call routing functions which may be based on DNS and DNS Storage of E.164 Numbers (ENUM)[[10]](#footnote-10), for example, instead of an IP address.

SIP is an “extensible” protocol by design. RFC 4904 is one example of an RFC containing an extension to the SIP protocol. RFC 5411 “The Hitchhiker’s Guide (THHG) to the Session Initiation Protocol (SIP)”, authored in 2009 contains a snapshot of RFCs under the SIP umbrella up until that date. There are over 100 RFCs and non-RFC references in THHG to SIP. Many more RFCs have been added to the umbrella since 2009. It is probably safe to say there is not a SIP stack in existence, other than perhaps in a Fuzz Test suite, which includes software for every SIP-related RFC.

It is a significant point that SIP interoperability and SIP interworking is a key operational challenge for VoIP service providers. Promoting SIP interoperability was the genesis for the creation of the SIP Forum[[11]](#footnote-11) and the focus of their annual SIP Network Operators Conference (SIPNOC). It is also noteworthy, that the last three conferences were dedicated almost entirely to presentations and discussion of the STIR/SHAKEN call authentication technology and regulation. STIR/SHAKEN is a complex and evolving set of SIP call authentication standards when we discuss more later.

### SIP interoperability: Codecs and Transcoding

SIP interoperability is more than just ensuring a common subset of signaling protocol functionality; session media interworking is also an important aspect of SIP architecture and deployment.

One of the most common types of SIP session is a voice call. The bi-directional voice media streams are described in the SDP message body that is carried within SIP messages such as a SIP INVITE. The most common media type in calls in North America is based on an audio Compression/DECompression (codec) algorithm often referred to as ITU G.711 µ-Law or Pulse-Code Modulation (or PCMU where the U refers to the µ in µ-Law). The G.711 codec is old and based on math which tries to encode the most significant audio frequencies for human speech. In G.711, speech is most commonly sampled in 20 millisecond intervals and sent in a UDP packet carrying RTP at the rate of 50 times a second.

The G.711 codec has worked well but it uses significantly more bandwidth as compared to modern mobile device codecs, which can reproduce the same and even more frequencies with even better fidelity but with a more efficient modulation which saves on radio spectrum usage. G.711 uses 64 kbps while a modern high definition “wideband” mobile codec may only use 20% as much bandwidth.

The implication of G.711 used in conjunction with more modern mobile device codecs is that calls between wireless and wireline devices must be transcoded between the originating and terminating audio codec types. There are numerous audio and video codecs in use around the world, which can make transcoding both complicated and expensive. The codecs that can be used for interoperation between SIP-enabled service providers is often a matter of bilateral negotiation as part of an Inter-Connection Agreement (ICA).

Codecs also often have parameters, which describe the modes of operation supported by the particular device codec implementation. The Enhanced Voice Service (EVS) codec designed for VoIP over LTE (VoLTE), for instance, has narrowband, wideband, and full-band modes of operation and a special Channel-Aware Mode designed to use special error handling to improve speech encoding in challenging radio conditions. Codec parameters and modes can be device or operator specific. At least one codec and its associated parameter configuration must be agreed upon between the caller device and the callee device or a media session is not possible. Therefore, in order for landline and mobile devices to communicate using VoIP, a transcoding media gateway often must be inserted into the media path between the caller and the callee.

Transcoding media gateways are complicated and expensive. It is common for a media gateway to be traffic engineered for a subset of supported codecs. If the traffic engineering assumptions are inadequate, calls may be blocked because of insufficient or nonexistent resources. Transcoding does not just apply to audio codecs. Transcoding can also be required for video codecs and for interoperation between Real-Time Text (RTT) and TTY.

### SIP interoperability: NAT Traversal

An important aspect of home and small-office environments is the reliance on a conventional firewall, typically integrated in a low-cost WIFI router, and the absence of an SBC to manage SIP signaling in TCP (or UDP) packets, and RTP media carried in UDP packets.

The earlier discussion on NAPT mentioned that protocol port numbers are often translated by a firewall (and/or SBC), partly as a function of security through obscurity, but more often for addressing the scarcity of public IPv4 addresses by multiplexing sessions for many users by sharing a single untrusted public IP address. They would have otherwise each required globally unique IP addresses and the same well-known or default port number, e.g., 5060 for SIP.

Because conventional firewalls do not typically have computing resources or sophisticated Layer 7 software for interpreting SIP signaling messages, they are unable to track SIP session state, including for example, the state of a device’s SIP registration established using a REGISTER message exchanged with a SIP registrar. Therefore, port bindings in the application-unaware firewall are ephemeral. That is, the port binding used for mapping an internal port number such as UDP 5060, to a port number such as UDP 5070 used on the untrusted interface facing the Internet, will time-out and the now-unbound port number returned to the pool for future NAPT bindings.

Because SIP and RTP may be traversing non-SIP aware firewalls, three protocols have been developed to assist with SIP interoperability and interworking; Session Traversal Utilities for NAT (STUN)[[12]](#footnote-12), Traversal Using Relays Around NAT (TURN)[[13]](#footnote-13), and Interactive Connectivity Establishment (ICE)[[14]](#footnote-14).

A SIP-aware device in a small-office/home-office environment can use the STUN protocol to discover the public IP address on the untrusted interface of the firewall/router providing connectivity to the Internet. The device can put the public IP address into the Contact header of a SIP INVITE when calling another SIP-aware device on the Internet.

If STUN fails because the local NAPT firewall won’t permit the inbound session attempt, the SIP-enabled endpoints can attempt to use a signaling relay supporting TURN. The TURN protocol allows a TURN client to establish a TURN session with a TURN server. SIP protocol packets are encapsulated by a TURN client attempting to establish a SIP session and the TURN server de-encapsulates the SIP messages and mediates the SIP session as a SIP proxy.

The ICE protocol is a sophisticated mechanism which uses both STUN and TURN and other functions to more reliably establish communications between endpoints using peer-to-peer signaling protocols such as SIP.

The NAT Traversal mechanisms in STUN, TURN, and ICE may, or may not be valuable depending upon the use case and architectures of the networks in which SIP endpoints are located.

## SIP Service Enablers

SIP heavily relies on subsystems and service enablers such as ENUM and DNS, and these factor into SIP security and privacy. While a comprehensive analysis of the security of these service enablers are beyond the scope of this document, some of the considered services include:

* The Domain Name System
  + DNS SRV/NAPTR resource records, used in some SIP deployments
* ENUM (public and private implementations) leveraged to route SIP requests to telephone numbers
* Public Key Infrastructure (PKI)[[15]](#footnote-15) leveraged in support of web-style TLS integrity and confidentiality of SIP signaling
  + PKI is also leveraged for STIR/SHAKEN[[16]](#footnote-16) anti-spoofing technology

Security considerations and best practice guidelines should be applied to configuring and setting these services to protect SIP systems. Some examples include RFC 5280, as it relates to PKI, and RFCs 1034, 1035, 2543, and 3263, as they related to DNS and SIP.

Several other security related works produced by previous CSRICs should also be considered to protect SIP, including, but not limited to, the following:

* CSRIC VI Report on Best Practices and Recommendations to Mitigate Security Risks to Current IP-based Protocols
* CSRIC IV Report on Cybersecurity Best Practices
* CSRIC IV Report on Remediation of Server-Based DDoS Attacks Final Report
* NIST 800-58 Security Considerations for Voice Over IP Systems
* NIST Cybersecurity Framework
* Center for Internet Security Controls

## SIP Security

SIP messages present various security challenges and opportunities for attacks like spoofing, hijacking, interception, and message tampering. Messages may contain information that sending and receiving parties wish to keep private. The message body could also include sensitive user information, such as IP addresses. SIP messages can also be abused to possibly result in unauthorized access or Denial of Service attacks.

Beyond network isolation, firewalls, and other network segmentation techniques described in the proceeding sections, the security of the protocol level contents of SIP and related media needs to be considered for a more comprehensive security strategy. It is fairly common today to utilize protocol encryption using TLS to protect the open internet use of SIP. For example, many over-the-top mobile and desktop clients that extend the ability to use a telephone service beyond managed devices that are secured within the trust boundaries of a managed network employ TLS protection of signaling and media. However, current security practices are being extended to consider attack vectors that may involve provider network penetration. Whether due to external hackers or even unauthorized internal employees that have access to internal privileged parts of the network, giving them the ability to access unprotected protocols or data related to customer calls, including Customer Proprietary Network Information (CPNI). The scope of these security breaches is complex and a general topic that many providers are only beginning to comprehensively address. Insider threat programs should be considered for future comprehensive reviews of SIP Security, or more broadly in the industry. Both protocol level encryption, and data at rest encryption should be considered for protecting customer data, especially when considering internal security threat scenarios.

**SIP protocol level network encryption**

The baseline SIP specification[[17]](#footnote-17) focuses on web-style security procedures, such as the use of symmetric keys to authenticate the senders of requests, the use of TLS to secure a channel for hop-by-hop security and recommends the use of Secure Multipurpose Internet Mail Extensions (S/MIME)[[18]](#footnote-18) for message body security. While S/MIME has seen minimal deployment, SIP Digest Authentication[[19]](#footnote-19) for requests and unidirectional TLS are frequently utilized. Operators often leverage closed networks to prevent attackers from tampering with SIP signaling. The open and distributed nature of SIP and its interaction/dependency with the variety of SIP service enablers, makes establishing a secure environment a very difficult and complicated task. As discussed earlier in this report, the SIP protocol can be used over various transport protocols, such as TCP and UDP. As a result, SIP will inherit those transport protocol’s vulnerabilities, such as spoofing, session hijacking, etc.

## Related Activities

There is significant ongoing work on SIP security in the broader industry addressing security challenges as new vulnerabilities arise. The IETF, which has change control over SIP, continues to produce new specifications on the security of SIP, SDP, and RTP, much of which occurs in either the SIPCORE working group or in the STIR WG. ATIS’s[[20]](#footnote-20) IP-NNI Task Force, in conjunction with the SIP Forum[[21]](#footnote-21), conducts SIP security work relevant to the North American IP NNI environment. 3GPP[[22]](#footnote-22) has its own IMS SIP security profile in 24.229 and related specifications. The GSMA[[23]](#footnote-23) SIPSEC effort tabulates in particular vulnerabilities specific to the implementation of SIP in international mobile networks. Vendors and operators are also advised to track known SIP implementation vulnerabilities logged in resources such as the National Vulnerability Database (https://nvd.nist.gov/)

# Analysis of Known SIP Issues and Vulnerabilities

The following classes are proposed for SIP security issues and the initial breakdown is done via the Microsoft STRIDE[[24]](#footnote-24) threat model methodology. The issues are grouped into attack classes, which often share common root causes and similar mitigation strategies. Vulnerabilities considered in this document may be intrinsic to the SIP protocol (design gaps or flaws), or extrinsic vulnerabilities that arise from implementations or deployments. There are similarly categories of implementation/deployment in which these extrinsic vulnerabilities may arise: landline carrier (IMS), mobile carrier (IMS), over-the-top (including CPaaS), enterprise, conferencing, consumer, and emergency services. Many practical security problems facing SIP today are predicated on the environment in which the protocol is used. In order to scope this table and maintain focus, vulnerabilities introduced by and inherent to common substrate protocols (e.g., TCP SYN floods when TCP is the transport protocol) are not catalogued here.

|  |  |
| --- | --- |
| Attack Class | Description |
| Spoofing | * Caller ID spoofing   + Both calling number and calling name * Leverage Spoofing to Gain trust of users   + Impersonation emergency services * Route hijacking / fraudulent retargeting |
| Tampering | * Tampering with SDP   + Tampering with media keying material   + Tampering with non-SDP SIP bodies * Tampering with SIP headers * Tampering with RTP streams   + Tampering with RTCP |
| Information Disclosure | * Implementation Fingerprinting * Disclosure of internal infrastructure coming from Via headers * Disclosure of Server and endpoint information from Server header * Disclosure of Geolocation * Disclosure of Digest handshake via sniffing a clear text connection * Media eavesdropping   + Disclosure of RTP stream   + Disclosure of media nodes and IPs to third parties * Leaking closed network/proprietary headers outside trusted environments |
| Repudiation | * Toll avoidance |
| Injection | * RTP Injection and MitM attacks * Clear text SIP INVITE may leak SRTP key * SIP MitM / Injecting proxy headers   + BYE / CANCEL Injection   + Injecting fraudulent UPDATE, MESSAGE, INFO, etc. into session |
| Denial of service | * INVITE Flood and denial of service * REGISTER Flood   BYE / CANCEL Flood (with “silent INVITE”) |
| Elevation of Privilege | * SIP registration hijacking |

Table – Analysis of Known SIP Issues and Vulnerabilities

## Spoofing

Spoofing is a broad term for situations where a party during a communication falsely identifies itself as some other party, in order to gain some form of an illegitimate advantage. Spoofing can take many forms through the exploitation or misuse of various protocols and systems. This category of STRIDE is concerned with authenticity.

### Caller ID Spoofing

**Impact**

Allows attackers to spoof a calling party’s number (“caller ID”) for social engineering purposes to gain the trust of end user targets. Spoofing has been carried out to impersonate trusted endpoints as well as perform "neighbor spoofing", using either the same area code and telephone prefix of the person being called, or the name of a person or business in the area. Caller ID spoofing techniques misrepresent the display number (and in some cases display name) rather than the URI(s) needed in the actual communication process for return routability to the caller's phone.

Spoofing caller ID is a capability attackers leverage to prevent traceback (i.e. repudiation) or to elevate privilege.

**Details**

Ability for an attacker or malicious user to spoof the caller ID of the SIP INVITE request by causing the From header field value or the P-Asserted-Identity (depending on the infrastructure and device that the SIP server is deployed) to deliver a chosen value.

It is a common feature of consumer applications to allow users to specify a custom caller ID. If an attacker has direct access to SIP trunking, they have the ability to set their own headers. In a Man in the Middle (MITM) attack scenario in which SSL / TLS is not used hop-by-hop, active attackers can edit these fields on traffic traversing transit, or they can cut-and-paste and replay modified signaling, though attackers rarely have to resort to these measures thanks to the ease of simply falsifying headers at origination.

INVITE sip:RTP\_AVP@192.168.1.11:5080 SIP/2.0

Via: SIP/2.0/UDP 172.23.32.41:5060;branch=z9hG4bK14c25803

Max-Forwards: 70

**From: "asterisk" <sip:asterisk@192.168.1.11>;tag=as4536ce29**

To: <sip:RTP\_AVP@192.168.1.1:5080>

Contact: <sip:asterisk@172.23.32.41:5060>

Call-ID: caler\_test@192.168.1.11::5060

CSeq: 102 INVITE

User-Agent: Asterisk PBX 15.3.0-rc2

Date: Mon, 04 Nov 2019 21:06:33 GMT

Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY, INFO, PUBLISH, MESSAGE

**Mitigation Options**

* **Number verification of “From” number:** The entity initiating the call should have ways to ensure that only verified numbers and parameters are used in the “From” / PAI header field value(s) of the SIP INVITE message. In some cases, upstream services are responsible for validating or rewriting the “From” or PAI header field value to ensure that spoofing cannot take place.
* **Cryptographic Signing protocols:** STIR/SHAKEN signing attests that a carrier has verified that the calling party number is being used legitimately. In the long term, this will prevent unauthorized use of telephone numbers by spammers. It will also stop downstream systems from tampering with or changing the From header field value.
* STIR / SHAKEN certificates and signatures have been designed and deployed to help mitigate call spoofing attacks.

### Route Hijacking

**Impact**

An attacker may set itself up to be the destination to calls for a particular telephone number or other Uniform Resource Identifier (URI) for the purposes of intercepting traffic to a target. For example, an attacker may want to pose as a financial institution, in order to capture information about legitimate customers of the business. This can be considered the inverse of calling party spoofing: it is called party spoofing.

**Details**

Routing of SIP calls, especially to telephone numbers, today remains a largely ad-hoc process. Those calls that are not simply dropped to the Public Switched Telephone Network (PSTN) are often routed based on local forwarding tables, or through querying a third-party service (via protocols like ENUM). The lack of centralized and authoritative databases for SIP call routing creates the possibility that attackers will compromise some components of this routing architecture to provision bogus data, or that they may intercept or spoof traffic from third-party services in order to direct calls to an Internet resource controlled by the attacker.

**Mitigation Options**

The use of TLS as described in RFC3261 can help originators ensure that signaling is being sent where it was intended to be sent on a hop-by-hop basis. Ultimately, the security of the routing infrastructure, especially private ENUM deployments, remains an open problem for the deployment of SIP as a replacement for the traditional telephone network. The infrastructure developed for STIR/SHAKEN, which itself relies on numbering databases, provides a vector that may help in the long term (i.e., Connected Identity for STIR[[25]](#footnote-25), “work in progress”) to mitigate this as well by cryptographically attesting the proper destination for calls to a given telephone number.

## Tampering

While tampering can refer to many forms of sabotage, in an information systems context it typically refers to malicious modification of data or processes. Data can be tampered while at rest, in transit, or within a process. Data tampering often coincides with other potential threats such as spoofing or privilege escalation. This category of STRIDE is concerned with integrity.

### SDP Tampering

**Impact**

Intermediary attackers can tamper with cleartext unsigned SDP in transit, potentially changing the media endpoints, and capturing or substituting keying material for Datagram Transport Layer Security Extension to Establish Keys for the Secure Real-Time Transport Protocol (DTLS-SRTP)[[26]](#footnote-26).

**Details**

By changing the media lines in SDP, which carry the IP addresses of the communications endpoints used for RTP or DTLS-SRTP, the attacker can copy media, deny service, or potentially change media in transit. Similarly, substituting keying material in transit can make end users believe that DTLS-SRTP has been successfully negotiated, and that a conversation is not susceptible to eavesdropping, when in fact the security association has been formed with the attacker rather than the intended media endpoint. SDP modification is common in many deployments due to Communications Assistance for Law Enforcement Act (CALEA)[[27]](#footnote-27) compliance. Differentiating these “legitimate” forms of tampering from genuine attacks is very difficult from the perspective of the endpoints establishing a SIP session.

**Mitigation Options**

* **Use of cryptographic integrity in transit:** The hop-by-hop use of SIP over TLS can prevent active attackers from tampering with SDP; however, they cannot prevent tampering at the proxies that serve as TLS endpoints. The only standard mitigation to this in the SIP security suite is the use of Best Practices for Securing RTP Media Signaled with SIP (SIPBRANDY)[[28]](#footnote-28), which leverages the signature of STIR/SHAKEN over the media keying material in SDP in order to provide integrity for DTLS/SRTP key in transit. While SIPBRANDY cannot protect the IP addresses in SDP, which are often modified for NATP as described above, it can assure that any attempt to set up media with an unauthorized party will fail due to DTLS/SRTP key negotiation. Other non-standard mechanisms, like zRTP: Media Path Key Agreement for Unicast Secure RTP[[29]](#footnote-29), provide a weaker out-of-band assurance.

### SIP Header Injection

**Impact**

Header injection can lead to compromise of endpoints, discovery and information disclosure based on error messages, second order injection into databases and storage mediums, number spoofing, and SSRF (Server-Side Request Forgery) attacks among others.

**Details**

Due to the development of the SIP across multiple platforms and the complexity of parsing text-based protocols, there have been multiple instances of attacks against SIP systems using header injection, either in proprietary or open source systems. Examples include:

* <http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2010-0580>
* <http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2010-0581>
* <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-17850>

**Mitigation Options**

* Most importantly, maintain all SIP elements with the latest patches from vendors and monitor bug reporting sites where new execution vulnerabilities are tabulated for the implementations in the field.
* **Testing of SIP protocol and headers:** Extensive testing of SIP headers should be performed to ensure that edge and core nodes are resilient to malicious inputs and parameters that are required are verified.
* **Use of B2BUA “normalization”:** The use of a B2BUA on the edge node will allow strict filtering rules for an incoming call. This means that the SBC can have a full allow list of the SIP Header parameters that are permitted to propagate and rewrite the message only with the correct required header values.The SBC or equivalent edge node should attempt to filter and clean parameters before passing the message to downstream systems, which in turn parse and process the SIP parameters.
* **Use a sandbox to increase resilience in complex parsing systems:** The SIP protocol is very flexible and extensible so fully filtering with “allow” listings values in a complex environment may not always be possible. To protect sensitive and critical systems from being affected, the use of virtualization and test parsing should be adopted. This means automating the creation of a virtual environment (e.g. Docker) that would have SIP parsing capabilities and a read-only file system. The message can be parsed and verified in the virtual environment before being passed to core systems. If the message is malicious or causes damage to systems, the virtual environment can be shut down and re-started without damaging core infrastructure.

### SDP Injection

**Impact**

The Session Description Protocol (SDP) is carried within key SIP messages to facilitate the negotiation of a mutually acceptable set of capabilities and parameters for conveying media between endpoints. Attackers can attempt to insert SDP wholesale or tamper with parameters of SDP in order to launch a variety of attacks.

**Details**

SDP Injection can occur in SIP INVITE requests, 200 OK responses, and various provisional media responses (e.g. 183) or mid-dialog requests (e.g. UPDATE). SDP is another text-based protocol and suffers from similar complex parsing problems to HTTP and thus makes it vulnerable to similar injection attacks. For example, a buffer overflow in SDP: <http://www.securityfocus.com/archive/1/526123>

RFC3261 recommends the use of S/MIME to protect the bodies of SIP messages. Unfortunately, S/MIME has seen virtually no deployment uptake, largely in part because so many SBC functions involve manipulation of SDP in order to enforce operator policies. As such, complete end-to-end integrity of SDP is mostly a lost cause.

**Mitigation Options**

* Most importantly, maintain all SIP elements with the latest patches from vendors and monitor bug reporting sites where new execution vulnerabilities are tabulated for the implementations in the field.
* **Testing of SDP protocol and headers:** Extensive testing of SDP parameters should be performed to ensure that edge and core nodes are resilient to malicious inputs and parameters that are required are verified.
* **Use of cryptographic integrity in transit:** The use of hop-by-hop SIP over TLS will prevent intermediate entities tampering with SDP headers other than the TLS endpoints themselves (e.g. proxies or SBCs). SIP over TCP and TLS will both partially resolve this problem.
* **Use of B2BUA:** The use of a B2BUA on the edge nodes will allow strict filtering rules for an incoming call. This means that the SBC can have a full allow list of the SDP parameters and re-write the packet, as well as perform filtering on parameters that are required to be part of the message.
* **Use of virtualization to increase resilience in complex parsing systems:** Similar to SIP header injection attacks, containerized or virtualized instances could be utilized to sanitize and segregate inputs for validation prior to disseminating the data to other core systems.

### RTP / RTCP Injection

**Impact**

RTP / RTCP injection can lead to tampering with voice streams, injecting malicious voice streams into conversations, creating “noise” to disrupt calls, cause DoS conditions, and attack network endpoints that process RTP traffic.

**Details**

RTP injection can be done in MITM scenarios where SRTP is not employed.

**Mitigation Options**

* **Encrypt RTP packets:** The user of DTLS/SRTP or zRTP can be used as way to mitigate injection of RTP packets that are in transit. Since keys are exchanged in the SDP headers during the INVITE message, proper implementation of this mitigation should include the encryption of the SIP message during transport to ensure their integrity (per RFC 8862). The use of standard cryptographic mechanisms, including TLS hop-by-hop connections between servers, will prevent eavesdroppers from learning credentials

### RTP Bleed

**Impact**

Similar to RTP injection attacks, the ability to inject RTP traffic into RTP symmetric proxies allows the extraction of legitimate RTP user traffic without requiring the in-path position of an MITM attacker.

**Details**

RTP Bleed has been thoroughly described with mitigations in the following document: <https://www.rtpbleed.com/>

**Mitigation Options**

* **Encrypt RTP packets:** The user of DTLS/SRTP or ZRTP can be used as way to mitigate injection of RTP packets that are in transit. Since keys are exchanged in the SDP headers during the INVITE message, proper implementation of this mitigation should include the encryption of the SIP message during transport to ensure their integrity (per RFC 8862).
* **Disable Symmetric RTP:** Symmetric RTP can be used to initiate RTP bleed attacks and should be disabled across the environment, if it is not required.

## Repudiation

Repudiation refers to the ability to deny that an action or an event has occurred, or acceptance/denial of responsibility. This category of STRIDE is concerned with non-repudiation.

### Toll Avoidance

**Impact**

Attackers on a commercial SIP service attempt to place calls without having an account on the system, when they have compromised another account, or otherwise prevent the system from correctly billing the originator of the call.

**Details**

There are numerous ways that an attacker can attempt to make unauthorized use of a commercial SIP service. Attackers may simply spoof or compromise another account, if the attacker can take control of the endpoint and there are not sufficient security mechanisms in the deployment to prevent the attacker from impersonating another user. Attackers may also compromise endpoints belonging to legitimate users. Even attackers outside a closed commercial network may attempt to spoof traffic to gateways or border elements in order place fraudulent calls.

**Mitigation Options**

* **Account protections:** Baseline RFC3261 recommends the use of Digest Authentication (borrowed from HTTP) for account management. Some legacy services may however still use HTTP Basic authentication, which was permitted in older versions of SIP. A number of more secure mechanisms are however used in deployments or at least specified today, 3GPP AKA solutions[[30]](#footnote-30), SAML[[31]](#footnote-31), OATH, and OpenID[[32]](#footnote-32).
* **User and Service Endpoint hardening and control:** Ensure endpoints have the latest patches deployed, have undergone testing, and any unnecessary services or open ports have been disabled. Consider deploying SIP over TLS and enforce strong authentication mechanisms on all registrars.
* **Protection of traffic in transit and SIP routing:** The use of standard SIP confidentiality and integrity mechanisms, including TLS hop-by-hop connections between UAs and operator servers, will prevent eavesdroppers from learning credentials. For SIP REGISTER in particular, forming a direct TLS connection between the UAC and the registrar without intermediary proxies is preferred. Integrity validation of SIP headers and ensuring encryption is enabled in transit will reduce the ability of malicious actors to change routing parameters and reduce their ability to re-route traffic.

## Information Disclosure

Information disclosure, which includes active attacks such as data breaches, is the intentional or unintentional release of private information to an untrusted environment. Information disclosure can occur while data is in transit or at rest or being accessed/manipulated by a process. This category of STRIDE is concerned withconfidentiality.

### Disclosure of Valid Extensions or Usernames

**Impact**

Probing for SIP responses can be used to understand whether a user or extension exists on the system. The difference between a 407, 403 and 401 can indicate whether a valid user exists and similarly the Supported, Accepted, and Required header field values can reveal whether it would be possible to target that user / extension in a further attack.

**Details**

An example of this would be the Asterisk username disclosure (CVE-2013-226). When authenticating via SIP with always authreject enabled, allowguest disabled, and autocreatepeer disabled, Asterisk discloses whether a user exists for INVITE, SUBSCRIBE, and REGISTER transactions in multiple ways.

**Mitigation Options**

* **Create best practices around response codes:** At the moment, response codes and header fields, that are returned during an INVITE, SUBSCRIBE, and REGISTER request, can reveal information to attackers if SIP extensions are valid or if they exist. By standardizing response codes and ensuring that information is not leaked via these vectors before a user has authenticated correctly or selected the correct extension would make credential stuffing and brute force attacks much harder to execute. Practically speaking, however, SIP extensibility broadly relies on Supported/Accepted/Required semantics for capability negotiation and backwards compatibility, and these features necessarily reveal to communications endpoints which extensions are in use.
* Confidentiality for SIP messages can prevent eavesdroppers (non-parties to communication) from learning these feature sets.

### Fingerprinting Internal Infrastructure from Via Headers

**Impact**

Leak of internal or private IP address, which can result in more targeted attacks such as SSRF (Server-Side Request Forgery) using edge nodes such as proxies and redirect servers.

**Details**

If not using a B2BUA on edge nodes (as with an SBC) or filtering out Via header field values, it is possible for internal network elements such as proxies and SIP servers to leak information about their internal network IP addresses as they add Via headers when they interact with SIP traffic. Similar concerns exist around the Route and Request-Route header field values, as well as Contact, and potentially even the To and From, when these contain URLs that identify nodes that should not be revealed outside a closed network.

**Mitigation Options**

* **Use B2BUA at the edge:** The use of B2BUA at the edge allows not only filtering into the network, but also the ability to filter and remove all Via headers leaving the network. This would mean that from an external perspective the attacker would only see a single edge node and not be able to map topology of internal networks. Fundamentally, however, the security of using internal network IP addresses is predicated on those addresses not being routable, and thus not available to outside attackers.

### Fingerprinting SIP Servers via Banner

**Impact**

Similar fingerprinting vulnerabilities can be seen in other protocols, such as Telnet / SSH banner grabbing or web server fingerprinting. Fingerprinting banners in SIP packets allows attackers to verify what end points are running the SIP services.

**Details**

When attackers are performing the recon phase of the attack chain, they will look for information disclosure to help them pinpoint the attack vectors to use and if there are known exploits for specific endpoints. By default, SIP reveals the SIP Server version and potentially User Agent as well. An example of a SIP response revealing information and versioning numbers can be seen below:

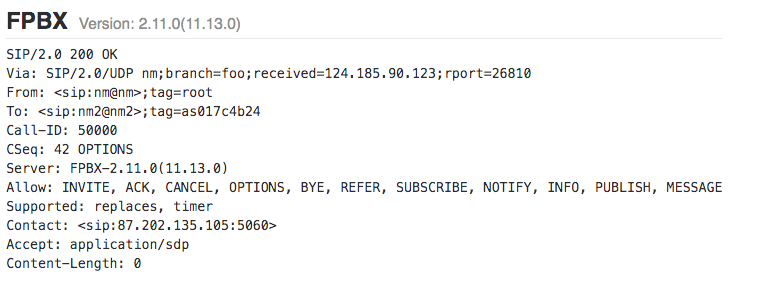


Figure - Fingerprinting Internal Infrastructure from Via Headers

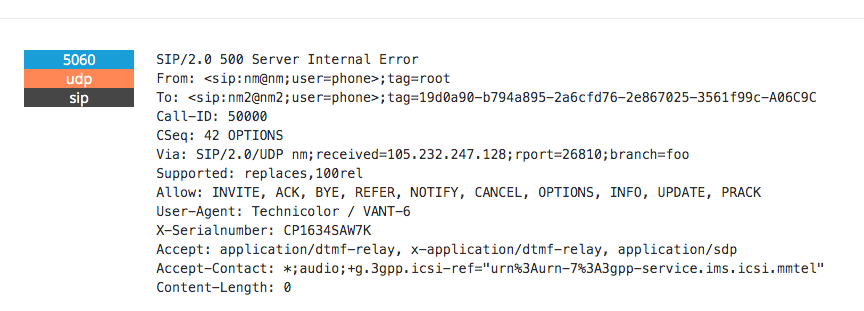
This can usually be acquired by performing active scans of endpoints, which do not require credentials or even valid SIP packets. Even 500 Error responses can reveal similar information: 

Figure - Fingerprint Internal Infrastructure from Via Headers

**Mitigation Options**

* **Access Controls:** When capable, SIP transactions should not be accepted from unknown entities, e.g. endpoints that do not require credentials or valid SIP packets.
* **Filter Headers:** Filtering the Server header and any related proprietary headers (like the “X-SerialNumber” above) in responses will stop the ability of malicious users in fingerprinting systems and taking advantage of known issues or vulnerabilities, as well as make it more difficult to target certain types of builds or environments

### Offline Brute Force Attack Using Captured Handshake

**Impact**

The attacker gains access to credentials by performing an offline brute force dictionary attack against a captured handshake. These credentials can then allow them to carry out spoofing attacks, fraud-based attacks, interception-based attacks, and normally would permit them to gain access to very sensitive information (Critical PII) of the victim.

**Details**

For this to be successful, the attackers must capture a full handshake. In case of Digest Authentication, the following parameters would need to be captured:

* (Optional) The initial INVITE request,
* The 401 response with the nonce and other parameters
* The resent INVITE request with the HMAC
* (Optional) The response confirming that the resent INVITE with the HMAC was correct

Once the above information is captured, it is possible to perform a brute force attack against the hash using a dictionary. Even though the nonce provides a “salt”, the lack of multi-round hashing and the usage prevalence of the MD5 algorithm means these brute force attacks are very often successful.

**Mitigation Options**

* **Encryption in transit:** By implementing encryption in transit, it will make it more difficult for attackers to capture a full SIP handshake and get access to the parameters that are required to run an offline brute-force attack against SIP Digest Authentication.
* **Standardize Authentication and Key Agreement (AKA) or other strong authentication:** A standard AKA authentication model for SIP is currently used by IMS (IP Multimedia Subsystems) based on [RFC3310.](https://www.ietf.org/rfc/rfc3310.txt)However**,** AKA is not very widely adopted outside of IMS for SIP, and removing barriers to entry in adopting this standard could greatly improve the security posture of SIP authentication handshakes. Other candidates proposed for use with SIP include Security Assertion and Markup Language (SAML), Open Standard Authorization (OAuth), Open Standard and Decentralized Authentication (OpenID), and similar single-sign on (SSO) technologies.

### Online Brute Force Attack Against Authentication

**Impact**

Using credential stuffing or dictionary-based online brute force attacks, malicious users can gain access to private SIP devices and PBX infrastructure by correctly guessing SIP credentials or shared secrets. This incurs similar impact as “Offline Brute Force attack using captured handshake” as described above.

**Details**

Attackers need to identify a SIP endpoint and are then free to perform online brute-force based attacks using a known dictionary and normally with known extension or usernames.

**Mitigation Options**

* **Standardize AKA:** Create a standard AKA authentication model for SIP that is currently used IMS (IP Multimedia Subsystems) based on: <https://www.ietf.org/rfc/rfc3310.txt>**.** AKA is not very widely adopted outside of IMS for SIP and removing barriers to entry in adopting this standard could greatly improve the security posture of SIP authentication handshakes.
* **Introduce best practice online brute-force protection:** By mandating the use of brute force protection mechanisms that analyze behavioral activity (e.g., multiple authentication failures) and provisioning network, transport, or application / user level controls, such as user account suspension or rate-limiting, the likelihood of an online brute force attack being successful is greatly reduced. Operational controls and monitoring such as those outlined in CIS Controls[[33]](#footnote-33) 4 and 16, and NIST publication 800-63B[[34]](#footnote-34) provide recommendations related to detecting and mitigating these attacks.
* **IP allow lists:** For critical infrastructure systems consider adding IP allow access control lists with implicit deny policies as well as encryption in transit. IP allow listing would stop access to SIP endpoints by unauthorized sources and adding encryption in transit would make the spoofing UDP/IP packets much more difficult.

### Disclosure of Geolocation

**Impact**

Subscriber geolocation is exposed in the INVITE message and in the response messages (1XX provisional responses or 200 OK). This can lead to a leak of subscriber location data to malicious parties. The carriage of subscriber geolocation is especially common in Voice over Long Term Evolution (VoLTE) networks, and in emergency services use cases. Geolocation information can be passed in a SIP header for 9-1-1- calls, which could potentially be a leak of PII; however, there is explicit provision in law that states that privacy is given up when making a 9-1-1-call[[35]](#footnote-35).

**Details**

During call establishment in a VoLTE session, the caller or the callee may propagate VoLTE specific SIP headers that that disclose the LAC and CellID of the other party.

**Mitigation Options**

* **Sanitize SIP Headers:** When SIP traffic exits closed networks, ensure headers are sanitized with a B2BUA as described above.

## Injection

Injection attacks are a broad class of attack vectors, in which an attacker supplies untrusted input to a program. The input is processed by the system and alters the execution to benefit the attacker.

### SIP MITM

**Impact**

Some SIP deployments utilize clear text connections, especially inside closed-network internal infrastructure, and thus may allow not only tampering but interception of metadata and signaling data, as well as key parameters that may be passed in the SDP headers. This creates an opportunity for the injection of additional SIP headers by attackers, as well as the injection of damaging mid-dialog requests such as UPDATE or INFO, as well dialog-ending requests such as CANCEL or BYE, any of which can deny service or alter sessions in ways unintended by the participants.

**Details**

Lack of deployment of SIP over TCP and TLS means many endpoints from vendors and service providers continually use SIP over UDP and do not support integrity over this transport layer.

**Mitigation Options**

* **Authenticate all transactions:** Use TLS over TCP to authenticate the other party and provide integrity for signaling. Verify credentials and only allow connections from known authorized entities.
* **Encryption in Transit:** Encryption also makes it far more difficult to launch injection attacks. Consider implementing the use of encryption in transit either hop-by-hop TLS or IPSec.Since SIP is addressed and routed based on the application layer information, end-to-end encryption in transit maybe difficult to achieve. However, any traversal over public networks should utilize TLS or IPSec as a layer of protection. Moreover, end-to-end integrity is more achievable; the STIR SIPBRANDY profile supports end-to-end integrity for SRTP parameters in SDP, for example.

### RTP MITM

**Impact**

Similar to SIP, RTP usage over UDP and lack of DTLS/SRTP or zRTP support means clear text media set up with SIP can be intercepted and unauthorized traffic injected. In this case, media data will be impacted, and voice calls can be eaves-dropped. RTP data is trivial to reconstruct and many tools exists to reconstruct voice streams unless there are protections in the form of encryption for the media streams.

**Details**

The prevalence of B2BUAs in SIP network deployments necessarily creates an opportunity for MitM to alter SDP to inject middleboxes into media streams. This is used as a feature in lawful intercept and similar use cases. However, it is difficult to distinguish legitimate use cases from attacks.

Although SIP and SDP have long supported the means to negotiate SRTP, many implementations and services either do not support the feature at all and certainly few support it by default. Additionally, the long history of many SRTP keying options has balkanized implementation.

**Mitigation Options**

* **Encrypt RTP packets:** SRTP / zRTP can be used as way to mitigate injection of RTP packets that are in transit. Since keys or key fingerprints are exchanged in the SDP parameters carried in an INVITE message, those elements must also have integrity protection, and if possible, also encryption, during transport to ensure the integrity of keys is not compromised.
* **Deploy DTLS-SRTP**: Deprecate all other SRTP keying methods.

### SIP Text

**Impact**

SIP may also be used to deliver text, via SIP headers, MIME bodies and media streams using SIP, notably through the MESSAGE method. It is also possible to use SIP to set up message-as-media streams with SIP Message Session Relay Protocol (MSRP)[[36]](#footnote-36) and RTP Payload for Text Conversation (RTT)[[37]](#footnote-37). Deployment of SIP text transfer is not yet widespread, but impact may be critical especially if Personally Identifiable Information (PII) data may be compromised via message interceptions. Furthermore, lack of testing frameworks and guidelines for SIP messaging best practice leads to custom deployments and lack of consistency across environments.

**Details**

Text in SIP is accomplished with three mechanisms. RFC4103 describes Real Time Text using the T.140 protocol carried in RTP. This is character-at-a-time text and replaces TTY (Baudot tones in an audio stream). The SIP MESSAGE transaction provides a “page-mode” text message, like Short Message Service (SMS). MSRP (and similar standard and proprietary protocols) provide “Instant Messaging”, line-at-a-time conversations within a session (“session-mode”). SIP is used to establish an MSRP session. Without protection, the text may be sent “in the clear” and be subject to surveillance or modification.

**Mitigation Options**

* A number of mechanisms for protecting the integrity and confidentiality of SIP messaging exist, but they have seen little uptake. S/MIME, for example, can provide security features for SIP request bodies, including the MIME body of a SIP MESSAGE request. When SIP is used to negotiate a message-as-media stream, keys for securing that media can be negotiated with SDP, provided that sufficient integrity protections are provided for SDP itself.
* Encryption in Transit : Ensure that SIP can utilize TLS or DTLS for encryption throughout transit . As the new headers will carry user data as well as signaling data, ensuring confidentiality of user messages inside the SIP packet becomes more critical.
* If there are concerns around confidentiality of data, consider implementing encryption at the application layer on the message body itself.
* For page-mode “MESSAGE” transactions, and session mode MSRP, S/MIME can be used to protect the message end to end, but little deployment has been achieved. Use of TLS hop-by-hop can be effective if appropriate authentication is used on every hop.
* For RFC4103, DTLS-SRTP can be used to protect the text stream

**Text, letter

Description automatically generated**

Figure - SIP MESSAGE Method transmitting critical PII over the signaling channel

## Denial of Service

In the context of information systems, a denial-of-service attack (DoS attack) is a cyber-attack in which the perpetrator performs some type of disruptive activity to make a system or network resource unavailable to its intended users. This is typically achieved by flooding the targeted system or network with superfluous requests. This category of STRIDE is concerned with availability.

In the non-SIP world, we are unfortunately used to very large scale (terabit per second scale) attacks on infrastructure. We have learned to mitigate them. There are devices and services that can successfully mitigate attacks of that scale using a variety of mechanisms to separate good traffic from malicious traffic. Usually, these are deployed as services with terabits of bandwidth in so called “scrubbing centers”. Traffic is redirected from the intended target by withdrawing the actual BGP routes from the victim and introducing new routes for the attacked address range to the scrubbing centers. Skilled teams and purpose-built infrastructure are able to handle newly discovered attack patterns and can quickly fashion new filters or use the source IP address of the attacking systems as a filter. Good traffic is then routed back to the original target using some private connection such as a VPN or even direct interconnection. This has been proven to work well. Highly targeted sites don’t succumb to these attacks because the scrubbing and / or transaction servicing capacity is effective, and the bandwidth at which the scrubbing centers operate on exceeds the largest attacks.

It turns out that at least some of these devices and services can mitigate SIP attacks as effectively as attacks on other protocols and are successfully providing mitigation to SIP networks today. The services (which may be “in-house”) have trained staff who can effectively use the detection and mitigation capabilities on new attack patterns. The cost for these services tends to be modest compared with the value they provide, although it is an arms race in an asymmetry battle with attackers.

There is a challenge with using these devices and services however: most SIP networks aren’t built to allow traffic to be re-routed to the scrubbing centers by manipulating BGP. They make simplifying assumptions that all the traffic is on network or goes through well-defined border elements (e.g. session border controllers). As a result, they are often not able to take advantage of these devices and services. Often, system managers point to the fact that there are no reported very high-volume SIP attacks as evidence that deploying such devices and services is not needed.

### Denial of Service

**Impact**

DoS scenarios can range from degradation of service all the way to full services being down, users unable to establish connections, and in worst case scenarios not being able to reach out to emergency services.

**Details**

SIP DoS is well described in three separate ways in the publication “Survey of network security systems to counter SIP-based denial-of-service attacks”[[38]](#footnote-38).

**SIP message payload tampering:** The ﬁrst class of attacks is based on tampering with the actual SIP message or more speciﬁcally, the SIP payload. SIP is a text-based protocol and messages are transported usually in clear text. Attackers can try to inject harmful content into a message, e.g. by entering meaningless or wrong information with the goal of exploiting a buffer overﬂow at the target and causing a denial of service condition due to a crash. Also, such messages can be used to probe for vulnerabilities in the target. Harmful code that will be executed in an unforeseen context can be introduced into the payload. An example is SQL code injection, which allows the attacker to execute SQL code within a database.

**SIP message ﬂow tampering:** A special case of DoS attacks in real time communication networks are attacks that disturb the ongoing communication between users. Common internet services like web browsing or email communication have an asynchronous time model (i.e., a requested web page is directly delivered to a user). The user will read it without further communication to the web server. The same applies to email – a user downloads the email and studies it independently of a server connection. In contrast, in SIP real time communication networks two communicating users establish a constant connection with each other whereby content is transmitted continuously between both parties. An attacker can now target this connection by introducing fake signaling messages into the communication channel. Several different SIP signaling messages can be misused for this task. A BYE message with the right credentials can prematurely terminate a session.

**SIP message ﬂooding:** When talking about a DoS attack, one generally means volumetric or ﬂooding attacks that overwhelm a victim’s resources. There are three main resources that can be targeted in a SIP ﬂooding attack: bandwidth, CPU, or memory.

**Mitigation Options**

* SIP DoS mitigations can be split into four categories:
  + Harden devices, and keep them up to date with patches so tampering that causes failures are not successful.
  + Secure messaging using TLS so attackers cannot tamper or inject malformed or malicious content.
  + Enable attack pattern recognition on middle boxes such as SBCs that recognize attacks and block them. This is only effective if the pipe size and the mitigation capacity of the pattern recognizer is greater than the attack size that reaches the device.
  + Use high volume TDoS mitigation services, which requires VoIP networks to be architected to be able to use them
* VoIP networks should be architected to be able to take advantage of high-volume DDoS mitigation capabilities and they should be widely accessible, even if developed inhouse.

### Denial of Service Using SIP Based Numbers against Critical Infrastructure

**Impact**

The impact / target would be to take down critical infrastructure services and stop operations by exhausting resources and continuously keeping systems busy.

**Details**

It is trivial to purchase large amounts of virtual numbers and setup automation for dialing, interactive voice response (IVR), and even automated responses. This combined with caller ID spoofing and UDP packet spoofing can lead to a very successful DoS attack against targets, such as switch boards and other critical infrastructure services.

**Mitigation Options**

A defense-in-depth approach should be taken to ensure a combination of mitigation

options are deployed to lessen the impact of this threat.

The following mitigation options should be considered in addition to the mitigations mentioned in Section 5.6.1:

* Utilize number verification services to ensure number blocks are dialing virtual numbers that have been recently purchased or belong to a single entity. Other checks like number age, last use, last SIM swap, and reputation can be used as factors in deciding to block or ban certain number ranges from carrying out incoming calls.
* Consider deploying SIP over TCP or using DTLS to ensure integrity and confidentiality of SIP messages. This prevents number spoofing and forces attackers, who want to perform large scale DoS attacks to buy numbers instead of spoofing their origin, which in turn raises the cost of executing such an attack.
* Use integrity verification techniques, such as STIR/SHAKEN, to ensure origin caller ID as well as time of call are verified before trunks or circuits are opened. This prevents replay attacks and large-scale DoS attacks.
* Have systems automatically disallow automated application to person (A2P) messaging during off-hours. This reduces the time window malicious actors have to utilize A2P infrastructure. This also allows organizations to prioritize their spending in terms of defensive systems and traffic analysis, if automatic blocking occurs during peak attack hours.

### SIP REGISTER / INVITE Flooding

**Impact**

SIP server can no longer establish or maintain connections due to resource consumption. This can lead to subscribers not being able to establish connectivity to the network or initiate or receive calls.

**Details**

A SIP Register flood consists of sending a high volume of SIP REGISTER or INVITE packets to SIP servers (indifferently accepting endpoint requests as first step of an authentication process), therefore exhausting their bandwidth and resource. Such floods are trivial to carry out and as SIP runs over UDP in a lot of instances, creating such floods with spoofed IP addresses allows attackers to greatly increase the effectiveness of the DoS attack as well as evade IP deny listing counter measures (e.g., Fail2Ban), which allows to setup rate-limiting controls on SIP measures that take into account originating IP and other indicators.

**Mitigation Options**

* See mitigations in Section 5.6.1

### Silent INVITE Denial of Service

**Impact**

DoS of one targeted subscriber from receiving calls.

**Details**

A SIP INVITE flood interlaced with a BYE /CANCEL flood before the end device can initiate respond or initiate ringing, would result in a DoS of that subscriber as other calls would register the device as busy.

**Mitigation Options**

* See all mitigations in Section 5.6.1
* Introduce rate-limiting mechanisms that block the same number initiating too many consecutive INIVITE / BYE / CANCEL requests.
* Introduce control flow to stop continuous use of BYE or CANEL requests before a 100 RINGING response is issued back to the calling entity to ensure silent DoS cannot be carried out.

### SIP UDP: Amplification Attacks and Spoofing

**Impact**

Victim’s resource and SIP architecture is used in a UDP amplification attack to take down another system. This may result in blocking of legitimate traffic, complaints from organizations or even legal action.

**Details**

A Distributed Reflective Denial of Service (DRDoS) attack is a form of Distributed Denial of Service (DDoS) that relies on the use of publicly accessible UDP servers, as well as bandwidth amplification factors, to overwhelm a victim system with UDP traffic. An attacker sends a single SIP message to the victim, with the spoofed IP address of the target (second victim) to be attacked. The victim’s system sends a SIP 401 or 403 packet to the target expecting a response with a new INVITE, however the target obviously does not respond, with the victim potentially sending dozens more of these packets for a single spoofed request.

**Mitigation Options**

* **Restricting access to SIP systems:** Consider restricting access to SIP systems through a number of techniques including:
  + IP allow listing (with implicit deny policies) to restrict to known proxies
  + Ensuring proper filtering mechanism on incoming traffic
  + Reducing SIP response numbers
  + Implementing SIP on connection-oriented transport protocols such as TCP / SCTP to ensure proper authentication

## Elevation of Privilege

Privilege escalation is the act of exploiting a vulnerability, design flaw, or a misconfiguration of a system to gain elevated access to resources that are normally protected. This attack typically results in the perpetrator being able to perform unauthorized actions. This category of STRIDE is concerned with authorization.

### SIP Registration Hijacking

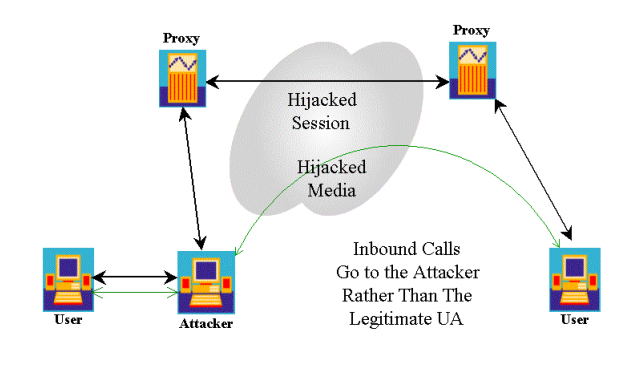


Figure - SIP Registration Hijacking

**Impact**

A malicious user registers against a SIP registrar, allowing them to carry out and initiate calls / access voicemail as a trusted user. This can lead to fraud, metadata and media data interception and spoofing.

**Details**

If the SIP registrar allows registration without authentication or if authentication can be bypassed, then a malicious user may execute SIP registration attacks.

**Mitigation Options**

* When feasible, ensure strict authentication is in place on top of IP allow lists.
* Use best practice password hygiene and user lock outs.

## Multi-Impact Attacks

Not all attacks squarely align within a single category of the STRIDE framework. The following security threats align with multiple categories.

### SIP Header and SDP Tunneling

**Impact**

Tunneling data in SIP packets can lead to loss of revenue and fraud amongst carriers as user’s and endpoints might be establishing out of band connections. If SIP headers and SDP parameters are not sanitized or allow listed, data tunneling can occur between two endpoints.

**Details**

Injecting custom SIP headers or custom SDP parameters into the INVITE message would allow data to be tunneled in these headers directly to the end user’s device as the INVITE message will propagate between devices.

**Mitigation Options**

* **Implement filtering at the edge / B2BUA:** Having strict filtering on the edge nodes at ingress and egress points mean that custom headers are unlikely to propagate. Header size and length restrictions may also be a valuable tool in tunneling and tampering attacks alike, as well as protocol anomaly detection to identify these attempted attacks.

### SIP Route Steering

**Impact**

Malicious proxies can route and intercept SIP traffic and call metadata via injection and tampering with the “Record-Route” and “Route” headers. This could lead to full call interception and meta data interception.

**Details**

Adding a “Record-Route” header or tampering with or adding a “Route Header” would allow a malicious proxy to redirect signaling traffic

**Mitigation Options**

* Introduce cryptographic controls to the transport layer through the use of DTLS or TLS if using TCP as a transport layer.
* Use authentication and avoid sending SIP requests to untrusted intermediaries.

### SIP Registering vs Non-Registering User Agents

**Impact**

Lack of registration requirements with authenticated endpoints for UAs (User Agents) can lead to fraud-based attacks and attacks of information disclosure due to voice mail compromise and caller ID spoofing.

**Details**

When a SIP UA connects to a network, it may require authentication and authorization to ensure that further signaling messages such as INVITE can be sent and that correct billing and restrictions are applied. Not all cases require registrations and there is a distinct lack of guidance and best practice to protect systems and organizations when it comes to non-registering UAs.

**Mitigation Options**

* Controls should be introduced to ensure that non-registering user agents have some strict authentication controls either through IP allow lists, private connections, or other mechanisms.
* Consider adding authentication criteria per message if registration cannot be supported.

### SIP Conferencing Information Disclosure

**Impact**

RTP stream mixing can lead to interception and RTP streams and injections. Furthermore, SIP signaling interception during conferencing can leak metadata to conference attendees and to conference users. This may reveal private information such as user agent, IPs, and even location information.

**Details**

SIP conferencing links a number of SIP users together and normally mixes RTP streams to ensure that communication is maintained between all users.

**Mitigation Options**

* Ensure good multi-tenancy controls are deployed and mixing testing has been thoroughly carried out against injection into audio stream mixing and MITM.
* Employ SRTP or ZRTP to encrypt audio streams and protect them from injection and interception.
* Remove and scrub any metadata in SIP headers at the SBC that is not strictly required to propagate the connection of the SIP channel. Cognizant of emergency communications requirements, consider carefully masking parameters such as geolocation, user agent, via headers, proxy headers, and any other metadata that may not be required for the conference call to properly connect.

# Findings, Gap Analysis, and Recommendations

## Root Cause Findings

The table below collects the various SIP security related issues identified in this document and describes their underlying root causes. Many of the issues share common root causes, which will be further discussed in the recommendation section.

|  |  |
| --- | --- |
| Issue | Root Cause |
| Spoofing Caller IDs: | * Implementation of caller ID verification * Lack of protocol verification or design to verify caller ID * Lack of TLS / IPSec implementation |
| Fingerprinting SIP Servers via Banner | * SIP Server header present by design * Lack of TLS / IPSec implementation * Lack of problem awareness * Information disclosure may prove useful for debugging |
| Offline Brute force attack | * Lack of TLS / IPSec implementation * Protocol authentication design and lack of upgrading (e.g., downgrade attacks) * Dated cryptographic algorithm use |
| Online Brute force attacks | * Lack of TLS / IPSec implementation * Protocol design and implementation over UDP * Lack of state or session management to force attackers to perform more difficult calculations |
| Denial of Service | * Lack of TLS / IPSec implementation * Issues in transport architecture * Lack of NAT handling * Implementation of stateless proxies and architecture * Easily spoofed packets and connectionless transport protocols * Time consuming operations for call establishment |
| Disclosure of valid extensions | * Extensibility by design at the protocol level * Timing attacks due to time consuming crypto operations can lead to disclosure of valid extensions |
| SIP Register / Invite flood | * Time consuming operations for call establishment * Verification are not done out of band, allowing for ease of resource consumption * Trusted networks with bad actors |
| Silent DoS | * Lack of TLS / IPSec implementation * Lack of application control flow allows timing abuse to occur. * Device engagement and the device initiating a ring become mismatched and can be abused by attackers. |
| SIP Data Tunneling | * Protocol design with user-set fields suitable for obfuscating data. |
| SDP Injection / SIP Header injection | * Lack of TLS / IPSec implementation * Lack of header integrity checks * Lack of SDP integrity checks * Text-based protocol makes it very difficult to build reliable bug free parsers |
| Insecure SIP Routing Proxies | * Protocol design allows header tampering by untrusted parties. |
| Reliance on “Walled Gardens” | * Regulatory entities introduced isolation and segregation requirements for critical infrastructure systems. While on the surface this does reduce attacks and exposure, the prolonged reliance on isolated environments slows down the hardening process of critical protocols. Lack of testing and difficult access also allowed the introduction and propagation of critical issues that were never addressed, as well flawed design principles. * Private IP address space. |
| Lack of standard “fuzz testing” regimen | * Current tooling for testing the security posture of SIP systems is outdated and is poorly maintained. Introducing new standards and mechanisms into testing SIP headers and SDP headers would be beneficial to vendors and open source systems to highlight gaps and flaws in parsing and business logic. |

Table - Root Cause Findings

## Gap Analysis

Open standards intended for use on the public Internet face difficult security choices – some of which may not admit of practical solutions. Every SIP deployment faces tensions about where and how security is implemented, and what exactly it will secure. Sometimes the security incentives of operators and end users may not be completely aligned.

SIP is most fundamentally a capability negotiation protocol. It allows endpoint UAs to connect to one another through a rendezvous service and exchange capability information about a sessions they can share. That process always involves a comparison of capabilities at both the header level (use of the SIP Required, Allowed, and Support headers) and at the body level (comparison of SDP support for media protocols and their characteristics). The intention of these mechanisms is to arrive at a highest-common denominator of features supported by both endpoints. However, protocols with this overall architecture will always be susceptible to downgrade attacks, especially when it comes to security features. Also, revealing feature sets and capabilities to a remote endpoints (or intermediary networks) always increases the risk that it will be possible to fingerprint and track end users. Negotiation is not something that can be simply removed, however – it is core to the purpose of SIP. It is unclear how concerns about exploiting the negotiation system could be effectively mitigated.

While the SIP tool suite provides potential means of securing media end-to-end (for example, with RFC8862), these tools see little practice use, as a number of operational and regulatory constraints discourage operators from allowing end-to-end media confidentiality (including CALEA requirements, financial services reporting requirements, and so on). In practice, this means that much of the media negotiated by SIP is sent in the clear, and only network-layer security practices of operators prevent that media from being captured by adversaries. End users have no visibility into whether their conversations are private or not.

Similarly, the needs for SIP protocol normalization and similar policies enforced at operator network edges preclude many integrity mechanisms that might be used to secure signaling as well. The integrity over call setup messages provided by STIR/SHAKEN has a very limited scope, effectively protecting just the called and calling party numbers. Even seemingly trivial alterations like NAT/NAPT translation of IP addresses in the SDP body make it impossible to provide integrity protection for one of the most security-sensitive things that SDP can convey: the IP address to which media should be sent. It is impossible for the endpoints negotiating a SIP session to distinguish changes legitimately made to SIP messages by operators from tampering introduced by attackers. This is a fundamental difficulty resulting from the tension between operator requirements and the imperatives of security which may not admit of any easy or practical resolution.

Nuisance calling remains one of the most publicly visible security defects in the telephone network, and the lack of accountability for it is primarily a result of calling party number spoofing.. The STIR/SHAKEN work, and surrounding regulatory framework, promises to prevent spoofing within carrier network-to-network interfaces, provided those interfaces are SIP. However, not all telephone calls pass between networks over SIP, and some networks may use legacy SBCs that would not support STIR/SHAKEN. Moreover, a whole set of actors other than carriers use SIP, including enterprises and individual users, who will not be eligible to receive SHAKEN certification under the existing policy framework. And lastly, STIR/SHAKEN deployment in North America is far ahead of the remainder of the world, and it remains to be seen if and how international calls will embrace STIR/SHAKEN. While the policy and technology around all of this is still evolving, there are currently significant gaps in the picture.

This highlights another difficulty that SIP faces: it often interworks with other protocols and environments. A call that begins over SIP might ultimately terminate on the PSTN after transiting a SIP-SS7 gateway. SIP may also interwork with other VoIP systems on the Internet. Ultimately, SIP’s security features, even when properly implemented and deployed, may not translate into other operating environments.

Finally, we recognize that a significant installed base of legacy SIP stacks curtails any improvements that have been made to SIP since the early 2000s. Although sending SIP over UDP is widely known to cause problems, it is unclear when or how we could effectively transition away from UDP, as there are significant difficulties with upgrading or replacing legacy SIP devices.

The group analysis suggests that there is an opportunity for ongoing security work in a number of areas. Today, we are aware of work ongoing in a number of forums to improve SIP’s security, to include.

IETF

* Change control of the SIP protocol remains in in the IETF.
* Most on-going work is focused on STIR
  + [draft-peterson-stir-servprovider-oob-01](https://datatracker.ietf.org/doc/draft-peterson-stir-servprovider-oob/)
  + [draft-ietf-stir-rph-emergency-services-03](https://datatracker.ietf.org/doc/draft-ietf-stir-rph-emergency-services/)
  + [draft-ietf-stir-passport-divert-09](https://datatracker.ietf.org/doc/draft-ietf-stir-passport-divert/) (Now RFC8946)

|  |  |
| --- | --- |
|  | * + [draft-ietf-stir-oob-07](https://datatracker.ietf.org/doc/draft-ietf-stir-oob/) (Now RFC8816) |

* + [draft-ietf-stir-cert-delegation-03](https://datatracker.ietf.org/doc/draft-ietf-stir-cert-delegation/)
  + [draft-ietf-stir-passport-rcd-09](https://urldefense.com/v3/__https:/tools.ietf.org/html/draft-ietf-stir-passport-rcd-09__;!!N14HnBHF!sH8zZyL_X-Qc2e5nbtlGK8Ix-ITaCHN60_E_qDTjFrvGlWdAyLJvMY7OwETzopA$)
  + [draft-ietf-sipcore-callinfo-rcd-01](https://urldefense.com/v3/__https:/tools.ietf.org/html/draft-ietf-sipcore-callinfo-rcd-01__;!!N14HnBHF!sH8zZyL_X-Qc2e5nbtlGK8Ix-ITaCHN60_E_qDTjFrvGlWdAyLJvMY7OnSf3OLY$)
  + [draft-peterson-stir-rfc4916-update-02](https://urldefense.com/v3/__https:/tools.ietf.org/html/draft-peterson-stir-rfc4916-update-02__;!!N14HnBHF!sH8zZyL_X-Qc2e5nbtlGK8Ix-ITaCHN60_E_qDTjFrvGlWdAyLJvMY7OixvPDDE$)
  + [draft-peterson-stir-messaging-00](https://urldefense.com/v3/__https:/tools.ietf.org/html/draft-peterson-stir-messaging-00__;!!N14HnBHF!sH8zZyL_X-Qc2e5nbtlGK8Ix-ITaCHN60_E_qDTjFrvGlWdAyLJvMY7OBZePSuw$)
* Limited “core” SIP protocol work current underway.

Organizations that Profile SIP

* ATIS IP-NNI/SIP Forum (as well as some ATIS PTSC activity)
* 3GPP
* GSMA
  + SIPSEC
  + VINES (validation of integrity end to end system confidentiality) (application of STIR/SHAKEN)

## Emergency Services / 9-1-1 Considerations

The current, E9-1-1 system is dangerously vulnerable to attack because it is based on having a very small number of Time Division Multiplexers (TDM) or Signaling System Number 7 (SS7) trunks from each service provider to a TDM switch. The number of trunks varies depending on the size of the Public Safety Answering Point (PSAP), but is always less than the number of call taker positions in the PSAP, and is calculated to provide a P.01 grade of service, meaning one call out of a hundred at peak busy hour may be dropped. For smaller PSAPs, 2 or 3 calls from one service provider network will block any legitimate calls from that service provider, and if 2 or 3 calls from each of two service providers were sent, no legitimate calls would go through until the bad calls were dismissed manually by the call taker. If a sustained attack that was sourced from several high-volume call sources (wireless networks and national cable networks, for example) to all or at least most PSAPs, the entire 9-1-1 system would be blocked. Since this is a TDM system, source authentication cannot currently be used, and since we’re already deploying the successor to E9-1-1, no investments are realistic for that system.

The Next Generation 9-1-1 system (NG9-1-1) is based on SIP and has IP network connections at both ends (service provider to NG9-1-1 system, NG9-1-1 system to PSAP). This both increases the attack surface and provides mitigation capability not available for E9-1-1. NG9-1-1 is now being deployed, but it will likely be a decade or more before it is ubiquitous, barring some fundamental change in funding.

NG9-1-1 is a fairly conventional SIP based VoIP system, and all of the attacks and mitigations covered in this document apply. Geolocation is passed in a SIP header for 9-1-1 calls which could be a leak of PII, but there are explicit provisions in law that state that privacy is given up when making a 9-1-1 call. If the geolocation header field can be manipulated, an attacker could cause resources to be directed to the wrong location, although as long as source authentication is provided, this attack is much less likely.

The 9-1-1 is a very attractive target, and it is entirely reasonable to expect all sorts of attacks. NG9-1-1 systems therefore should deploy all the countermeasures suggested in this document and aim to minimize the opportunity for downgrade attacks to nullify the value of those countermeasures.

Source authentication is a very important countermeasure for 9-1-1[[39]](#footnote-39). Stopping spoofing of telephone numbers makes some attacks much less likely, most notably “Swatting”, where a spoofed telephone number is used by a real person to claim a neighbor’s house is being invaded and the SWAT team should be deployed (or similar). Having a reliable telephone number can also help with other mitigation strategies. For example, the NG9-1-1 system allows a PSAP to mark a telephone number as a “Bad Actor” and have calls from that telephone number blocked at the entrance to the NG9-1-1 system. This mechanism works on any URI in the “From:” field, and if, for example, non-initialized devices send some unique identifier (e.g. the International Mobile station Equipment Identity (IMEI)) in the signaling, “Bad Actor” can be used to filter such calls. Note that “Bad Actor” is entirely within the NG9-1-1 system and does not rely on any service provider filtering of calls.

It is also essential that service providers sending calls to NG9-1-1 systems use TCP and not UDP to prevent IP address spoofing, because a reliable IP address can also be used in a filtering system to distinguish legitimate calls from attack calls. NG9-1-1 prefers TLS over TCP for signaling but accepts TCP without TLS and UDP.

NG9-1-1 is a very attractive target for local or nationwide volumetric TDoS. Mitigation of terabit level attacks is feasible using existing technology, but 9-1-1 call sources must be able to reroute high volumes of attack calls from within their networks to scrubbing centers using changes in BGP routes, or other appropriate diversion mechanisms. For this reason, we recommend implementation of BGP or similar reroute mechanisms, as well as high capacity scrubbing and / or transaction service capacity to mitigate large scale attacks, for all service providers sending calls to NG9-1-1.

## Recommendations

The gap analysis above aside, most of the things the industry could do that would be most impactful to improving SIP security are things we already know how to do, many of which have been long documented in SIP-related RFCs over the past two decades. The primary impediment to accomplishing them has been the difficulty in deprecating extremely common and widespread practices that are well-known to be insecure, but which the industry lacks sufficient motivation to abandon.

Based on the collective knowledge surrounding the identified SIP security issues, CSRIC VII recommends the following actions to improve the state of SIP security:

### Operators and vendors should adopt well-established security frameworks

The FCC should support SIP operators adopting and deploying well-established security frameworks. This recommendation is in alignment with CSRIC VII’s recommendations on securing 9-1-1 implementations through the use of CIS Controls and CSRIC IV’s recommendations on managing risk by using the NIST Cyber Security Framework.

Security frameworks provide a common language for understanding, managing, and expressing cybersecurity risk to internal and external stakeholders. It can be used to help identify and prioritize actions for reducing cybersecurity risk, and it is a tool for aligning policy, business, and technological approaches to managing that risk. It can be used to manage cybersecurity risk across entire organizations, or it can be focused on the delivery of critical services within an organization. Listed below are three widely adopted security frameworks.

* The US National Institute of Standards and Technology (NIST) Framework for Improving Critical Infrastructure Cybersecurity (NIST CSF)
* The Center for Internet Security Critical Security Controls (CIS)
* The International Standards Organization (ISO) frameworks ISO/IEC 27001 and 27002

These frameworks can be adapted, tailored, and deployed across the SIP ecosystem. Cybersecurity frameworks are often mandatory, or at least strongly encouraged, for companies that want to comply with state, industry and international cybersecurity regulations. CSRIC VII recommends a framework should be selected, and if possible, organizations should strive to implement the full set of controls applicable to their target profiles and environments whenever practical.

As part of the research for this report, a set of common minimal security practices were identified as actions that would significantly improve SIP security. While the lists below are tactical in nature, these general good security hygiene techniques would be contained within any security framework.

Industry should implement basic hygiene best practices to ensure that their SIP networks are secure:

* Ensure software and firmware is up to date.
* Create and require complex passwords. Avoid using pin codes for any authentication.
* Authenticate access with strong credentials and mandating the use of brute force protection mechanisms.
* Develop a security posture towards peers grounded in reputation history and commercial contracts.
* Understand and measure the characteristics of your signaling and media and prepare for DDoS and TDoS events.
* Require transport or network layer security for internetwork traffic.

Some additional best practices industry should implement are listed below:

* Use of cryptographic operations to ensure confidentiality and integrity.
* Additional data sanitization, filtering, and validation throughout the SIP network.
* Ensure testing is continuously performed against all endpoints that are publicly exposed.

This recommendation addresses the CSRIC VII charge to provide macro-level guidance as to how SIP security best practices can be achieved by leveraging well-established security frameworks.

### STIR/SHAKEN and End-to-End Security

The American telecommunications industry is currently deploying a major new security feature, the STIR/SHAKEN framework for mitigating the spoofing of calling party numbers. Once this capability is widely available, it will provide a building block for the implementation of new approaches to fraud prevention and similar security issues facing operators. There are however three factors that are possible roadblocks to the realization of STIR/SHAKEN’s potential:

* Lack of access to SHAKEN credentials by non-carrier or nontraditional entities (such a SIP-enabled enterprises)
* Difficulty interfacing North American SHAKEN with international operators outside its governance model
* Failure to leverage STIR/SHAKEN for a full suite (RFC8862) approach to securing signaling and media

Although the FCC has already amended the eligibility rules for access to SHAKEN credentials, they still focus on entities who are eligible to receive Operating Carrier Numbers (OCNs), a traditional North American telephone network identifier. The world of SIP telephony, however, is rapidly evolving, becoming more connected internationally and more dependent on unregulated entities who originate SIP traffic. In order for the full benefits of STIR/SHAKEN to be realized, we recommend further study of the eligibility of entities that use telephone numbers, and SIP, but will never possess an OCN. The core STIR standards have long supported credential assignments for telephone number ranges, for example. The commission would do well to encourage further study of non-OCN-based models for building STIR/SHAKEN trust to accommodate non-American and non-carrier entities.

The lack of end-to-end media security for SIP calls is, bluntly, a situation that would never be tolerated in e-commerce applications or similar consumer-facing Internet services. Due to the highly mediated nature of deployed SIP networks, American consumers and enterprises simply have no expectation of privacy when placing telephone calls, which poses risks not just to individual privacy, but also national security risks as foreign adversaries seek to exploit these vulnerabilities to spy on the activities of corporations and citizens. CSRIC VII urges the commission to study the potential trade-offs between caller privacy and law enforcement requirements to find a balance point that increases confidence in the confidentiality of calls.

### Further Study of SIP Primitives that Inhibit Better Security

Because it is very common for SIP sessions to traverse multiple SIP proxies and operators, there is a growing need and desire to guarantee end-to-end security for SIP traffic. In order to enable this end-to-end security, operational primitives that inhibit end-to-end security must be identified and further studied. CSRIC VII believes that one of those primitives is the UDP protocol used throughout SIP deployments. To that end, CSRIC VII recommends the FCC to further study if and or when downgrades from more secure protocols, such as TCP with TLS, should be allowed in SIP. It should be noted that any intermediate system which needs to parse the application layer of the SIP packet, either the SIP headers or the SDP to make routing decisions, will need to have TLS offloading capabilities to be able to decrypt and read addressing information.

As thoroughly described throughout this document, insecure UDP-based SIP signaling is a leading root cause of numerous security vulnerabilities found in SIP-based services. As an alternative to the connectionless orientation of UDP, TCP provides a connection-oriented reliable, ordered, and error checked delivery of packet streams between supported SIP endpoints. Moving SIP away from UDP to TCP will improve the call signaling function and result in fewer call drops, reduced one-way audio issues, generally be able to better withstand packet loss, and prevent many of the SIP vulnerabilities identified in this document.

Similarly, the use of Basic authentication is far beyond its end-of-life, and today Digest Authentication does not provide adequate security against offline dictionary attacks. Numerous alternatives that are widely used in e-commerce are available, as detailed above, and their operation in the SIP context has long been studied. Encouraging a move away from these legacy technologies, and similar weak approaches like IP-address access control lists, would yield immediate benefits in securing this critical infrastructure.

Finally, CSRIC VII believes one of the best ways to improve the speed and effectiveness of scrubbing and filtering is to improve features of SIP signaling that make it easier to identify sources of traffic; specifically, using federally mandated call authentication technology (e.g. STIR/SHAKEN). Marking and policing is most effective against trunk-based sources of attack which is the vector used for virtually all illegal robocalls.

# Conclusions

This report reviews the security vulnerabilities affecting SIP, how the industry is addressing these vulnerabilities, and identifies several gaps in industry action. The culmination of this work results in a series of best practices relevant to SIP. A key primitive that CSRIC VII has identified is that many of the security concerns can be remediated by basic security hygiene practices. Significant SIP security improvements would be materialized by doing these basic security practices. Furthermore, CSRIC VII believes additional work should be conducted to better understand the security benefits of a connection-oriented protocol (TCP) that can utilize cryptographic features is a desirable future endeavor.

The members of CSRIC VII, Working Group 6, thank all those who have contributed research and perspective to this CSRIC VII report and look forward to supporting continued collaboration in best practices and recommendations surrounding SIP.

# Appendix

## Acronyms and Definitions

|  |  |
| --- | --- |
| B2BUA | Back-to-back User Agent |
| CPaaS | Communications Platform as a Service |
| CSRIC | Communications Security, Reliability, and Interoperability Council |
| CVE | Common Vulnerabilities and Exposure |
| DDoS | Distributed Denial of Service |
| DNS | Domain Name System |
| DTLS | Datagram Transport Layer Security |
| HTTP | Hypertext Transfer Protocol |
| IMS | IP Multimedia System |
| IMS | IP Multimedia Subsystem |
| IP | Internet Protocol |
| IPSEC | Internet Protocol Security |
| MPLS | Multiprotocol Label Switching |
| PBX | Private Branch Exchange |
| PKI | Public key infrastructure |
| PSTN | Public Switched Telephone Network |
| RFC | Request for Comments |
| RTP | Real-time Transport Protocol |
| S/MIME | Signed / Multipurpose Internet Mail Extensions |
| SBC | Session Border Controller |
| SDP | Session Description Protocol |
| SIP | Session Initiation Protocol |
| SMTP | Simple Mail Transfer Protocol |
| SQL | Structured Query Language |
| SRTP | Secure Real-time Transport Protocol |
| SSH | Secure Shell |
| SSRF | Server-side Request Forgery |
| STIR/SHAKEN | Secure Telephony Identity Revisited / Secure Handling of Asserted information using toKENs |
| STRIDE | Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege |
| TCP | Transmission Control Protocol |
| TDoS | Telephony Denial of Service |
| TLS | Transport Layer Security |
| UAC | User Agent Client |
| UAS | User Agent Server |
| UDP | User Datagram Protocol |
| VoIP | Voice over IP |
| VoLTE | Voice over Long Term Evolution |
| ZRTP | Zimmermann Real-time Transport Protocol |

1. J. Rosenberg, et al., SIP: Session Initiation Protocol (2002), <https://www.ietf.org/rfc/rfc3261.txt>. [↑](#footnote-ref-1)
2. The Transport Layer Security (TLS) Protocol Version 1.3 (2018), <https://tools.ietf.org/html/rfc8446> [↑](#footnote-ref-2)
3. Internet Protocol (1981), <https://tools.ietf.org/html/rfc791> [↑](#footnote-ref-3)
4. Path MTU Discovery (1990), <https://tools.ietf.org/html/rfc1191> [↑](#footnote-ref-4)
5. Domain Names – Concepts and Facilities (1987), <https://tools.ietf.org/html/rfc1034> [↑](#footnote-ref-5)
6. HTTP Over TLS (2000), <https://tools.ietf.org/html/rfc2818> [↑](#footnote-ref-6)
7. Simple Mail Transfer Protocol (2008), <https://tools.ietf.org/html/rfc5321> [↑](#footnote-ref-7)
8. Multiprotocol Label Switching Architecture (2001), <https://tools.ietf.org/html/rfc3031> [↑](#footnote-ref-8)
9. IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap (2011), <https://tools.ietf.org/html/rfc6071> [↑](#footnote-ref-9)
10. The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM) (2011), https://tools.ietf.org/html/rfc6116 [↑](#footnote-ref-10)
11. SIP Forum, https://www.sipforum.org/ [↑](#footnote-ref-11)
12. Session Traversal Utilities for NAT (STUN) (2020), <https://tools.ietf.org/html/rfc8489> [↑](#footnote-ref-12)
13. Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN) (2010), <https://tools.ietf.org/html/rfc5766> [↑](#footnote-ref-13)
14. Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols (2010), <https://tools.ietf.org/html/rfc5245> [↑](#footnote-ref-14)
15. Internet X.509 Public Key Infrastructure Certificate and <https://tools.ietf.org/html/rfc5245>Certificate Revocation List (CRL) Profile (2008), <https://tools.ietf.org/html/rfc5280> [↑](#footnote-ref-15)
16. IETF STIR, <https://datatracker.ietf.org/wg/stir/documents/> [↑](#footnote-ref-16)
17. SIP: Session Initiation Protocol (2002), <https://www.ietf.org/rfc/rfc3261.txt> [↑](#footnote-ref-17)
18. Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification (2019), <https://tools.ietf.org/html/rfc8551> [↑](#footnote-ref-18)
19. The Session Initiation Protocol (SIP) Digest Access Authentication Scheme (2020), <https://tools.ietf.org/html/rfc8760> [↑](#footnote-ref-19)
20. Alliance for Telecommunications Industry Solutions, <https://www.atis.org> [↑](#footnote-ref-20)
21. SIP Forum, https://[www.sipforum.org](http://www.sipforum.org) [↑](#footnote-ref-21)
22. 3rd Generation Partnership Project, <https://www.3gpp.org/> [↑](#footnote-ref-22)
23. Global System for Mobile Communications, <https://www.gsma.com/> [↑](#footnote-ref-23)
24. STRIDE, <https://en.wikipedia.org/wiki/STRIDE_(security)> [↑](#footnote-ref-24)
25. Work in Progress: Connected Identity for STIR draft-peterson-stir-rfc4916-update-02 (2020), <https://tools.ietf.org/html/draft-peterson-stir-rfc4916-update-02> [↑](#footnote-ref-25)
26. Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP) (2010), <https://tools.ietf.org/html/rfc5764> [↑](#footnote-ref-26)
27. Communications Assistance for Law Enforcement Act, <https://www.fcc.gov/public-safety-and-homeland-security/policy-and-licensing-division/general/communications-assistance> [↑](#footnote-ref-27)
28. Best Practices for Securing RTP Media Signaled with SIP (2021), <https://tools.ietf.org/html/rfc8862> [↑](#footnote-ref-28)
29. ZRTP: Media Path Key Agreement for Unicast Secure RTP (2011), <https://tools.ietf.org/html/rfc6189> [↑](#footnote-ref-29)
30. Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA) (2002), https://tools.ietf.org/html/rfc3310 [↑](#footnote-ref-30)
31. Security Assertion Markup Language (SAML) 2.0 Profile for OAuth 2.0 Client Authentication and Authorization Grants (2015), https://tools.ietf.org/html/rfc7522 [↑](#footnote-ref-31)
32. Third-Party Token-Based Authentication and Authorization for Session Initiation Protocol (SIP)(2008), https://tools.ietf.org/html/rfc8898 [↑](#footnote-ref-32)
33. CIS Controls, <https://www.cisecurity.org/controls/> [↑](#footnote-ref-33)
34. NIST Special Publication 800-63B, <https://pages.nist.gov/800-63-3/sp800-63b.html> [↑](#footnote-ref-34)
35. To ensure compliance, CMRS providers must certify that neither they nor any third party they rely on to obtain dispatchable location information for 911 purposes will use such information for any non-911 purpose, except with prior express consent or as required by law. 47 CFR § 9.10(i)(4)(iv)-(v) paragraph 56 [↑](#footnote-ref-35)
36. The Message Session Relay Protocol (MSRP) (2007), <https://tools.ietf.org/html/rfc4975> [↑](#footnote-ref-36)
37. RTP Payload for Text Conversation (2005), <https://tools.ietf.org/html/rfc4103> [↑](#footnote-ref-37)
38. Survey of network security systems to counter SIP-based denial-of-service attacks (2010), <http://www.cs.columbia.edu/~dgen/papers/journal/Journal-08.pdf> [↑](#footnote-ref-38)
39. NENA 9-1-1 Association, <https://www.nena.org/page/NG911_Security> [↑](#footnote-ref-39)