

SS7 Attacker Heaven turns into Riot: How to make Nation-State and Intelligence Attackers' lives much harder on mobile networks

SigFW

Open Source SS7/Diameter firewall
for Antisniff, Antispoof & Threat Hunt

Martin Káčer
Philippe Langlois
P1 Security

Content

Content	2
1 Abstract	5
2 Introduction	6
2.1 Problem Statement	6
2.2 Related work	6
3 The Approach	7
3.1 SS7 firewall - Technical capabilities	7
3.2 Diameter firewall - Technical capabilities	7
4 The Current Status	9
4.1 SS7 / Sigtran stack overview	9
4.2 Perimeters of SS7 overview	9
4.3 SS7 message categories	10
4.4 SS7 screening categories grouped by protocol layers	11
4.5 Possible SS7 filtering by existing infrastructure without FW	12
4.6 Current status conclusion and acknowledgement	13
5 Advanced SS7 Attacks	14
5.1 Category 2 attack example - VLR profile manipulation	14
5.2 Category 2 attack example - GPRS/LTE profile manipulation	15
5.3 Category 3 attack example - Hostile Location Update	16
5.4 Category 3 attack example - Register/Activate SS	17
5.5 Category 2 protection bypass	18
5.6 Category 3 protection bypass	19
5.7 MITM	20
5.8 Passive Attacks	20
5.9 Combining Passive and Active Attacks	21
5.10 Malformed messages	22
5.11 Advanced Attacks Conclusion	22
6 SigFW	23
6.1 Open SS7 Firewall	24
6.1.1 Architecture	24
6.1.2 Deployment	25

6.1.3	APIs	25
6.1.4	Config	25
6.1.5	Signaling Message Evaluation API	26
6.1.6	SS7 Firewall Passive Mode	27
6.1.7	SS7 Encryption	28
6.1.8	SS7 Encryption Flow	28
6.1.9	SS7 Encryption Algorithm	29
6.1.10	SS7 Encryption Example	29
6.1.11	SCCP UDT / XUDT	30
6.1.12	SS7 Encryption Autodiscovery	30
6.1.13	SS7 Encryption Flow - autodiscovery	31
6.1.14	SS7 Signature	31
6.1.15	SS7 Signature Algorithm	32
6.1.16	SS7 Signature Example	32
6.1.17	DNAT to Honeypot	32
6.1.18	DNAT to Honeypot Example	33
6.1.19	mThreat	33
6.1.20	mThreat Example	34
6.2	Open Diameter Firewall	35
6.2.1	Architecture	35
6.2.2	Deployment	36
6.2.3	Diameter Encryption Flow	36
6.2.4	Diameter Encryption Algorithm	37
6.2.5	Diameter Encryption Example	37
6.2.6	Diameter Encryption Autodiscovery	38
6.2.7	Diameter Signature Algorithm	38
6.2.8	Diameter Signature	39
7	Closing remarks	40
7.1	VM architecture	40
7.2	SigFW use cases	40
8	Related Open Source Contribution	42
8.1	Tshark to Elasticsearch export and security monitoring with Kibana	42
9	References and Acknowledgement	44

10	Annex	45
10.1	SS7FW VM readme	45
10.2	SS7FW configuration example	45
10.3	DiameterFW configuration example	49
10.4	SS7FW API specification	50
10.4.1	Provisioning FW rules API	50
10.4.2	Evaluation API	57
10.4.3	mThreat API	58

1 Abstract

The SS7 mobile vulnerabilities affect the security of all mobile users worldwide. The SS7 is signalisation between Mobile Operators Core Network about where your mobile phone is located and where to send media, so the secured end-device does not help here, as it is only a consequence of having legitimate SS7 traffic. To protect against SS7 vulnerabilities, you need to play at operator-level. And this was not really the kind of thing you could do up till now.

Let's change this. In this talk we propose methods that allow any operator in the world - not only the rich ones - to protect themselves and send the attackers' tricks back to the sender. What if SS7 became a much more difficult and problematic playground for the attacker?

In this talk, we will discuss the current status, possible solutions, and outline advanced SS7 attacks and defenses using open-source SS7 firewall which we will publish after the talk. The signaling firewall is new, so we will not only use it to reduce the vulnerabilities in the SS7 networks, but we also show how to trick and abuse the attackers to make the work much harder for attackers, and give them a hard time interpreting the results. Intelligence agencies love SS7 for the wrong reasons. We will show examples and how we can make eavesdropping and geolocation a nightmare for these nation-state attackers.

The adoption of such signaling firewall could help to reduce the exposure for both active and passive attacks on a larger scale. We will present the capabilities of this solution including the encryption of signaling, report the attacks to central threat intelligence and forward the attackers to honeypot. So what about to find where these SS7 attacks are coming and to start protecting the networks?

2 Introduction

2.1 Problem Statement

The international SS7 network has been standardized and built in past as trusted network with only trusted partners. The network itself and by design does not authenticate and authorize the peers in the network and also does not encrypt the signalling communication. The exposure of these networks comes from the design and the architecture requirement of roaming architecture in past architecture releases.

Additionally we should not expect that the SS7 network will be phased out soon. The voice could be replaced by VoLTE (4G) with IMS home routed architecture, but such deployment requires VoLTE capable devices and VoLTE networks with the similar radio coverage compared to 2G, 3G. So before some operator decide to shut-down both 2G and 3G network, all the home subscribers should be VoLTE enabled. And the operator should consider also inbound-roamers.

In the LTE the Diameter protocol has replaced the SS7 signalling. However the similar issues are still present. Lack of authentication and no encryption of the signalling communication.

2.2 Related work

Several companies are offering commercial signalling firewalls and also there has been significant work on GSMA level. However we still think the problem is not fully covered. These commercial firewall solutions are reducing the risk up to some level mainly with focus on HPLMN protection, but are not so widely adopted and still there are several ways how the protection could be bypassed. These technical corner cases comes mainly from possibility of spoofing of the SCCP and Diameter messages and lack of protection of subscribers while being in roaming. Here we provide novel approach to fixing this thanks to open source approach and new signing and encryption approach.

3 The Approach

In this work we will outline some advanced SS7 attacks, including spoofing of messages, targeting roaming subscribers, some possible attacks done by MITM and passive attacks which are not addressed much by the industry today.

We will describe the open source SS7 and Diameter firewall (SigFW) using open source SS7 and Diameter stack which could be used to help to address the signalling vulnerabilities and the advanced attacks.

The open-source SigFW should be considered as **reference implementation** and **research project** but **without any warranty** and it is not a carrier grade solution.

3.1 SS7 firewall - Technical capabilities

- Open SS7 TCAP encryption and signing of the SS7 messages, including auto encryption setup
- SS7 SCCP blacklists (Category 0)
- SS7 TCAP blacklists (Category 1)
- SS7 MAP firewall rules (Category 2)
- Signalling IDS integration (for Category 3 and advanced detection)
- SS7 Filtering and honeypoting
- Centralized threat reporting with mThreat integration
- Collaboration with other SS7 and signaling security systems
- Management through open APIs
- Passive run (re-run traffic from pcap or passive interface to test the firewall)
- LUA programmable firewall rules
- Scalable/Decentralized solution

3.2 Diameter firewall - Technical capabilities

- Open Diameter encryption and signing of the Diameter messages, including auto encryption setup
- Diameter host and realms blacklists (Category 0)
- Diameter Command Code blacklists and Realm whitelist (Category 1)
- Diameter firewall rules (Category 2)
- Signalling IDS integration (for Category 3 and advanced detection)
- Diameter Filtering and honeypoting
- Centralized threat reporting with mThreat integration
- Collaboration with other Diameter and signaling security systems
- Management through open APIs
- Passive run (re-run traffic from pcap or passive interface to test the firewall)
- LUA programmable firewall rules

- Scalable/Decentralized solution

Additionally we will outline also the contribution which could be used for network monitoring and could be used in this domain but also in other domains.

- Tshark to Elasticsearch export and security monitoring with Kibana

4 The Current Status

In the following chapter we would be briefly outline the current possible approach regarding the message filtering and screening on the network boundaries.

4.1 SS7 / Sigtran stack overview

On the following figure is illustrated SS7/Sigtran protocol stack. This is important to understand for decoding and filtering reasons.

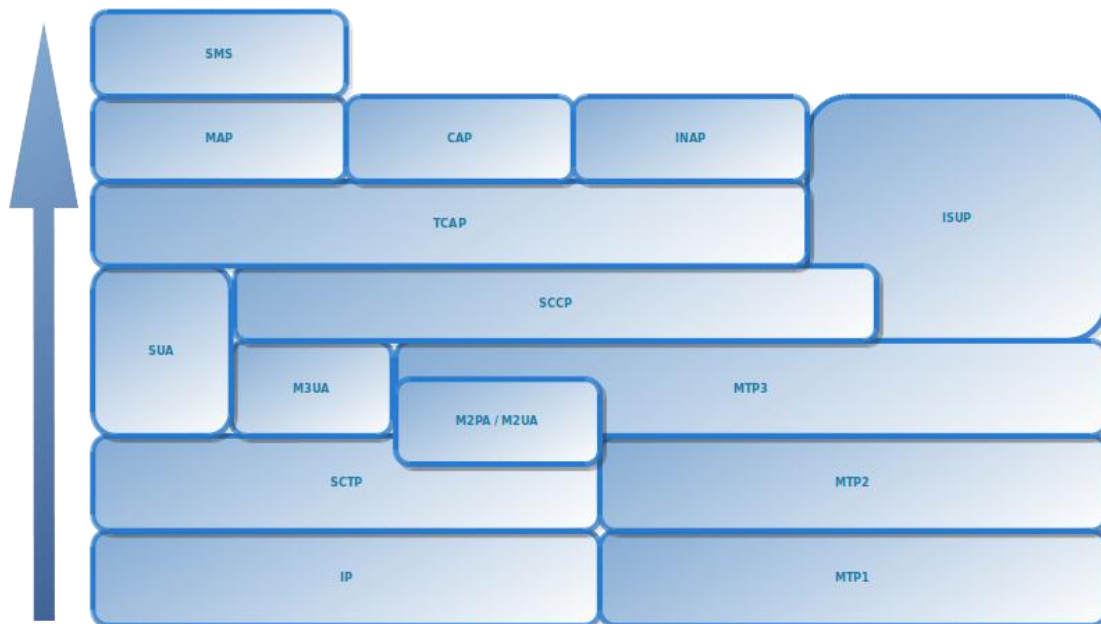


Figure 4.1 - SS7 and Sigtran stack

4.2 Perimeters of SS7 overview

The active filtering and the protection could be efficiently performed on the network boundaries and on the perimeters of the home network (HPLMN). We can consider mainly the following perimeters:

INAT 0: International interconnects (higher risk)

NAT 1: National interconnects (possibly lower risk)

There could exist the different security filtering for these perimeters. International interconnects are used mainly for inbound and outbound roaming subscribers. The national interconnects are commonly used for SMS delivery, roaming if the national roaming is allowed and forwarding signalling messages in case of number portability.

For overall security we should consider also other interfaces and interconnects e.g. with MVNOs or API towards SMSC and with 3rd party SMS aggregators.

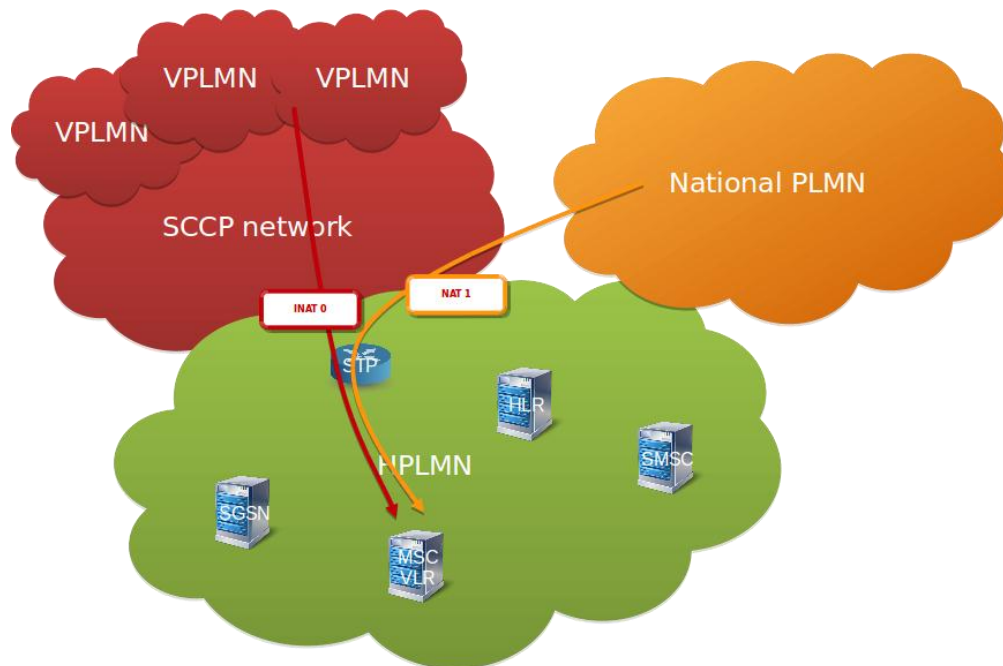


Figure 4.2 - SS7 perimeters

4.3 SS7 message categories

Category is just naming indicating the group of the similar messages. For messages in same category the same protection logic could be implemented. Mainly the message direction is important to decide into which category the message belongs. The normal call flows and normal use of the message is well described in 3GPP specifications.

MAP Cat1 messages are messages which should not be allowed towards HPLMN.

MAP Cat2 messages are messages which should be allowed towards HPLMN only if foreign network is targeting own subscribers (inbound-roamers).

MAP Cat3 messages are messages which should be allowed towards HPLMN from own subscribers in roaming (outbound-roamers) only if location condition matches.

SMS Cat: SMS messages which requires to decode SMS layer.

CAP Category 2 messages are Camel messages which should be allowed for inbound-roamers from HPLMN towards foreign network (inbound-roamers).

CAP Category 3 messages are Camel messages which should be allowed for outbound-roamers from VPLMN towards HPLMN.

From the above approach the messages could be classified into message categories and could be created protocol matrixes for SS7 but also for Diameter and GTP protocol. Then the protection could be implemented in the Signalling Firewall or in the Network Elements.

ISX - Command Codes			
Command Code	Command Name	Application-Id	Application-Id name
0 - 255	RADIUS compatibility codes		
256	Unassigned		
257	CCR / CEA - Capabilities Exchange		
MAP CC - INAP			
Operation Code	MAP Operation	MAP AC Code	Application Context
258	RAR / RAA - Register Request	2	updateLocation
259	Unassigned	3	cancelLocation
260	RMR / RMA - AA-Mobile-Number	4	provideRoutingNumber
261	Unassigned	5	cancelRoutingNumber
262	HAR / HAA - Home Agent-MIP	6	cancelHomeDataTransfer
263 - 264	Unassigned	6	cancelHomeDataTransfer
265	NAR / NAA - Authorize Authentication	7	insertSubscriberData
266 - 267	Unassigned	7	insertSubscriberData
268	CCR / CEA - Diameter-EAP-Request	7	insertSubscriberData
269 - 270	Unassigned	7	insertSubscriberData
271	ACR / ACA - Accounting-Request	8	deleteSubscriberData
272	CCR / CCA - Credit-Control-Request	9	deleteSubscriberData
273	Unassigned	9	deleteSubscriberData
274	ASR / ASA - Abort-Session-Request	9	deleteSubscriberData
275	STR / STA - Session-Termination-Request	9	deleteSubscriberData
276 - 279	Unassigned	9	deleteSubscriberData
280	DWR / DWA - Device-Watching	11	registerSS
281	Unassigned	12	cancelSS
282	DPR / DPA - Disconnect-Peer	13	cancelSS
283	UAR / UAA - User-Authorization-Request	14	cancelSS
284	SAR / SAA - Server-Assignment-Request	15	cancelSS
285	LIR / LIA - Location-Info-Request	16	cancelSS
286	MAR / MAA - Multimedia-Auth-Request	17	cancelSS
287	RTR / RTA - Registration-Termination-Request	18	cancelSS
288	Unassigned	19	cancelSS
289	Unassigned	20	cancelSS
290	Unassigned	21	cancelSS
291	Unassigned	22	cancelSS
292	Unassigned	23	cancelSS
293	Unassigned	24	cancelSS
294	Unassigned	25	cancelSS
295	Unassigned	26	cancelSS
296	Unassigned	27	cancelSS
297	Unassigned	28	cancelSS
298	Unassigned	29	cancelSS
299	Unassigned	30	cancelSS
300	Unassigned	31	cancelSS
301	Unassigned	32	cancelSS
302	Unassigned	33	cancelSS
303	Unassigned	34	cancelSS
304	Unassigned	35	cancelSS
305	Unassigned	36	cancelSS
306	Unassigned	37	cancelSS
307	Unassigned	38	cancelSS
308	Unassigned	39	cancelSS
309	Unassigned	40	cancelSS
310	Unassigned	41	cancelSS
311	Unassigned	42	cancelSS
312	Unassigned	43	cancelSS
313	Unassigned	44	cancelSS
314	Unassigned	45	cancelSS
315	Unassigned	46	cancelSS
316	Unassigned	47	cancelSS
317	Unassigned	48	cancelSS
318	Unassigned	49	cancelSS
319	Unassigned	50	cancelSS
320	Unassigned	51	cancelSS
321	Unassigned	52	cancelSS
322	Unassigned	53	cancelSS
323	Unassigned	54	cancelSS
324	Unassigned	55	cancelSS
325	Unassigned	56	cancelSS
326	Unassigned	57	cancelSS
327	Unassigned	58	cancelSS
328	Unassigned	59	cancelSS
329	Unassigned	60	cancelSS
330	Unassigned	61	cancelSS
331	Unassigned	62	cancelSS
332	Unassigned	63	cancelSS
333	Unassigned	64	cancelSS
334	Unassigned	65	cancelSS
335	Unassigned	66	cancelSS
336	Unassigned	67	cancelSS
337	Unassigned	68	cancelSS
338	Unassigned	69	cancelSS
339	Unassigned	70	cancelSS
340	Unassigned	71	cancelSS
341	Unassigned	72	cancelSS
342	Unassigned	73	cancelSS
343	Unassigned	74	cancelSS
344	Unassigned	75	cancelSS
345	Unassigned	76	cancelSS
346	Unassigned	77	cancelSS
347	Unassigned	78	cancelSS
348	Unassigned	79	cancelSS
349	Unassigned	80	cancelSS
350	Unassigned	81	cancelSS
351	Unassigned	82	cancelSS
352	Unassigned	83	cancelSS
353	Unassigned	84	cancelSS
354	Unassigned	85	cancelSS
355	Unassigned	86	cancelSS
356	Unassigned	87	cancelSS
357	Unassigned	88	cancelSS
358	Unassigned	89	cancelSS
359	Unassigned	90	cancelSS
360	Unassigned	91	cancelSS
361	Unassigned	92	cancelSS
362	Unassigned	93	cancelSS
363	Unassigned	94	cancelSS
364	Unassigned	95	cancelSS
365	Unassigned	96	cancelSS
366	Unassigned	97	cancelSS
367	Unassigned	98	cancelSS
368	Unassigned	99	cancelSS
369	Unassigned	100	cancelSS
370	Unassigned	101	cancelSS
371	Unassigned	102	cancelSS
372	Unassigned	103	cancelSS
373	Unassigned	104	cancelSS
374	Unassigned	105	cancelSS
375	Unassigned	106	cancelSS
376	Unassigned	107	cancelSS
377	Unassigned	108	cancelSS
378	Unassigned	109	cancelSS
379	Unassigned	110	cancelSS
380	Unassigned	111	cancelSS
381	Unassigned	112	cancelSS
382	Unassigned	113	cancelSS
383	Unassigned	114	cancelSS
384	Unassigned	115	cancelSS
385	Unassigned	116	cancelSS
386	Unassigned	117	cancelSS
387	Unassigned	118	cancelSS
388	Unassigned	119	cancelSS
389	Unassigned	120	cancelSS
390	Unassigned	121	cancelSS
391	Unassigned	122	cancelSS
392	Unassigned	123	cancelSS
393	Unassigned	124	cancelSS
394	Unassigned	125	cancelSS
395	Unassigned	126	cancelSS
396	Unassigned	127	cancelSS
397	Unassigned	128	cancelSS
398	Unassigned	129	cancelSS
399	Unassigned	130	cancelSS
400	Unassigned	131	cancelSS
401	Unassigned	132	cancelSS
402	Unassigned	133	cancelSS
403	Unassigned	134	cancelSS
404	Unassigned	135	cancelSS
405	Unassigned	136	cancelSS
406	Unassigned	137	cancelSS
407	Unassigned	138	cancelSS
408	Unassigned	139	cancelSS
409	Unassigned	140	cancelSS
410	Unassigned	141	cancelSS
411	Unassigned	142	cancelSS
412	Unassigned	143	cancelSS
413	Unassigned	144	cancelSS
414	Unassigned	145	cancelSS
415	Unassigned	146	cancelSS
416	Unassigned	147	cancelSS
417	Unassigned	148	cancelSS
418	Unassigned	149	cancelSS
419	Unassigned	150	cancelSS
420	Unassigned	151	cancelSS
421	Unassigned	152	cancelSS
422	Unassigned	153	cancelSS
423	Unassigned	154	cancelSS
424	Unassigned	155	cancelSS
425	Unassigned	156	cancelSS
426	Unassigned	157	cancelSS
427	Unassigned	158	cancelSS
428	Unassigned	159	cancelSS
429	Unassigned	160	cancelSS
430	Unassigned	161	cancelSS
431	Unassigned	162	cancelSS
432	Unassigned	163	cancelSS
433	Unassigned	164	cancelSS
434	Unassigned	165	cancelSS
435	Unassigned	166	cancelSS
436	Unassigned	167	cancelSS
437	Unassigned	168	cancelSS
438	Unassigned	169	cancelSS
439	Unassigned	170	cancelSS
440	Unassigned	171	cancelSS
441	Unassigned	172	cancelSS
442	Unassigned	173	cancelSS
443	Unassigned	174	cancelSS
444	Unassigned	175	cancelSS
445	Unassigned	176	cancelSS
446	Unassigned	177	cancelSS
447	Unassigned	178	cancelSS
448	Unassigned	179	cancelSS
449	Unassigned	180	cancelSS
450	Unassigned	181	cancelSS
451	Unassigned	182	cancelSS
452	Unassigned	183	cancelSS
453	Unassigned	184	cancelSS
454	Unassigned	185	cancelSS
455	Unassigned	186	cancelSS
456	Unassigned	187	cancelSS
457	Unassigned	188	cancelSS
458	Unassigned	189	cancelSS
459	Unassigned	190	cancelSS
460	Unassigned	191	cancelSS
461	Unassigned	192	cancelSS
462	Unassigned	193	cancelSS
463	Unassigned	194	cancelSS
464	Unassigned	195	cancelSS
465	Unassigned	196	cancelSS
466	Unassigned	197	cancelSS
467	Unassigned	198	cancelSS
468	Unassigned	199	cancelSS
469	Unassigned	200	cancelSS
470	Unassigned	201	cancelSS
471	Unassigned	202	cancelSS
472	Unassigned	203	cancelSS
473	Unassigned	204	cancelSS
474	Unassigned	205	cancelSS
475	Unassigned	206	cancelSS
476	Unassigned	207	cancelSS
477	Unassigned	208	cancelSS
478	Unassigned	209	cancelSS
479	Unassigned	210	cancelSS
480	Unassigned	211	cancelSS
481	Unassigned	212	cancelSS
482	Unassigned	213	cancelSS
483	Unassigned	214	cancelSS
484	Unassigned	215	cancelSS
485	Unassigned	216	cancelSS
486	Unassigned	217	cancelSS
487	Unassigned	218	cancelSS
488	Unassigned	219	cancelSS
489	Unassigned	220	cancelSS
490	Unassigned	221	cancelSS
491	Unassigned	222	cancelSS
492	Unassigned	223	cancelSS
493	Unassigned	224	cancelSS
494	Unassigned	225	cancelSS
495	Unassigned	226	cancelSS
496	Unassigned	227	cancelSS
497	Unassigned	228	cancelSS
498	Unassigned	229	cancelSS
499	Unassigned	230	cancelSS
500	Unassigned	231	cancelSS
501	Unassigned	232	cancelSS
502	Unassigned	233	cancelSS
503	Unassigned	234	cancelSS
504	Unassigned	235	cancelSS
505	Unassigned	236	cancelSS
506	Unassigned	237	cancelSS
507	Unassigned	238	cancelSS
508	Unassigned	239	cancelSS
509	Unassigned	240	cancelSS
510	Unassigned	241	cancelSS
511	Unassigned	242	cancelSS
512	Unassigned	243	cancelSS
513	Unassigned	244	cancelSS
514	Unassigned	245	cancelSS
515	Unassigned	246	cancelSS
516	Unassigned	247	cancelSS
517	Unassigned	248	cancelSS
518	Unassigned	249	cancelSS
519	Unassigned	250	cancelSS
520	Unassigned	251	cancelSS
521	Unassigned	252	cancelSS
522	Unassigned	253	cancelSS
523	Unassigned	254	cancelSS
524	Unassigned	255	cancelSS
525	Unassigned	256	cancelSS
526	Unassigned	257	cancelSS
527	Unassigned	258	cancelSS
528	Unassigned	259	cancelSS
529	Unassigned	260	cancelSS
530	Unassigned	261	cancelSS
531	Unassigned	262	cancelSS
532	Unassigned	263	cancelSS
533	Unassigned	264	cancelSS
534	Unassigned	265	cancelSS
535	Unassigned	266	cancelSS
536	Unassigned	267	cancelSS
537	Unassigned	268	cancelSS
538	Unassigned	269	cancelSS
539	Unassigned	270	cancelSS
540	Unassigned	271	cancelSS
541	Unassigned	272	cancelSS
542	Unassigned	273	cancelSS
543	Unassigned	274	cancelSS
544	Unassigned	275	cancelSS
545	Unassigned	276	cancelSS
546	Unassigned	277	cancelSS
547	Unassigned	278	cancelSS
548	Unassigned	279	cancelSS
549	Unassigned	280	cancelSS
550	Unassigned	281	cancelSS
551	Unassigned	282	cancelSS
552	Unassigned	283	cancelSS
553	Unassigned	284	cancelSS
554	Unassigned	285	cancelSS
555	Unassigned	286	cancelSS
556	Unassigned		

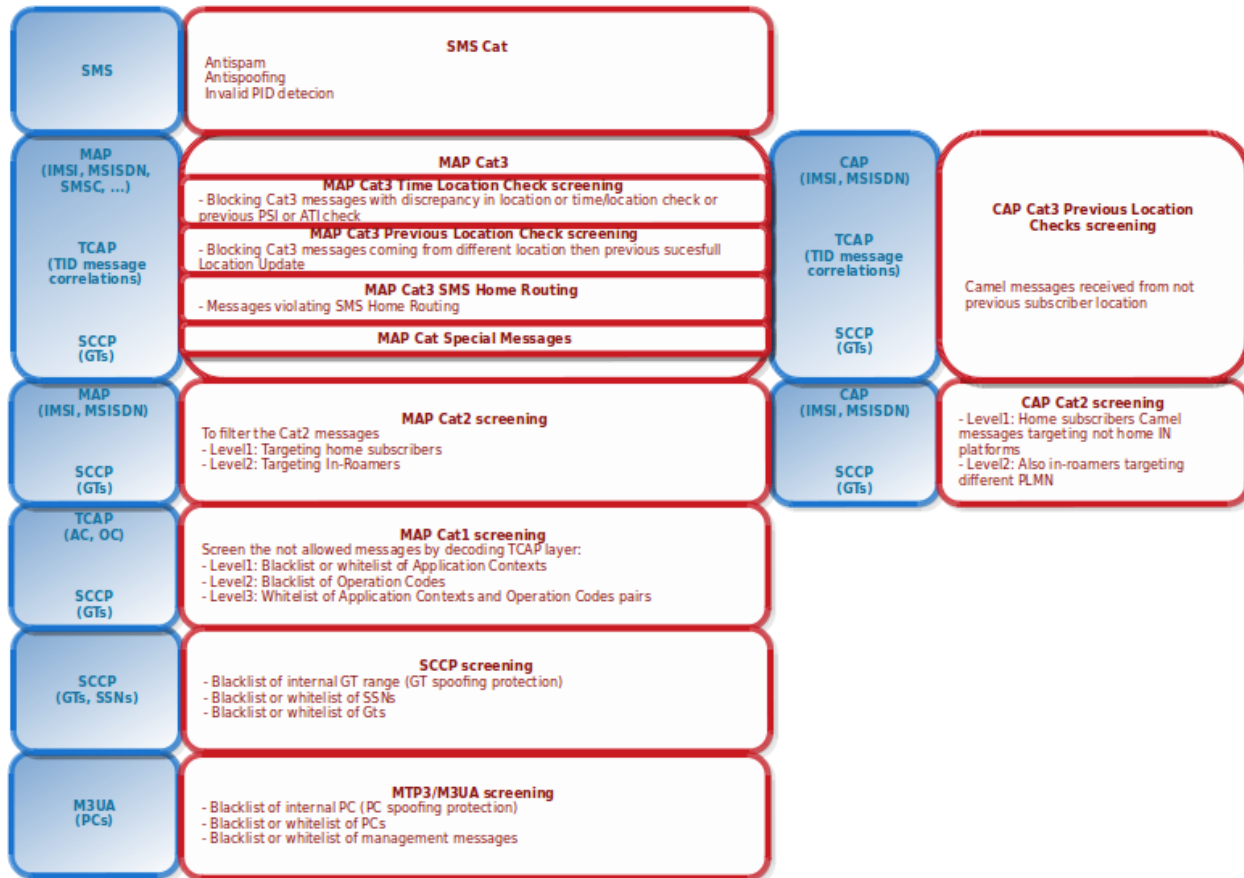


Figure 4.4 - SS7 screening categories with protocol layers

4.5 Possible SS7 filtering by existing infrastructure without FW

The filtering is possible also inside the infrastructure without having external firewall, but there are several disadvantages in this approach. (e.g. no perimeter defense, no centralized control)

Also in this approach it is hard to manage the confidentiality and integrity protection of signalling messages.

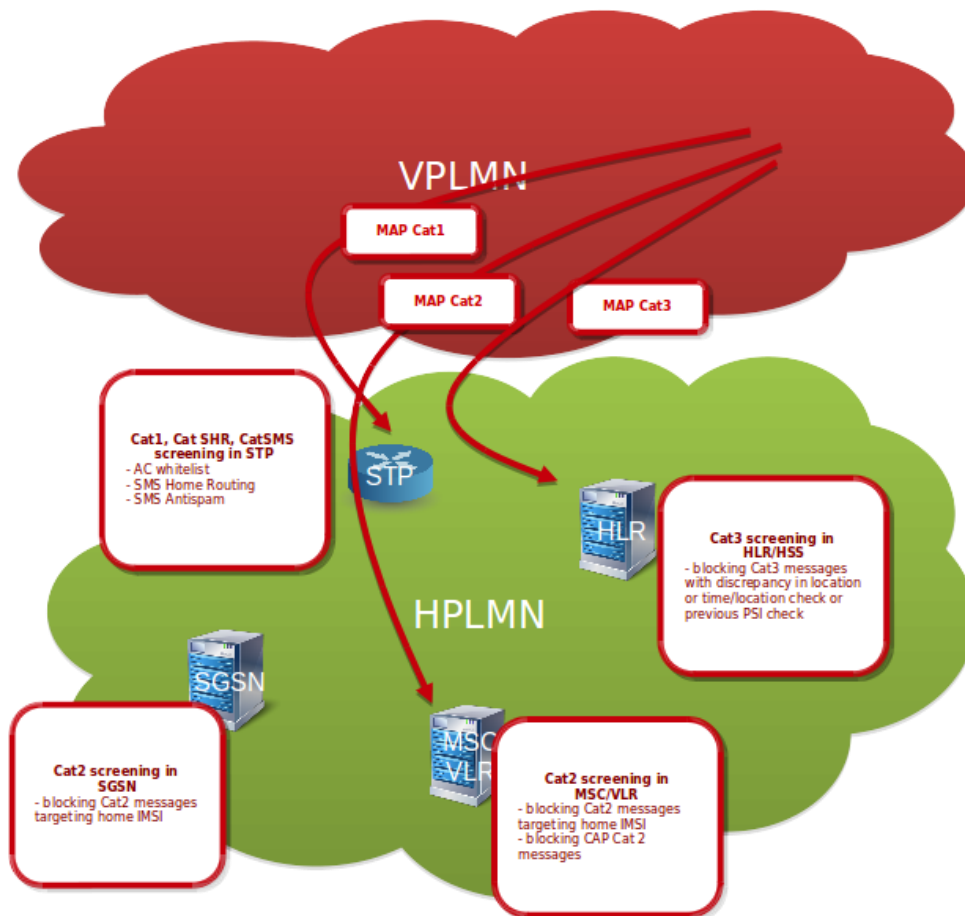


Figure 4.5 - SS7 network protected by existing infrastructure

4.6 Current status conclusion and acknowledgement

In this chapter was briefly outlined the message filtering approach on the network boundaries.

The above figures illustrates the internal research/approach but the work is inline and evolves the current GSMA recommendations. Additionally we are contributing in this direction to GSMA.

For further details of the GSMA collaborative work it could be referred to FS.11, FS.19 and FS.20 GSMA documents.

5 Advanced SS7 Attacks

In the following chapter are highlighted some attacks as examples to demonstrate the message categories. Then this is followed by examples how the protection could be bypassed while the subscriber is in roaming.

5.1 Category 2 attack example - VLR profile manipulation

Category 2 example - VLR profile manipulation. The attacker could manipulation the profile of the subscriber in the VLR.

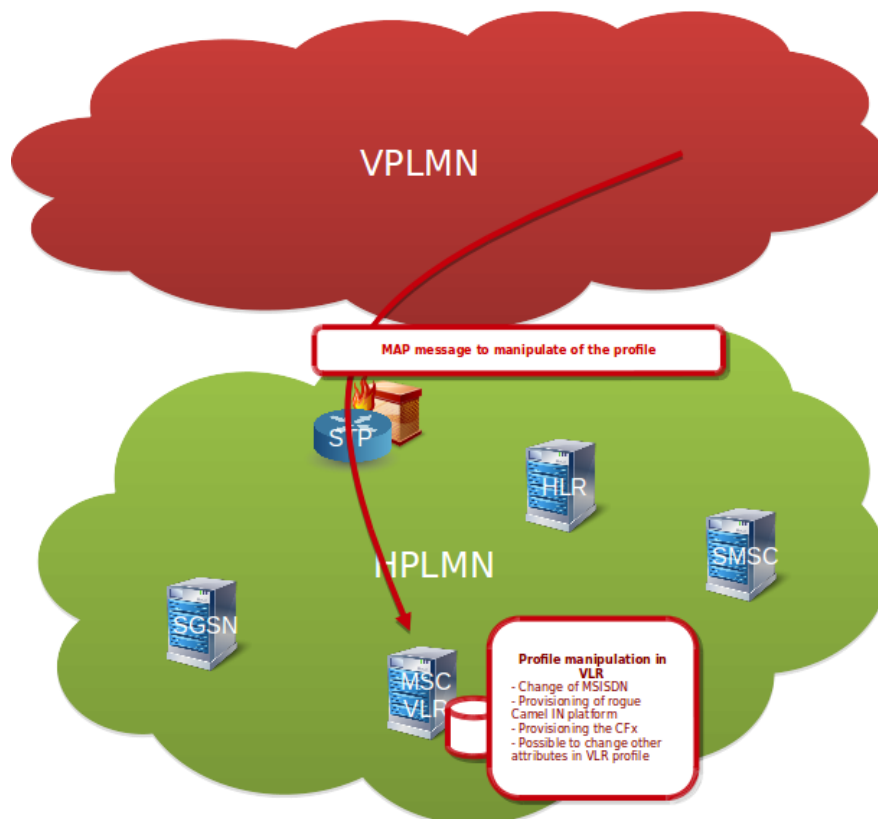


Figure 5.1 - VLR profile manipulation

Description: The figure illustrate that the attacker can craft the MAP ISD message and target the MSC/VLR which is currently serving the subscriber. If there is no protection against Category 2 attacks the attacker is able to alter the VLR profile from the attacker's GT. If in the HPLMN is Signalling FW or the protection against Category 2 attacks, the attack would fail because the attacker's GTs will belongs to different country as the HLR of the targeted subscriber.

Impact: The attacker can manipulate the whole VLR profile which could lead in the modification of MSISDN, tele/bearer services, supplementary services, barring, camel flags and the

provisioned IN platform. The possible impact is the call and SMS interception, persistent location tracking, frauds or targeted DoS of the subscriber.

5.2 Category 2 attack example - GPRS/LTE profile manipulation

Category 2 example - GPRS/LTE profile manipulation. The attacker could manipulation the profile of the subscriber in the SGSN/MME.

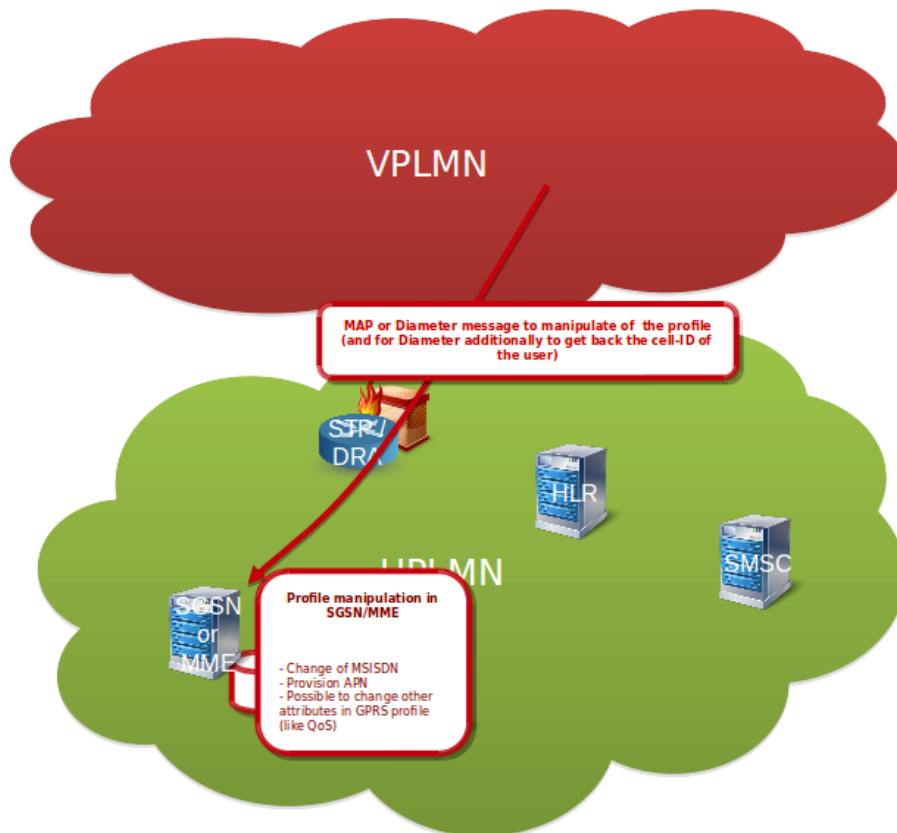


Figure 5.2 - SGSN/MME profile manipulation

Description: The figure illustrate that the attacker can craft the MAP ISD or Diameter IDR message and target the SGSN or MME which is currently serving the subscriber. If there is no protection against Category 2 attacks the attacker is able to alter the SGSN/MME profile from the attacker's GT (or Diameter Origin-Host/Realm). If in the HPLMN is Signalling FW or the protection against Category 2 attacks, the attack would fail because the attacker's GTs (or Diameter Origin-Host/Realm) will belongs to different country as the HLR/HSS of the targeted subscriber.

Impact: The attacker can manipulate the whole GPRS/LTE profile which could lead in the modification of MSISDN, APNs, QoS, camel flags and the provisioned IN platform. The possible impact is the bypass of MSISDN authentication (if HTTP enrichment and latter MSISDN authentication is used), access to private APNs and the possibly the data interception if the latter Camel is enabled in the Packet Core.

5.3 Category 3 attack example - Hostile Location Update

Category 3 example - Hostile Location Update. The attacker could change location in the HLR/HSS.

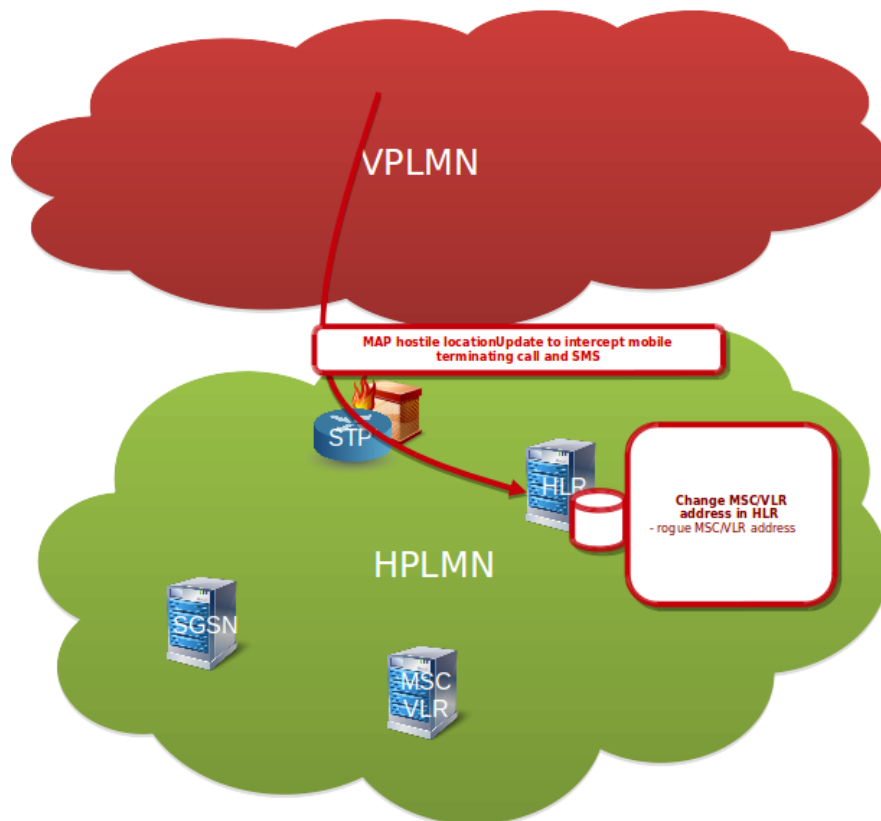


Figure 5.3 - Hostile Location Update

Description: The figure illustrate that the attacker can craft the MAP LU message towards the HLR/HSS and change the location of the subscriber to own GT. If in the HPLMN is Signalling FW or the protection against Category 3 attacks, the attack would fail because the Location Update would be interpreted as suspicious if coming from too different location compared to current location of the subscriber.

Impact: The attacker can change the subscriber GT in HLR/HSS. This could lead into MT-SMS interception, possibly MT-Call interception if the attacker can also connect the original B-party after or targeted DoS of the subscriber. Additionally could be used also as precondition for latter Category 3 attacks.

5.4 Category 3 attack example - Register/Activate SS

Category 3 example - Register/Activate SS. The attacker could manipulate the supplementary services in HLR/HSS.

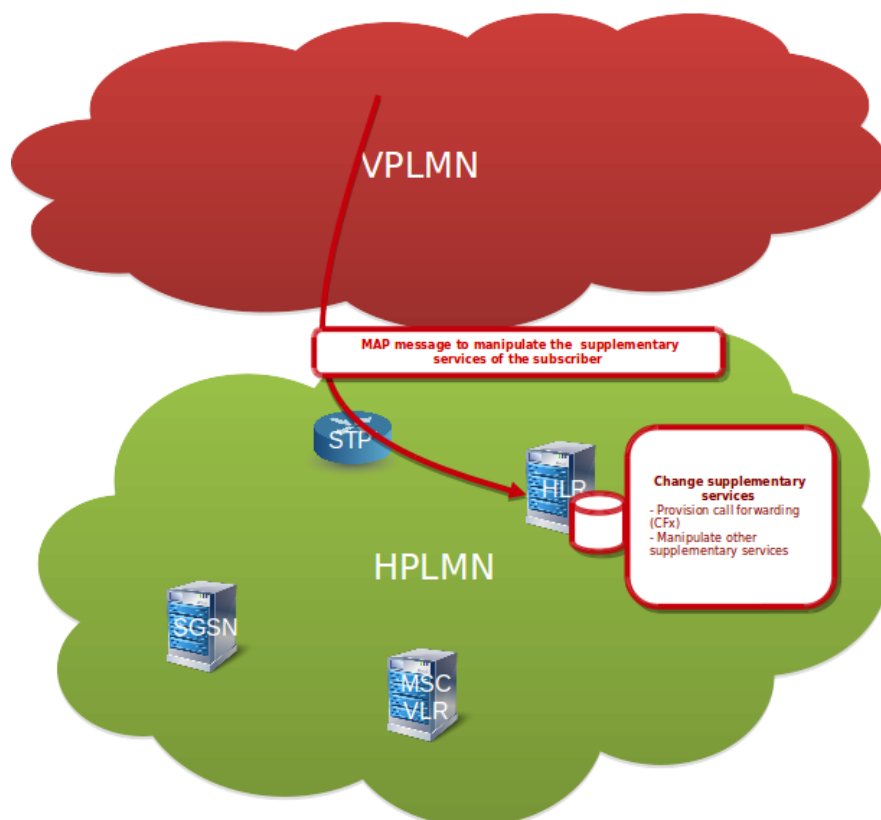


Figure 5.4 - Register/Activate SS

Description: The figure illustrate that the attacker can craft the Register/Activate SS message and target the HLR/HSS. If there is no protection against Category 3 attacks the attacker is able to alter the SS services in HLR. If in the HPLMN is Signalling FW or the protection against Category 3 attacks, the attack would fail because the attacker's GTs will not match with the current subscriber location.

Impact: The attacker can manipulate the whole SS service in HLR/HSS, which could lead on activation of call/SMS forwarding and other SS manipulation.

5.5 Category 2 protection bypass

Outbound-roamer in VPLMN: Attack targeting outbound-roamers with Cat2 messages with spoofed calling GT.

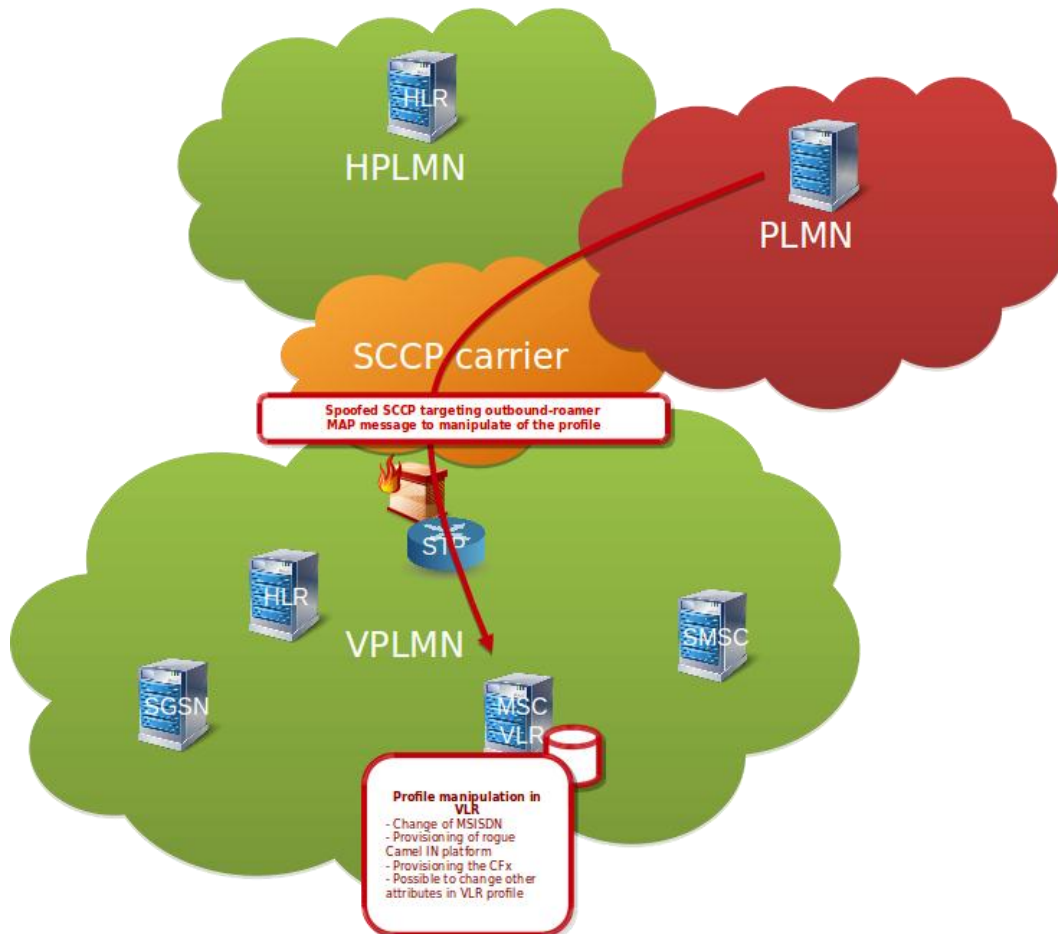


Figure 5.5 - Category 2 protection bypass

Description: The figure illustrate that when the subscriber is located in roaming network (VPLMN) and if the attacker knows his VLR/SGSN address (e.g. discovered by other SS7 messages, like SRI-SM or by passive sniffing), the attacker can send spoofed Cat2 SS7 messages and impersonate subscriber HLR from HPLMN. For such attack the signalling firewall in VPLMN would be not able to discard the message and differentiate it from legitimate signalling, because the message is spoofed with the correct Calling SCCP Address.

Impact: Subscriber in roaming could not be easily protected against spoofed SCCP attacks or could be difficult if the Calling SCCP Address is from same country as the legitimate one. This results into possible VLR and SGSN profile manipulation, which could lead into setting call forwarding, removing services, provisioning Camel services and other. (DoS, tracking,

interception). For spoofed messages for SS7 the attacker would not get result message but for Diameter would, because of the Route-Record AVP.

5.6 Category 3 protection bypass

Outbound-roamer in VPLMN: Attacker first performing hostile LocationUpdate (if not working could use spoofed Cancel Location first). After performing Cat3 messages.

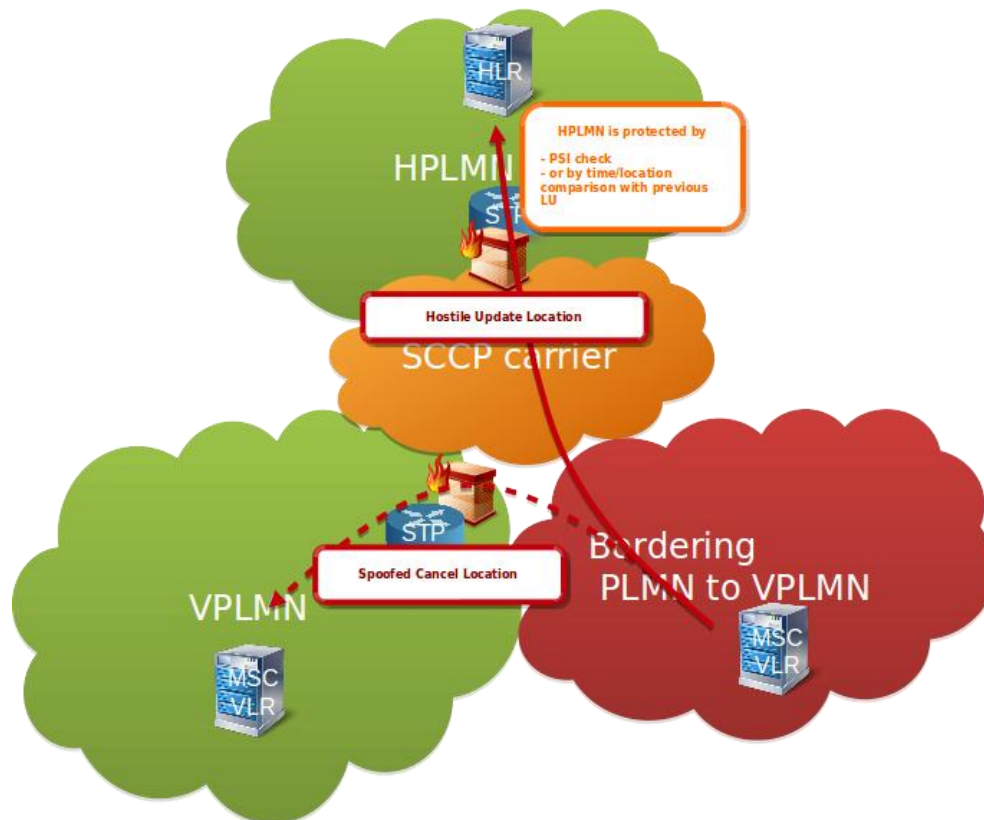


Figure 5.6 - Category 3 protection bypass

Description: The hostile Location Update sent by attacker will try change the VLR/SGSN address in the HLR first before sending later Category 3 messages. The reason for this is that in HPLMN is implemented Signalling Firewall or other protection against Category 3 messages with the following behaviour.

Option 1. - the protection in HPLMN for Cat3 messages is implemented by sending PSI message to previously known subscriber location, to verify that the subscriber is not located anymore there. This protection is possible to bypass by the hostile location update first.

Option 2. - the protection in HPLMN is implemented by time/distance analysis of previous and current location updates. This is possible to bypass by sending the hostile location update from not suspicious location (e.g. bordering country).

If the Hostile Location Update is not successful, the attacker can try to first send the spoofed Cancel Location to the current MSC/VLR to bypass any PSI checks and then try send again LU or any other Cat3 messages.

Additionally attacker can also spoof directly the calling GT of latter Category 3 messages if knows the current subscriber location.

Impact: Hostile Location Update could lead directly to DoS, SMS interception and call interception (in case the attacker is capable of receiving media and connect back the B-party). Further also enables to attacker to send later the Cat3 messages (e.g. supplementary services activation, mobile originating SMS, USSD and other) because the protection by comparing the previous subscriber location with origin of the message would be bypassed.

5.7 MITM

Description: Not encrypted SCTP protocol used for Sigtran and Diameter is vulnerable to man-in-the-middle attacks. See below extract from RFC.

SCTP (RFC 3257)

5.3 Security Issues with both TCP and SCTP

```
It is important to note that neither TCP nor SCTP protect itself from
man-in-the-middle attacks where an established session might be
hijacked (assuming the attacker can see the traffic from and inject
its own packets to either endpoints).
```

Impact: Attacker could get access into SS7 network by MITM in SCTP without being configured or provisioned on SS7 network. By having such capability motivated attacker with physical access to links could inject traffic into signalling network. This means not only attacker having SCCP address and connectivity with STP or with other network element could get access into SS7 network. Additionally in MITM scenario further attacks are possible, like ISD/profile modification, authentication vectors modification (RES, IK, CK, AUTN), modification and integrity changes also of SS7 Result messages.

5.8 Passive Attacks

Description: SS7 signalisation is not confidentiality protected.

Impact: This could be used for mass collection of signalling data includes mainly:

- SMS content with A-party, B-party information
- Locations (MAP, CAP, Diameter)
- From SS7 MAP possible to get CK, IK
- Get TCAP TID which could be used for latter attacks

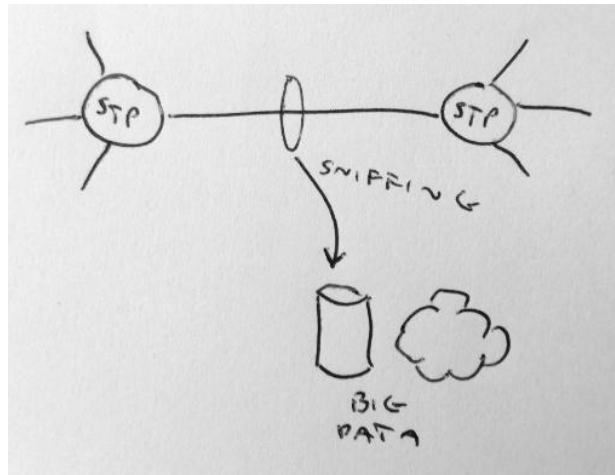


Figure 5.8 - SS7 passive attack

5.9 Combining Passive and Active Attacks

Description: By knowing the TCAP TID in real time and exact user location it could lead to more sophisticated attacks. And if the attacker is able to capture the result messages answered to spoofed messages this will also increase the capabilities.

Impact:

- Injection of messages into TCAP dialog, possibly hijacking the state machine in network elements and other effects
- Camel manipulation towards the IN platforms
- Better targeted spoofing of the SCCP messages
- Capturing the result messages to spoofed messages

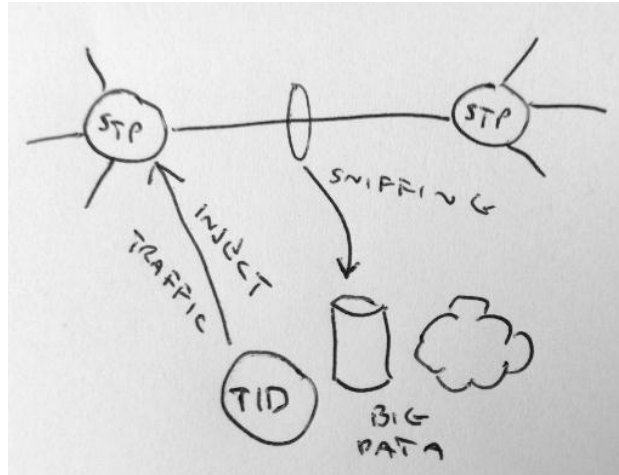


Figure 5.9 - SS7 passive and active attack

5.10 Malformed messages

Description: There is various ways of manipulating and malforming the messages. This could lead into exploitation of the vulnerability in the specific product/version of the network element.

Impact: Could lead to DoS or Exploitation (even DoS of the whole network)

5.11 Advanced Attacks Conclusion

To address the above advanced types of attacks the signalling should be confidentially and integrity protected.

A firewall with only filtering could well protect the home subscribers in HPLMN. But the home subscribers in VPLMN or inbound-roamers in HPLMN could not be easily protected mainly because the SS7, Diameter is vulnerable to spoofing and the Location Update is not authenticated.

The encryption can be done on TCAP layer or Diameter/AVP. (the current work is using proprietary implementation using asymmetric encryption)

Messages can be integrity protected carrying signature. (the current work is using proprietary implementation)

**IPSec is not suitable, because the SCCP and IPX network is required to perform routing.*

6 SigFW

Open-source SigFW

- SS7 and Diameter Firewall created under P1 Labs
- Source code is available at <https://github.com/P1sec/SigFW>

The open-source SigFW should be considered as **reference implementation** and **research project** but **without any warranty** and it is not carrier grade solution.

6.1 Open SS7 Firewall

The SS7 firewall could be considered as roaming and interconnection protection (the reference implementation) for 2G and 3G networks.

6.1.1 Architecture

Frames are forwarded on SCCP layer (using SCCP state-machine).

Filtering is possible up to application layer (in code is currently implemented SCCP, TCAP, MAP).

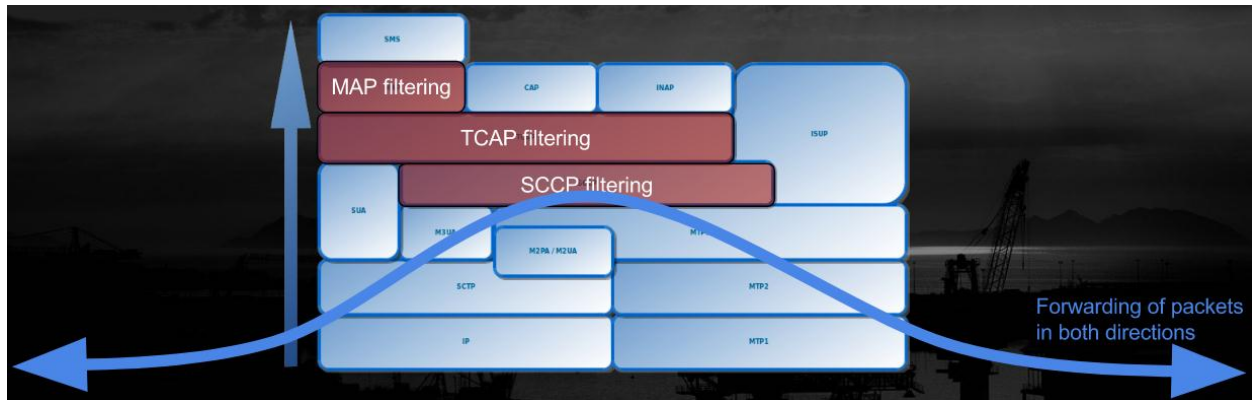


Figure 6.1.1a - SS7 Firewall decoding and filtering

Firewall is acting like M3UA server and M3UA client, without having SCCP GT. Below is an illustration of the direction of links and associations establishment.

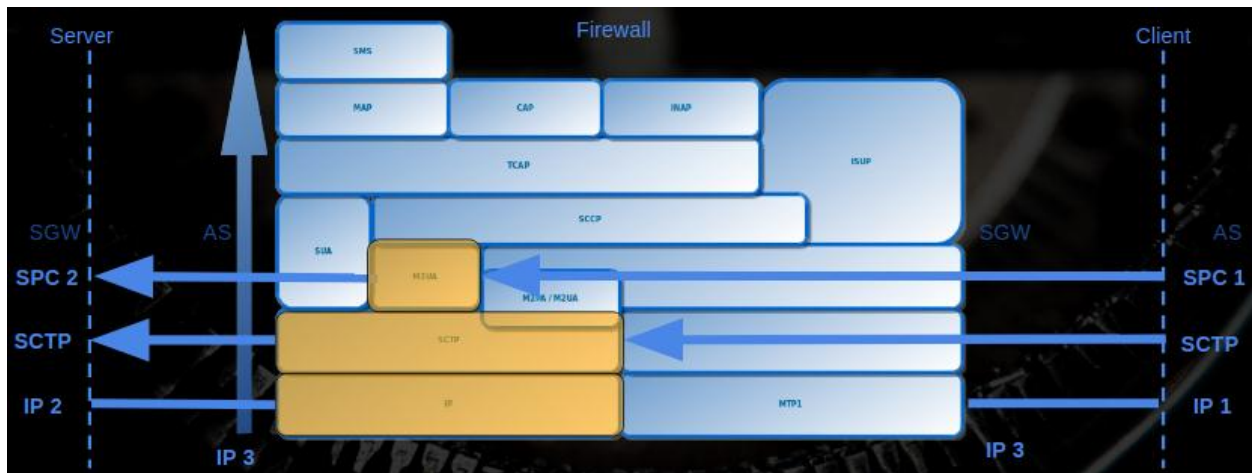


Figure 6.1.1b - SS7 Firewall connections

6.1.2 Deployment

Possible deployment can be loopback on STP towards the FW. Also other deployment scenarios could be FW deployed directly on the link or FW just protecting single network element.

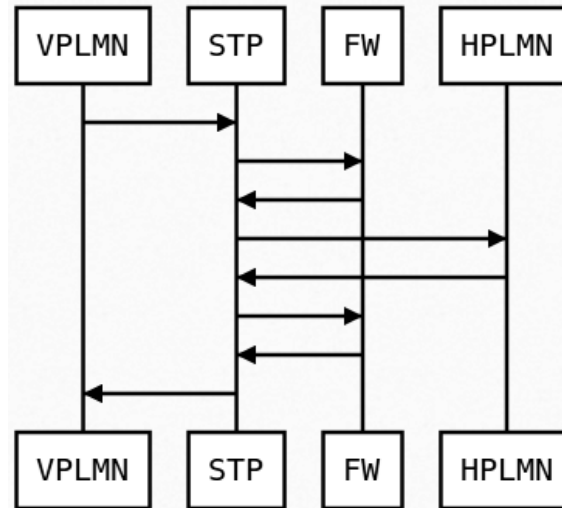


Figure 6.1.2 - SS7 Firewall deployment

6.1.3 APIs

The following REST API are currently implemented on the firewall. The API allows the remote management, provisioning the firewall rules or evaluating the messages or reporting the alerts.

1. Signaling Filter Push API (Manage Firewall Rules)
2. Signaling Message Evaluation API (Message evaluation with external IDS signalling system)
3. mThreat API (to report the detected attacks)

6.1.4 Config

- JSON syntax
- IP, SCTP, M3UA configuration
- Firewall filtering rules
- Encryption and signature keys
- Config is periodically saved to store the changes (changes over API or collected Public Keys if autodiscovery is enabled)

Figure below is the example of the configuration file. For full examples for both SS7 and Diameter see annex.

```
'firewall_rules': {
  "firewall_rules_comment": "# Firewall filtering rules con

  "firewall_policy_comment": "# Allowed value is one from:
  "firewall_policy": "DROP_WITH_SCCP_ERROR",

  "sccp": {
    >> "sccp_comment": "# SCCP firewall rules",
    >> "calling_gt_whitelist": [
    >>     "4*"
    >> ],
    >> "calling_gt_blacklist": [
    >>     "10000000000",
    >>     "222*"
    >> ],
  },

  "tcap": {
    >> "tcap_comment": "# TCAP Cat1 firewall rules",
    >> "oc_blacklist": [
    >>     "5",
    >>     "6",
    >>     "9",
    >>     "16",
    >>     "20",
    >>     "21",
    >>     "22",
    >>     "24",
    >>     "25",
```

Figure 6.1.4 - SS7 Firewall config example

6.1.5 Signaling Message Evaluation API

Signalling Message Evaluation API can be used to forward the messages which has not been detected by internal firewall rules to evaluate them in the IDS platform with more advanced detection capabilities.

- FW forwards the SCCP message to Signalling IDS
- Signalling IDS responds back with the result (allow/filter message)
- FW performs the filtering action
- By this integration no need for FW to contain own centralized DB and there could be deployed multiple FW instances
- Signalling IDS can handle more advanced Cat2, Cat3 detection, anomaly detection or threat intelligence decision

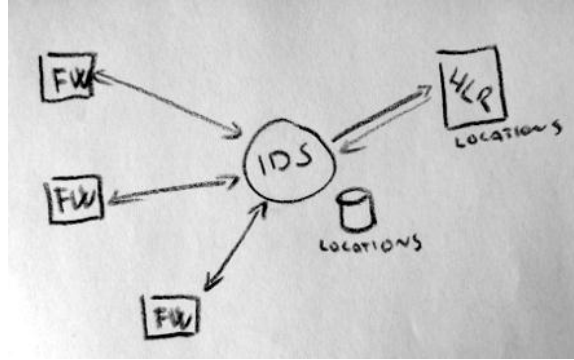


Figure 6.1.5 - SigFW with ISD integration

6.1.6 SS7 Firewall Passive Mode

The firewall can be first tested in passive mode without establishing any active signalling link. The traffic can be mirrored and be send to the FW passive network interface or the pcap/json can be directly replayed. Then the traffic is replayed on the localhost through the local client, firewall and towards the local server.

Passive mode is implemented in VM the following way:

1. tshark live capture to Json EK
2. SS7ClientLiveInput is reading scpp_raw from named pipe and forwarding it to FW
3. SS7FW performs the filtering
4. SS7Server receives the not filtered traffic

Example of replayed traffic on localhost "Passive mode":

No.	Time	Src port	Dst Port	Source	Destination	Proto	Len	Info
440776	158695.542	2345	3433	127.0.0.1	127.0.0.1	MSUA (RFC 466..	80	ASPUP
440778	158695.553	3433	2345	127.0.0.1	127.0.0.1	MSUA (RFC 466..	80	ASPUP_ACK
440780	158695.554	3433	2345	127.0.0.1	127.0.0.1	MSUA (RFC 466..	96	NTFY
440781	158695.554	2345	3433	127.0.0.1	127.0.0.1	MSUA (RFC 466..	104	SACK ASPAC
440782	158695.556	3433	2345	127.0.0.1	127.0.0.1	MSUA (RFC 466..	104	SACK ASPAC_ACK
440788	158695.753	3433	2345	127.0.0.1	127.0.0.1	MSUA (RFC 466..	96	NTFY
440886	158699.575	2344	3434	127.0.0.1	127.0.0.1	MSUA (RFC 466..	80	ASPUP
440888	158699.576	3434	2344	127.0.0.1	127.0.0.1	MSUA (RFC 466..	80	ASPUP_ACK
440890	158699.576	3434	2344	127.0.0.1	127.0.0.1	MSUA (RFC 466..	96	NTFY
440891	158699.577	2344	3434	127.0.0.1	127.0.0.1	MSUA (RFC 466..	104	SACK ASPAC
440892	158699.578	3434	2344	127.0.0.1	127.0.0.1	MSUA (RFC 466..	104	SACK ASPAC_ACK
440898	158699.777	3434	2344	127.0.0.1	127.0.0.1	MSUA (RFC 466..	96	NTFY
440974	158703.732	2345	3433	1	2	GSM SPS	296	invoke forwardSM
440997	158703.928	2345	3433	1	2	GSM MAP	31644	returnResultLast updateLocation invoke sendAuthenticat...
440999	158703.981	2344	3434	1	2	GSM MAP	9344	returnResultLast updateLocation invoke sendAuthenticat...
441005	158704.129	2345	3433	1	2	GSM MAP	9776	invoke sendRoutingInfoForSM invoke cancelLocation retu...
441007	158704.181	2344	3434	1	2	TCAP	16856	invoke sendAuthenticationInfo invoke insertSubscriberD...
441009	158704.329	2345	3433	1	2	GSM MAP	30716	returnResultLast sendAuthenticationInfo invoke insertS...
441017	158704.381	2344	3434	1	2	GSM MAP	15248	invoke sendAuthenticationInfo invoke sendAuthentication...
441021	158704.520	2345	3433	1	2	GSM MAP	10000	invoke sendAuthenticationInfo returnResultLast sendAut...
441025	158704.581	2344	3434	1	2	GSM MAP	25176	returnResultLast sendAuthenticationInfo invoke insertS...
441029	158704.729	2345	3433	1	2	GSM MAP	30852	returnError invoke sendAuthenticationInfo invoke updat...
441031	158704.781	2344	3434	1	2	GSM MAP	12420	returnResultLast insertSubscriberData returnError invo...
441037	158704.929	2345	3433	1	2	GSM MAP	9560	invoke sendAuthenticationInfo invoke sendAuthentication...

Figure 6.1.6 - SS7 Firewall passive mode

6.1.7 SS7 Encryption

Current version is capable additionally of

- Signing/Verify the SS7 message
- Encrypting/Decrypting SS7 messages

Public/Private keys are used and the security model is similar to email security (signing, encrypting).

Encryption is performed on TCAP level to pass through the STPs.

SCCP layer is not encrypted, but the SCCP addresses are used to calculate signature.

```
"encryption_rules": {
    "called_gt_encryption": [
        {
            "called_gt": "0*",
            "public_key":
"MIgfMA0GCSqGSib3DQEBAQUAA4GNADCBiQKBgQCm/PAsXOj7cjirJsQsiIeHauFNLwBIuM1brkUm3aVXeraDIEJ2BWXmWlKmmX/FRZh4Qhe9mUy6YgwTO8PndWdMDRWMw8vvXJFI7HPJpsNfcBykefSqr5X4h6HyQr73V800U5PtgCBuVoyuOFIj87WFwaLuajHiQgps7N0loeHlwIDAQAB"
        }
    ],
    "called_gt_decryption": [
    ],
},
"signature_rules": {
    "calling_gt_verify": [
    ],
    "calling_gt_signing": [
    ]
}
}
```

Figure 6.1.7 - SS7 Firewall encryption defined in the config

6.1.8 SS7 Encryption Flow

The below figure illustrates the encryption flow. The FW#1 instance in PLMN#1 encrypt the signaling messages towards the PLMN#2 because the messages matched with the GT prefix of the PLMN#2 network. The FW#2 instance in PLMN#2 network decrypt the traffic and forwards it into PLMN#2 network. The reverse direction is performed in the similar way that the FW#2 instance matches the message called GT with the GT prefix of PLMN#1 network and use the associated public key for message encryption. The messages in current model are encrypted individually without establishing session.

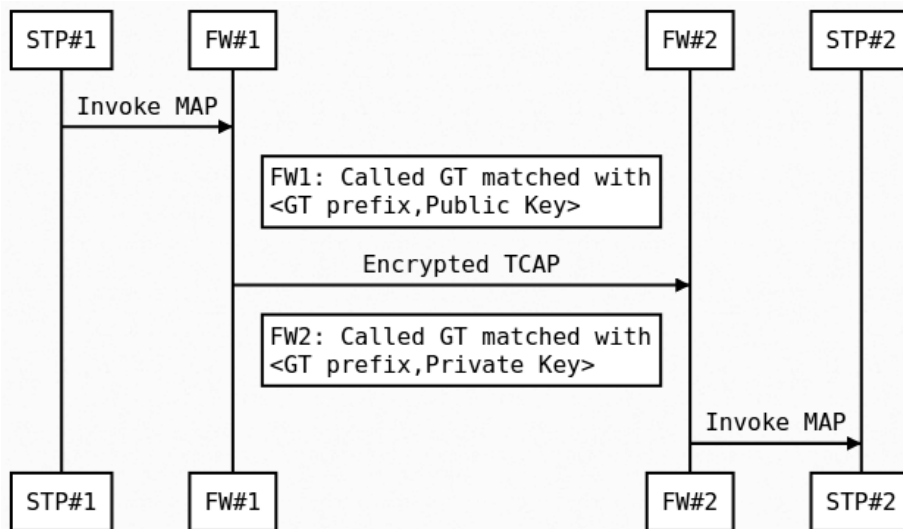


Figure 6.1.8 - SS7 Firewall encryption flow

6.1.9 SS7 Encryption Algorithm

1. Encrypted is the whole TCAP layer
2. Encrypted is the following payload:
 - a. version (4 bytes)
 - b. encrypted(timestamp (4 bytes) + tcap_layer) // If the key is short the multiple similar blocks are created
3. Encryption algorithm should be mapped with version. Currently in the code only RSA/ECB/PKCS1Padding is used
4. Timestamp is verified after decryption to prevent replay attacks

6.1.10 SS7 Encryption Example

25890	2017-04-04..	2344	3434	1111111111	VLR (Visitor Location Re...	0000000000	HLR (H...	111111...	GSM MAP	226	0000003b	invoke processUnstructuredSS-R...
25891	2017-04-04..	2345	3433	1111111111	HLR (Home Location Regis...	0000000000	VLR (V...	111111...	GSM MAP	222	0000003c	invoke unstructuredSS-Request
25893	2017-04-04..	2349	3439	1111111111,11...	HLR (Home Location Regis...	0000000000,...	VLR (V...		TCAP	614		XUDT (Message reassembled) XUD...
25895	2017-04-04..	2344	3434	1111111111	HLR (Home Location Regis...	0000000000	VLR (V...	111111...	GSM MAP	226	0000003c	invoke unstructuredSS-Request
25897	2017-04-04..	2345	3433	1111111111	HLR (Home Location Regis...	0000000000	VLR (V...	111111...	GSM MAP	210	0000003d	invoke unstructuredSS-Notify
25899	2017-04-04..	2349	3439	1111111111	HLR (Home Location Regis...	0000000000	VLR (V...		SCCP (...	362		XUDT (Message reassembled)
25903	2017-04-04..	2349	3439	1111111111	HLR (Home Location Regis...	0000000000	VLR (V...		TCAP	170		XUDT (Message reassembled)
25904	2017-04-04..	2344	3434	1111111111	HLR (Home Location Regis...	0000000000	VLR (V...	111111...	GSM MAP	218	0000003d	invoke unstructuredSS-Notify
25905	2017-04-04..	2345	3433	1111111111	gsmSCF (MAP) or IM-SSF (...	0000000000	HLR (H...		GSM MAP	206	0000003e	invoke anyTimeSubscriptionInte...
25909	2017-04-04..	2345	3433	1111111111	HLR (Home Location Regis...	0000000000	MSC (M...		GSM MAP	182	0000003f	invoke informServiceCentre
25910	2017-04-04..	2349	3439	1111111111	HLR (Home Location Regis...	0000000000	MSC (M...		SCCP (...	362		XUDT (Message reassembled)
25913	2017-04-04..	2349	3439	1111111111	HLR (Home Location Regis...	0000000000	MSC (M...		TCAP	170		XUDT (Message reassembled)
25914	2017-04-04..	2344	3434	1111111111	HLR (Home Location Regis...	0000000000	MSC (M...		GSM MAP	190	0000003f	invoke informServiceCentre
25915	2017-04-04..	2345	3433	1111111111	HLR (Home Location Regis...	0000000000	MSC (M...		GSM MAP	190	00000040	invoke alertServiceCentre
25917	2017-04-04..	2349	3439	1111111111,11...	HLR (Home Location Regis...	0000000000,...	MSC (M...		TCAP	486		XUDT (Message reassembled) XUD...
25919	2017-04-04..	2344	3434	1111111111	HLR (Home Location Regis...	0000000000	MSC (M...		GSM MAP	198	00000040	invoke alertServiceCentre
25921	2017-04-04..	2345	3433	1111111111	gsmSCF (MAP) or IM-SSF (...	0000000000	HLR (H...		GSM MAP	206	00000041	invoke anyTimeModification
25925	2017-04-04..	2345	3433	1111111111	HLR (Home Location Regis...	0000000000	VLR (V...		GSM MAP	190	00000042	invoke readyForSM
25926	2017-04-04..	2349	3439	1111111111	HLR (Home Location Regis...	0000000000	VLR (V...		SCCP (...	362		XUDT (Message reassembled)
25929	2017-04-04..	2349	3439	1111111111	HLR (Home Location Regis...	0000000000	VLR (V...		TCAP	170		XUDT (Message reassembled)
25930	2017-04-04..	2344	3434	1111111111	HLR (Home Location Regis...	0000000000	VLR (V...		GSM MAP	194	00000042	invoke readyForSM
25931	2017-04-04..	2345	3433	1111111111	VLR (Visitor Location Re...	0000000000	HLR (H...		GSM MAP	194	00000043	invoke purgeMS
25933	2017-04-04..	2349	3439	1111111111,11...	VLR (Visitor Location Re...	0000000000,...	HLR (H...		TCAP	486		XUDT (Message reassembled) XUD...
25935	2017-04-04..	2344	3434	1111111111	VLR (Visitor Location Re...	0000000000	HLR (H...		GSM MAP	198	00000043	invoke purgeMS
25937	2017-04-04..	2345	3433	1111111111	MSC (Mobile Switching Ce...	0000000000	MSC (M...		GSM MAP	186	00000044	invoke prepareHandover
25941	2017-04-04..	2345	3433	1111111111	MSC (Mobile Switching Ce...	0000000000	MSC (M...		GSM MAP	182	00000045	invoke prepareSubsequentHandov...
25943	2017-04-04..	2345	3433	1111111111	HLR (Home Location Regis...	0000000000	VLR (V...		GSM MAP	190	00000046	invoke provideSubscriberInfo

▶ Frame 25903: 170 bytes on wire (1360 bits), 170 bytes captured (1360 bits) on interface 0
 ▶ Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
 ▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
 ▶ Stream Control Transmission Protocol, Src Port: 2349 (2349), Dst Port: 3439 (3439)
 ▶ MTP 3 User Adaptation Layer
 ▶ Signalling Connection Control Part
 ▶ [2 Message fragments (264 bytes): #25899(229), #25903(35)]
 Transaction Capabilities Application Part

0000	00 00 00 00 00 00 00 00	7b 48 20 27 3d 29 b5 f6{
0010	63 ae ad 1b 57 ea dd 52	4f ce a7 de 55 e5 7e b3	c...W..R 0
0020	8d cf ed be a5 83 2b b8	de 32 bd ed 07 be 80 6a+.?
0030	b6 ad 3e 31 62 20 9b 7f	50 8e 27 97 c7 e8 0c 9f	..>1b...P
0040	ff d3 c2 9b 4f a3 83 0b	7d 3f c4 03 3d 34 3b 73	...0...}
0050	45 ac 96 44 0f b4 55 7b	db 3a 7f 3d 5c c6 a3 ed	E...U{
0060	82 ab 38 30 a9 46 2d 6b	6c 48 8d e6 7e cd 3e 26	..80.F-k l
0070	6f d0 f0 f5 ee 1a 31 f6	cc 11 81 94 d3 2c 56 ff	0.....1.
0080	03 09 5f ee 3d 09 0c e7	9c 5a 60 81 01 77 47 82	...=1...?
0090	f5 e6 47 98 1d c1 f1 61	d8 c7 ac fa f3 5d d0 bf	.6...a
00a0	ce 40 fb d6 1d 12 b3 fc	80 e0 c0 88 ca a1 93 41	@.....

Figure 6.1.10 - SS7 encryption example

6.1.11 SCCP UDT / XUDT

On previous figure has been seen XUDT messages.

The XUDT is used instead of UDT if the payload size has increased and reached the maximum limit of UDT message.

After decryption on the other end the message are again reconstructed into UDT message.

This is the limitation of the current solution, that the SCCP provider have to support and route the XUDT messages.

6.1.12 SS7 Encryption Autodiscovery

Firewall feature to enable encryption autodiscovery. The autodiscovery should enable easier initial key management to receive the public key over the signalling.

1. The FW #1 will send MAP Invoke (New OpCode 99) for destinations with no known Public Key
2. If there is FW #2 in path, it process the Invoke and send Result (including GT prefix and Public Key)
3. FW #1 config is updated with gathered public keys

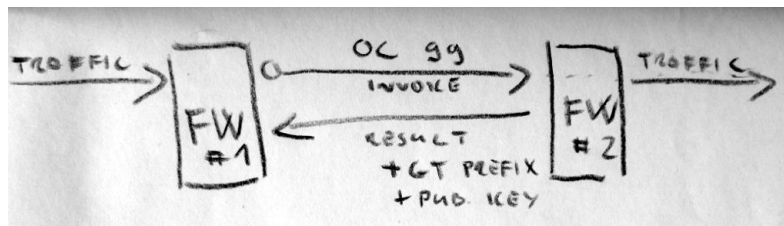


Figure 6.1.12 - SS7 Firewall autodiscovery

Limitation is that during the initial autodiscovery the remote party is not authenticated. If the remote key has expired or has been changed, the public key stored on FW#1 instance can be deleted to re-trigger the autodiscovery again. But during this process the above security aspect should be again considered and manual key management should be understood as more secure.

6.1.13 SS7 Encryption Flow - autodiscovery

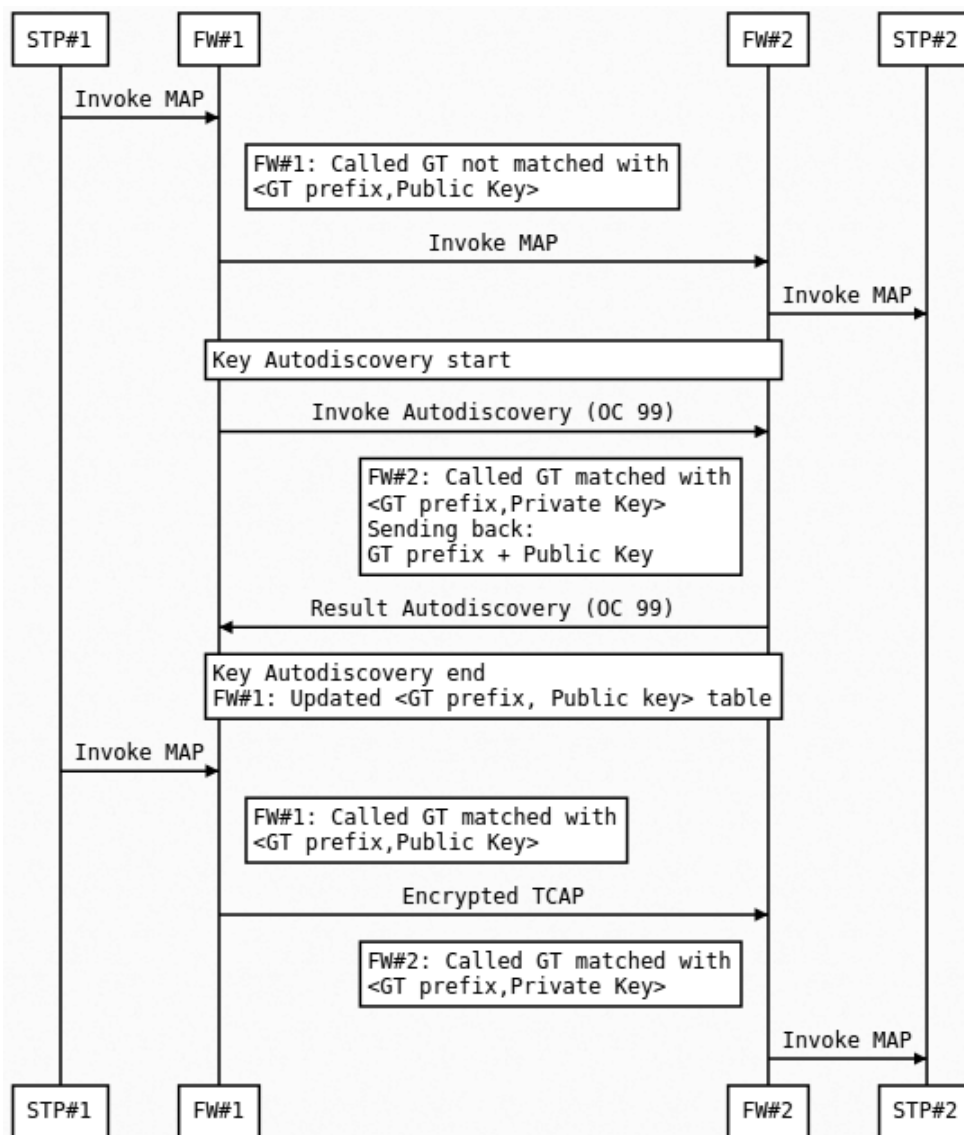


Figure 6.1.13 - SS7 Firewall autodiscovery flow

6.1.14 SS7 Signature

For every TCAP Begin, the second Invoke is added containing the TCAP signature.

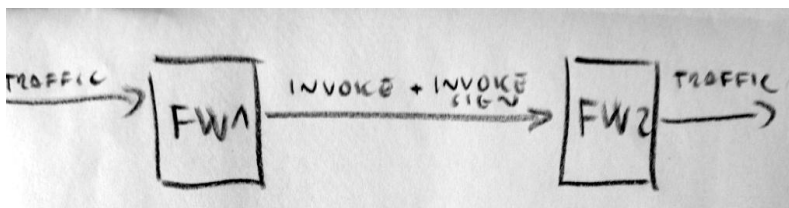


Figure 6.1.14 - SS7 signature

6.1.15 SS7 Signature Algorithm

1. Only TCAP Begins are signed
2. Check if the TCAP already contains some TCAP Invoke signature component. If not, sign it.
3. TCAP signature component will contains:
 - a. Version
 - b. Timestamp
 - c. Signature
4. Signature is calculated:
 - a. String dataToSign = calling_gt_digits + called_gt_digits + timestamp + tcap_layer
 - b. String tcap_layer = base64(tcap_component_1) + ... + base64(tcap_component_N);
 - c. String dataToSign is then hashed (currently in code SHA256WithRSA is used)

6.1.16 SS7 Signature Example

The screenshot displays a list of SS7 signaling messages on the left and their corresponding hex dump on the right. The messages are categorized by protocol, such as MSC (Mobile Switching Center), HLR (Home Location Register), and VLR (Visitor Location Register). The hex dump shows the raw data of the messages, with some fields highlighted in yellow, indicating the signature component.

Figure 6.1.16 - SS7 signature example

6.1.17 DNAT to Honeytrap

After detecting an attack the FW will perform DNAT for a defined time period for the attacker's GT.

By this approach the signalling honeypot can process the messages and send back the fake results. Additionally most time the attacker performs first the vulnerability probing of the target network and only if the network is vulnerable than conducts the real attack. Honeytrap could also enable to capture such latter messages and multistage attacks performed by attacker.

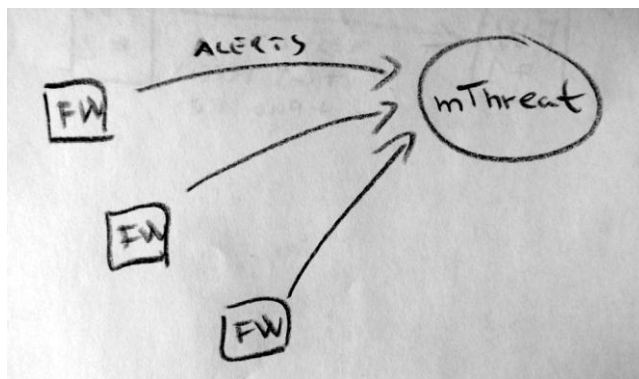


Figure 6.1.19 - SigFW reporting alerts to mThreat

6.1.20 mThreat Example

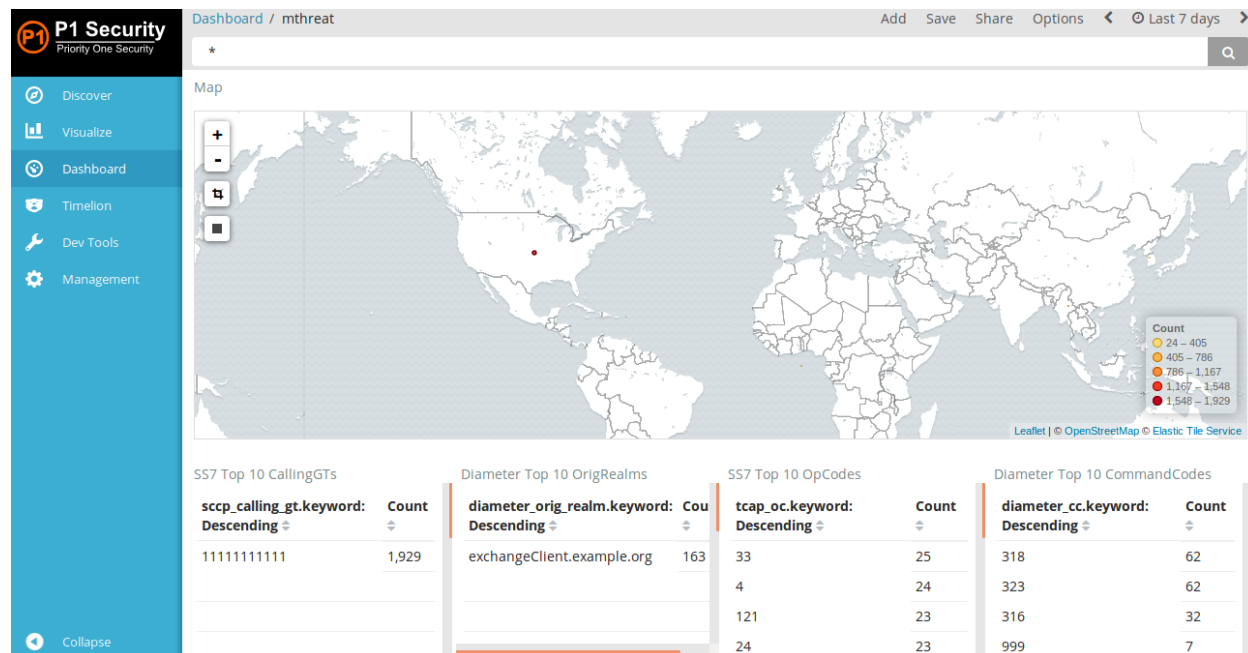


Figure 6.1.20 - mThreat UI using Kibana and Elasticsearch example

6.2 Open Diameter Firewall

The similar functionality has been developed for the Diameter protocol for 4G/LTE networks. The similar capabilities are included.

6.2.1 Architecture

Frames are forwarded on SCTP layer.

Filtering is possible up to application layer (Diameter layer).

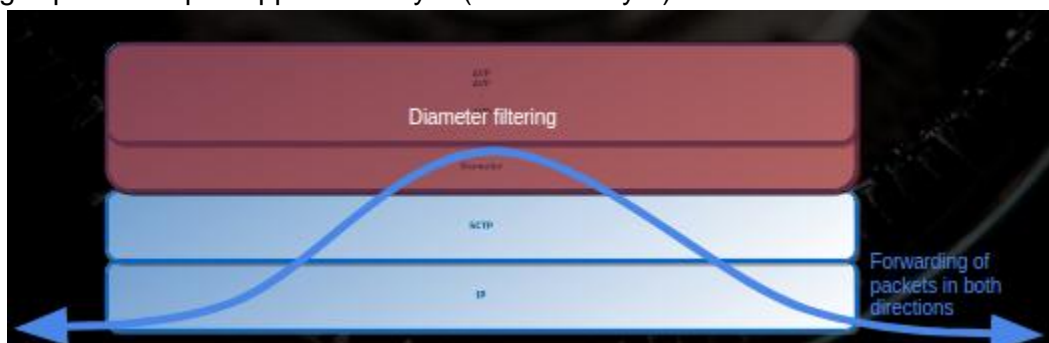


Figure 6.2.1a - Diameter Firewall decoding

Firewall is acting like SCTP server and SCTP client, without having Diameter Address. The Diameter CER, DWR, DPR or forwarded.

Below is illustrated direction of establishing links and associations.



Figure 6.2.1b - Diameter Firewall connections

6.2.2 Deployment

Possible deployment can be loopback on DRA towards the FW. Also other deployment scenarios could be FW deployed directly on the link or FW just protecting single network element.

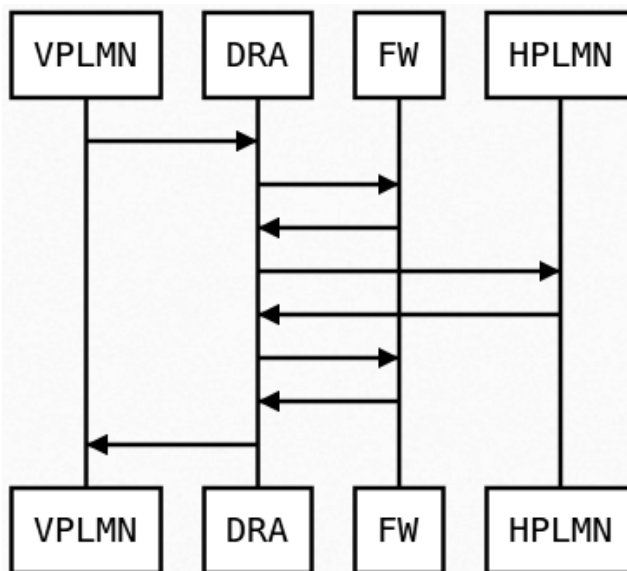


Figure 6.2.2 - Diameter Firewall deployment

6.2.3 Diameter Encryption Flow

The below figure illustrates the encryption flow. The principles are similar to SS7 FW, with the difference that the encryption is on AVP level in Diameter protocol.

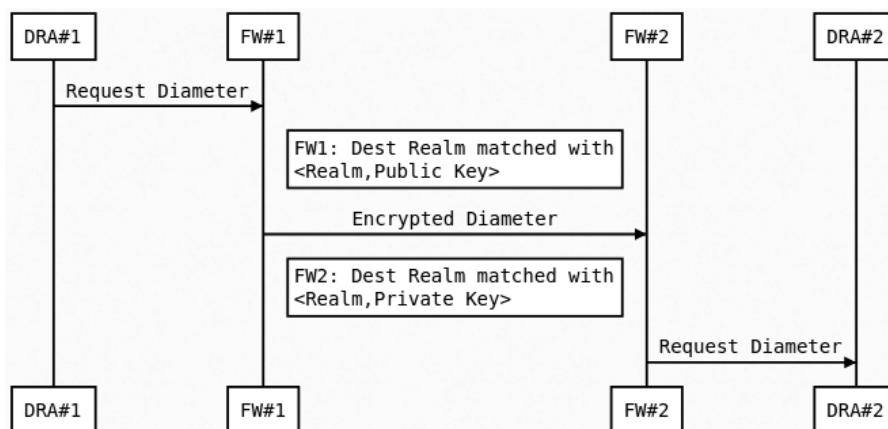


Figure 6.2.3 - Diameter Encryption Flow

6.2.4 Diameter Encryption Algorithm

1. Encrypted is the on the Diameter AVP level
2. Not encrypted AVPs are the AVPs required for IPX carriers (mainly host, realm, route)
3. Encrypted is the following payload for every AVP:
 - a. version (4 bytes)
 - b. encrypted(timestamp (4 bytes) + avp_bytes) // If the key is short the multiple similar blocks are created
4. Encryption algorithm should be mapped with version. Currently in the code only RSA/ECB/PKCS1Padding is used
5. Timestamp is verified after decryption to prevent replay attacks

6.2.5 Diameter Encryption Example

```

147 62.938384208 127.0.0.1 127.0.0.1 DIAMET... 462 cmd=3GPP-Notify Request(323) flags=R--- appl=3GPP S6a/S6d(1677251) h2h=4a49277c e2e=6f500011 |
148 62.931295117 127.0.0.1 127.0.0.1 DIAMET... 426 SACK cmd=3GPP-Notify Answer(323) flags=-E- appl=3GPP S6a/S6d(1677251) h2h=4a49277c e2e=6f500011 ...
151 62.939193161 127.0.0.1 127.0.0.1 DIAMET... 1334 SACK cmd=3GPP-Notify Answer(323) flags=-E- appl=3GPP S6a/S6d(1677251) h2h=4a49277c e2e=6f500011 ...
155 62.957918437 127.0.0.1 127.0.0.1 DIAMET... 426 SACK cmd=3GPP-Notify Answer(323) flags=-E- appl=3GPP S6a/S6d(1677251) h2h=4a49277c e2e=6f500011 ...
156 62.957935581 127.0.0.1 127.0.0.1 DIAMET... 410 SACK cmd=3GPP-Update-Location Request(316) flags=R--- appl=3GPP S6a/S6d(1677251) h2h=4a49277d e2e=...
159 62.960240472 127.0.0.1 127.0.0.1 DIAMET... 419 SACK cmd=3GPP-Update-Location Request(316) flags=R--- appl=3GPP S6a/S6d(1677251) h2h=4a49277d e2e=...
165 62.986540937 127.0.0.1 127.0.0.1 DIAMET... 350 SACK cmd=3GPP-Update-Location Answer(316) flags=-E- appl=3GPP S6a/S6d(1677251) h2h=4a49277d e2e=...
168 62.992970961 127.0.0.1 127.0.0.1 DIAMET... 1418 SACK cmd=3GPP-Update-Location Answer(316) flags=-E- appl=3GPP S6a/S6d(1677251) h2h=4a49277d e2e=...
173 63.009762391 127.0.0.1 127.0.0.1 DIAMET... 350 SACK cmd=3GPP-Update-Location Answer(316) flags=-E- appl=3GPP S6a/S6d(1677251) h2h=4a49277d e2e=...
186 62.995232305 127.0.0.1 127.0.0.1 DIAMET... 142 cmd=Device-Watchdog Request(280) flags=R--- appl=Diameter Common Messages(0) h2h=4a4927fc2 e2e=70b00...
187 62.996785046 127.0.0.1 127.0.0.1 DIAMET... 142 cmd=Device-Watchdog Request(280) flags=R--- appl=Diameter Common Messages(0) h2h=4a4927fc2 e2e=70b00...
188 62.998244255 127.0.0.1 127.0.0.1 DIAMET... 142 cmd=Device-Watchdog Request(280) flags=R--- appl=Diameter Common Messages(0) h2h=4a4927fc2 e2e=70b00...
189 62.999627596 127.0.0.1 127.0.0.1 DIAMET... 166 SACK cmd=Device-Watchdog Answer(280) flags=---- appl=Diameter Common Messages(0) h2h=4a4927fc2 e2e=...
190 63.000873609 127.0.0.1 127.0.0.1 DIAMET... 166 SACK cmd=Device-Watchdog Answer(280) flags=---- appl=Diameter Common Messages(0) h2h=4a4927fc2 e2e=...
191 63.002105486 127.0.0.1 127.0.0.1 DIAMET... 166 SACK cmd=Device-Watchdog Answer(280) flags=---- appl=Diameter Common Messages(0) h2h=4a4927fc2 e2e=...

▶ Flags: 0x80, Request
Command Code: 316 3GPP-Update-Location
ApplicationId: 3GPP S6a/S6d (1677251)
Hop-by-Hop Identifier: 0x4a49277d
End-to-End Identifier: 0x6f500014
[Answer In: 168]
▶ AVP: Session-Id(263) l=48 f=-M- val=CretedByDiameterLiveClient;1493747508867
▶ AVP: Unknown(1100) l=136 f=--- val=45e945a4a0758023a778a26b851e619db9c671e851e34178...
▶ AVP: Destination-Host(293) l=28 f=-M- val=aaa://127.0.0.1:3868
▶ AVP: Unknown(1100) l=136 f=--- val=3650b1097199e791b8d03f53f9eb1affbdc34cc5afdb80f...
▶ AVP: Origin-Host(264) l=59 f=-M- val=...
▶ AVP: Unknown(1100) l=136 f=--- val=90158db1699d26781fffbcfb040ec7023dabc728bf5556a...
▶ AVP: Unknown(1100) l=136 f=--- val=035f530ec93c18f991225cf91b05cc5dd4e167cb6d4c463b...
▶ AVP: Unknown(1100) l=136 f=--- val=013d585de64a1ef1a068386d375827cb2de27e1c720dbf51...
▶ AVP: Unknown(1100) l=136 f=--- val=5a1f8a2df193160d5fe39c3231630e09c2447ff00b879fd9...
▶ AVP: Unknown(1100) l=136 f=--- val=6510ea0a4bbd1c8ed21c50fee483110cb141c2f58f3d1a98...
▶ AVP: Destination-Realm(283) l=28 f=-M- val=exchange.example.org
▶ AVP: Origin-Realm(296) l=34 f=-M- val=exchangeClient.example.org
▶ AVP: Unknown(1100) l=264 f=--- val=7290d536001fccc9f95fe43055cb5f49ef273f3ce8a43b04...
    
```

Figure 6.2.5 - Diameter Encryption Example

6.2.6 Diameter Encryption Autodiscovery

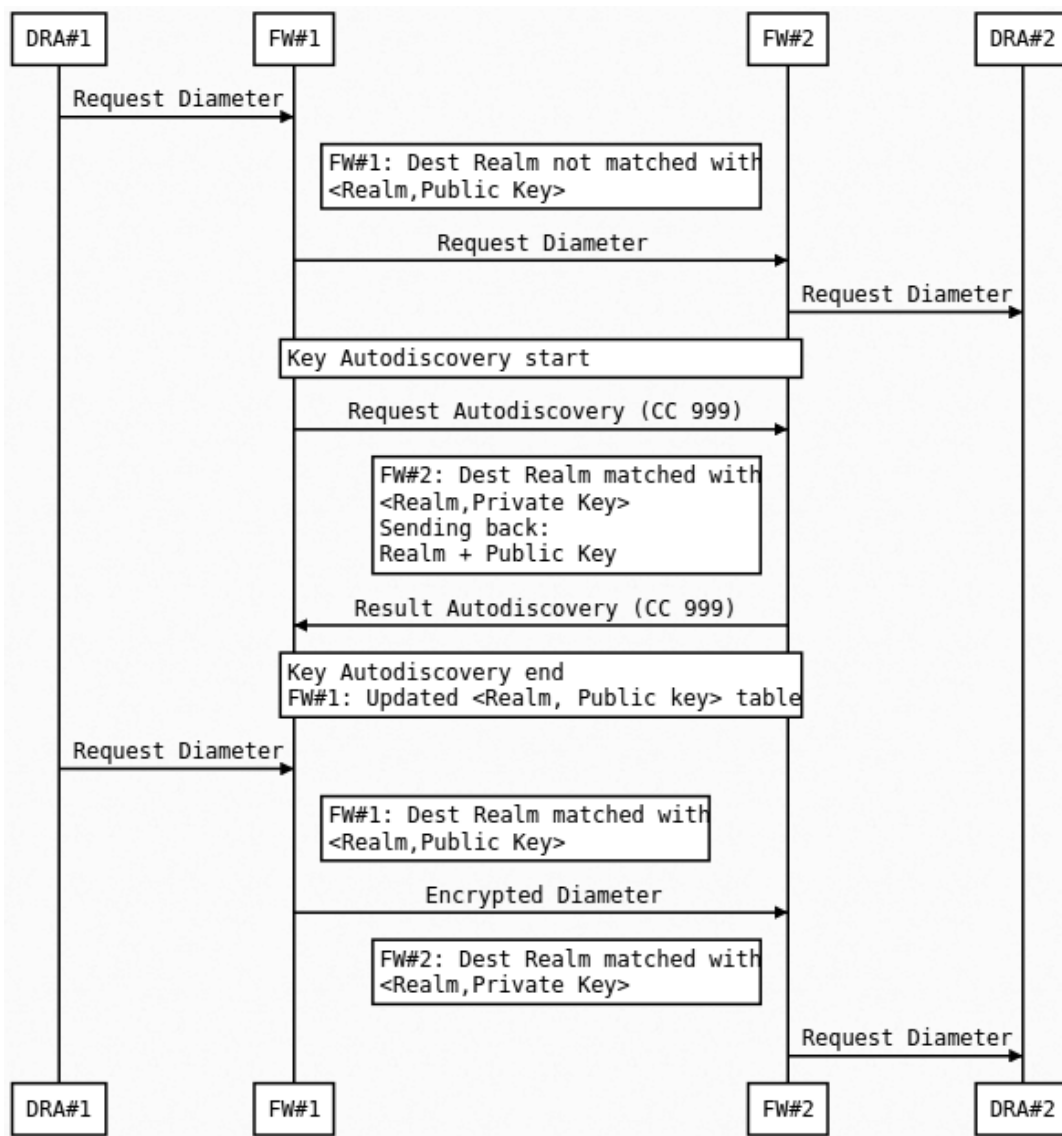


Figure 6.2.6 - Diameter Encryption Flow

6.2.7 Diameter Signature Algorithm

1. Only Diameter Requests are signed
2. Check if the Diameter message already contains some Diameter signature AVP. If not, sign it.
3. Diameter signature is Octet String of the following:
 - a. version (4 bytes)
 - b. timestamp (4 bytes)

- c. signature
4. Signature is calculated:
 - a. String dataToSign = getApplicationId + ":" + CommandCode + ":" + EndToEndIdentifier + ":" + timestamp + diameter_layer;
 - b. String diameter_layer = SORT_STRINGS(base64(avp_1) + ... + base64(avp_N)); // for AVP != RECORD_ROUTE
 - c. String dataToSign is then hashed (currently in code SHA256WithRSA is used)

6.2.8 Diameter Signature

```

368 258.634196162 127.0.0.1 127.0.0.1 DIAMET... 330 cmd=3GPP-Authentication-Information Answer(318) flags=-E- appl=3GPP S6a/S6d(16777251) h2h=4a49278...
379 259.889738089 127.0.0.1 127.0.0.1 DIAMET... 462 cmd=3GPP-Notify Request(323) flags=R--- appl=3GPP S6a/S6d(16777251) h2h=4a492785 e2e=6f500031 |
374 259.896722807 127.0.0.1 127.0.0.1 DIAMET... 602 cmd=3GPP-Notify Request(323) flags=R--- appl=3GPP S6a/S6d(16777251) h2h=4a492785 e2e=6f500031 |
378 259.902956863 127.0.0.1 127.0.0.1 DIAMET... 462 cmd=3GPP-Notify Request(323) flags=R--- appl=3GPP S6a/S6d(16777251) h2h=4a492785 e2e=6f500031 |
379 259.903929063 127.0.0.1 127.0.0.1 DIAMET... 426 SACK cmd=3GPP-Notify Answer(323) flags=-E- appl=3GPP S6a/S6d(16777251) h2h=4a492785 e2e=6f500031 ...
382 259.909959844 127.0.0.1 127.0.0.1 DIAMET... 426 SACK cmd=3GPP-Notify Answer(323) flags=-E- appl=3GPP S6a/S6d(16777251) h2h=4a492785 e2e=6f500031 ...
385 259.915694965 127.0.0.1 127.0.0.1 DIAMET... 426 SACK cmd=3GPP-Notify Answer(323) flags=-E- appl=3GPP S6a/S6d(16777251) h2h=4a492785 e2e=6f500031 ...
386 259.921751307 127.0.0.1 127.0.0.1 DIAMET... 410 SACK cmd=3GPP-Update-Location Request(316) flags=R--- appl=3GPP S6a/S6d(16777251) h2h=4a492786 e2e...
389 259.936512885 127.0.0.1 127.0.0.1 DIAMET... 550 SACK cmd=3GPP-Update-Location Request(316) flags=R--- appl=3GPP S6a/S6d(16777251) h2h=4a492786 e2e...
392 259.936816424 127.0.0.1 127.0.0.1 DIAMET... 410 SACK cmd=3GPP-Update-Location Request(316) flags=R--- appl=3GPP S6a/S6d(16777251) h2h=4a492786 e2e...
393 259.937619346 127.0.0.1 127.0.0.1 DIAMET... 350 SACK cmd=3GPP-Update-Location Answer(316) flags=-E- appl=3GPP S6a/S6d(16777251) h2h=4a492786 e2e...
396 259.944066943 127.0.0.1 127.0.0.1 DIAMET... 350 SACK cmd=3GPP-Update-Location Answer(316) flags=-E- appl=3GPP S6a/S6d(16777251) h2h=4a492786 e2e...
399 259.948708460 127.0.0.1 127.0.0.1 DIAMET... 350 SACK cmd=3GPP-Update-Location Answer(316) flags=-E- appl=3GPP S6a/S6d(16777251) h2h=4a492786 e2e...
-----
Flags: 0x80, Request
Command Code: 316 3GPP-Update-Location
ApplicationId: 3GPP S6a/S6d (16777251)
Hop-by-Hop Identifier: 0x4a492786
End-to-End Identifier: 0x6f500035
[Answer In: 396]
AVP: Session-Id(263) l=48 f=-M- val=CrededByDiameterLiveClient;1493747705878
AVP: Auth-Application-Id(258) l=12 f=-M- val=3GPP S6a/S6d (16777251)
AVP: Destination-Host(293) l=28 f=-M- val=aaa://127.0.0.1:3800
AVP: Auth-Session-State(277) l=12 f=-M- val=NO_STATE_MAINTAINED (1)
AVP: Origin-Host(264) l=59 f=-M- val=[REDACTED]
AVP: User-Name(1) l=23 f=-M- val=[REDACTED]
AVP: ULR-Flags(1405) l=16 f=VM- vnd=IGPP val=34
AVP: Visited-PLMN-Id([REDACTED])
AVP: RAT-Type(1032) l=16 f=VM- vnd=IGPP val=EUTRAN (1004)
AVP: UE-SRVCC-Capability(1615) l=16 f=V- vnd=IGPP val=UE-SRVCC-NOT-SUPPORTED (0)
AVP: Destination-Realm(283) l=28 f=-M- val=exchange.example.org
AVP: Origin-Realm(296) l=34 f=-M- val=exchangeClient.example.org
AVP: Unknown(1000) l=140 f=--- val=7a57cfc29a83b15d1b4e56b0fde3e185b1264dddf85a6f6be5...
-----
AVP Code: 1000 Unknown
AVP Flags: 0x00
AVP Length: 140
Value: 7a57cfc29a83b15d1b4e56b0fde3e185b1264dddf85a6f6be5...
    
```

Figure 6.2.8 - Diameter Signature Example

7 Closing remarks

The currently released version of the SigFW should be understood as a research project/reference implementation and not as operational ready solution. The work as well as the filtering capabilities and the confidentiality/integrity protection schemes should be evolved further to find solution which is addressing both operational and security needs.

By this open-source approach we hope we can help to improve the SS7/Diameter security and this project adoption can also help to reveal the source and origin of these SS7/Diameter attacks. The SS7/Diameter security is affecting all mobile users worldwide. We believe that the open source is the right way for the security and should be adopted also in telecom field.

As it is seen, the current work has been created thanks to Telestax open-source signalling stack and Wireshark, Elastic projects.

7.1 VM architecture

VM is available for download at <https://github.com/P1sec/SigFW/wiki/VM>

Ubuntu Server

- eth0 management

- eth1 signalling (possible to configure the firewall here)

- eth2 passive signalling (used by tshark to feed the VM in passive mode)

Installed ElasticSearch, Kibana

All firewall modules as systemd services

On localhost running SS7ClientLiveInput -> SS7Firewall -> SS7Server

pcap -> tshark -> SS7ClientLiveInput

eth2 -> tshark -> SS7ClientLiveInput

eth2 -> tshark -> curl -> ElasticSearch -> Kibana

7.2 SigFW use cases

The below figures illustrates high-level use cases of the SigFW. The figures outlines the use of SigFW for standard filtering capabilities, the confidentiality and integrity protection of the signalling and also the DNAT towards the honeypot.

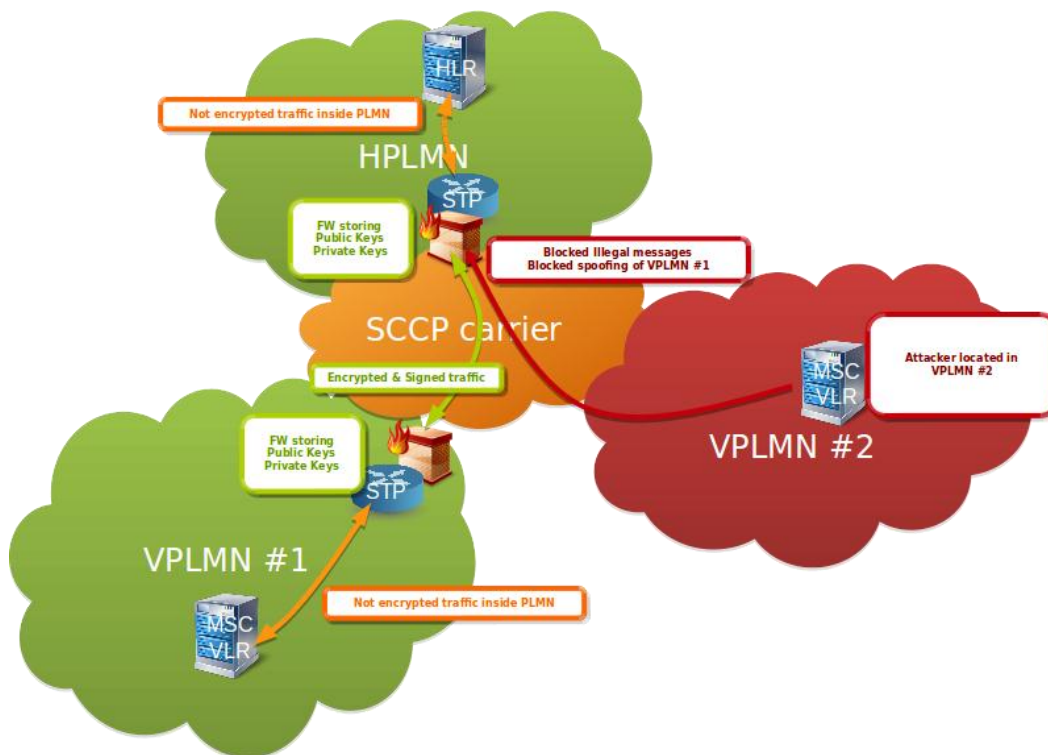


Figure 7.2a - SigFW filtering and confidentiality and integrity protection of signalling

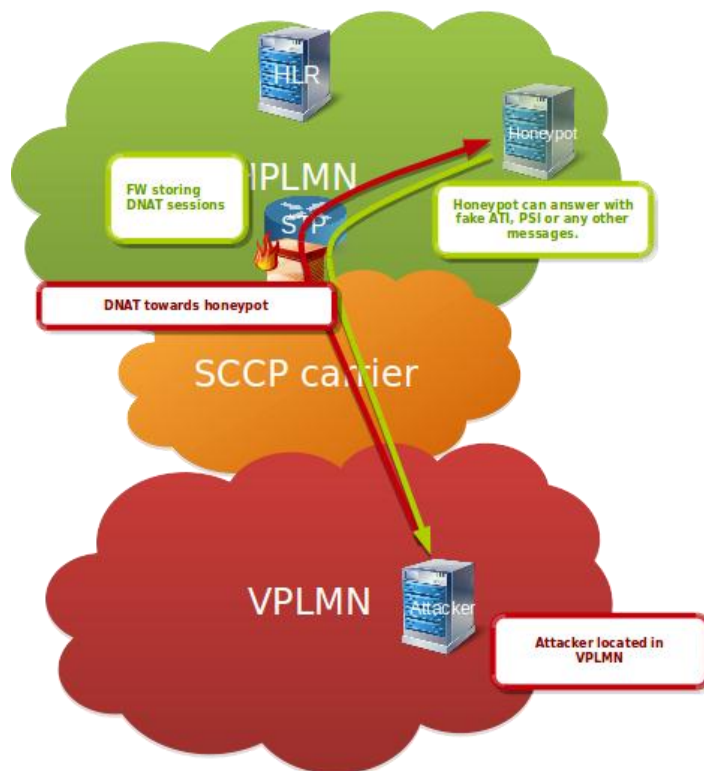


Figure 7.2b - SigFW forwarding the attacker to honeypot

8 Related Open Source Contribution

8.1 Tshark to Elasticsearch export and security monitoring with Kibana

We would like also to highlight the contributed patch to Wireshark project. This features are used in the SigFW VM.

Wireshark is capable to export decoded packets in json format. Additionally the tshark can export json format and also elasticsearch json which can be directly imported into elasticsearch cluster.

This could enable to use tshark as signalling probe and perform signalling monitoring as illustrated on the following figure.

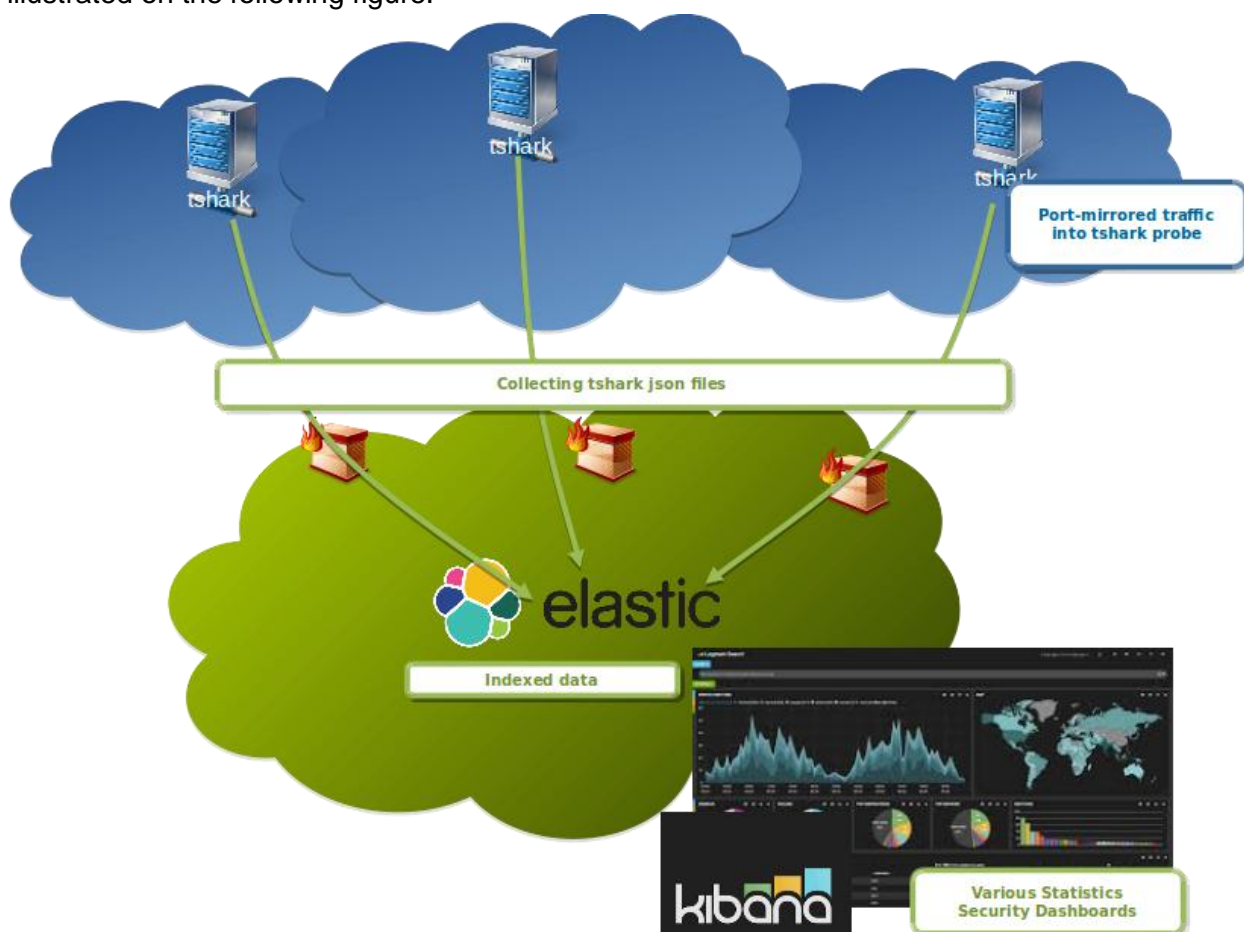


Figure 8.1a - tshark with Elasticsearch

The monitoring could be for network functionality or troubleshooting reasons but also could be used for security monitoring. The light solution could be just using Kibana dashboards for security monitoring.

The following figures illustrates signalling monitoring in Kibana and simple Dashboards.

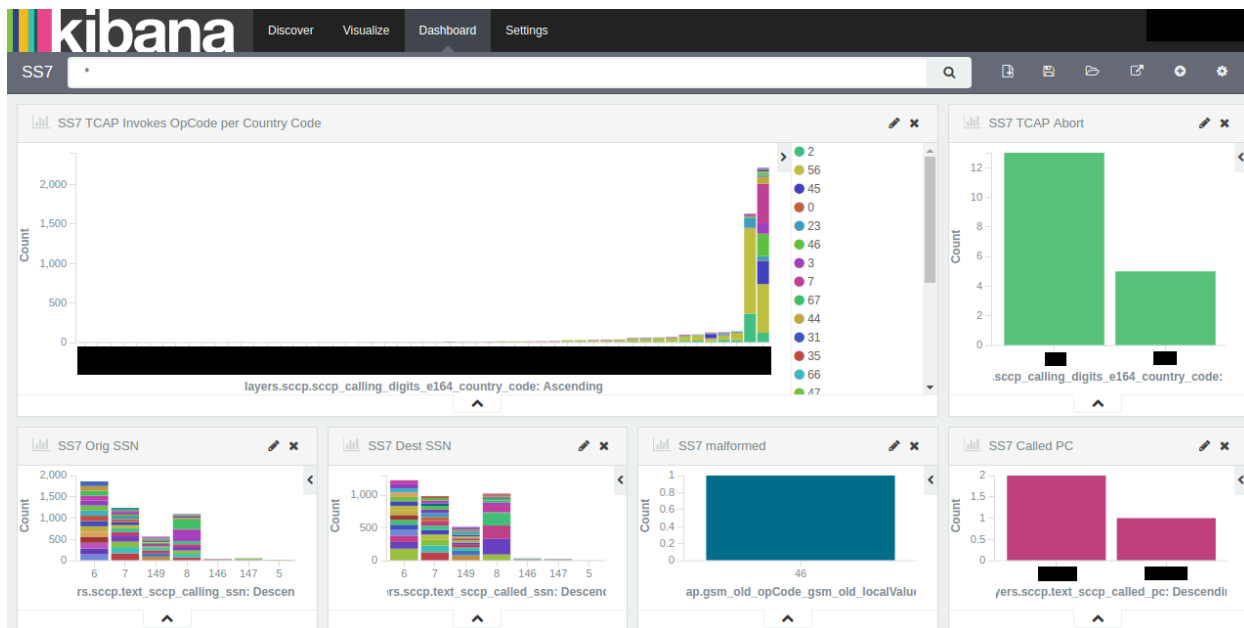


Figure 8.1b - tshark with Kibana example 1

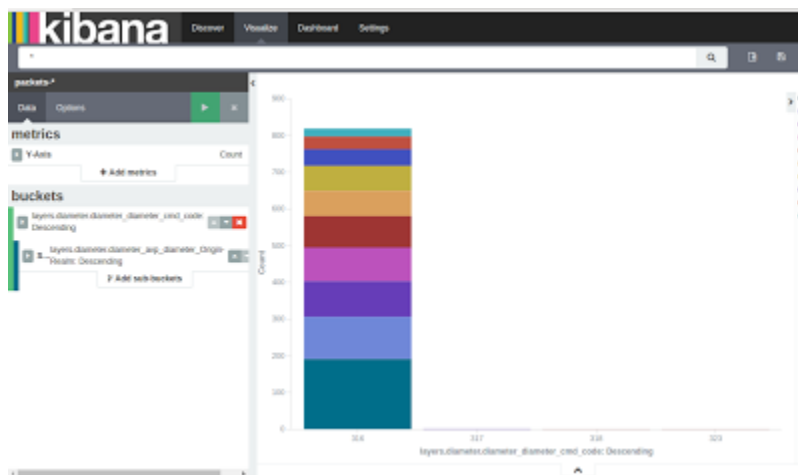


Figure 8.1c - tshark with Kibana example 2

More details are described on https://sites.google.com/site/h21lab/tools/tshark_elasticsearch.

9 References and Acknowledgement

- [1] GSMA workgroup collaboration (FS.11, FS.19, FS.20 ...)
- [2] 3GPP standardization on signaling (TS 29.002, TS 22.078, TS 29.204, TS 33.204, TS 29.272, TS 29.060, TS 29.274, ...)
- [3] P1 Security SS7 & Diameter security deployment (<http://www.p1sec.com>)
- [4] P1 Labs SS7map and security research (<http://ss7map.p1sec.com/>, <http://labs.p1sec.com/>)
- [5] H21 lab blogs, published tools, research (<https://sites.google.com/site/h21lab/>)

International conferences presentations:

- [6] SCTPscan - Finding entry points to SS7 Networks & Telecommunication Backbones, Philippe Langlois, Black Hat 2006
- [7] Locating Mobile Phones using SS7, Tobias Engel, CCC 2009
- [8] SCCP hacking, attacking the SS7 & SIGTRAN applications one step further and mapping the phone system, Philippe Langlois, CCC 2009
- [9] SCCP hacking Attacking the SS7 & SIGTRAN and Mapping the Phone System, Philippe Langlois, 2010
- [10] Getting in the SS7 kingdom: hard technology and disturbingly easy hacks to get entry points in the walled garden, Philippe Langlois, Hackito Ergo Sum 2010
- [11] Hack In The Box 2012: A 15 Year Perspective on Why Telcos Keep Getting Hacked, Philippe Langlois, Emmanuel Gadaix, Hack In The Box 2012
- [12] Worldwide attacks on SS7/SIGTRAN network, Pierre-Olivier Vauboin, Alexandre De Oliveira, P1 Security, Hackito Ergo Sum 2014
- [13] Mobile self--defense, Karsten Nohl, SR Labs, CCC 2014
- [14] Securing the SS7 Interconnect Tobias Engel, Troopers 2015
- [15] SS7: Locate. Track. Manipulate, Tobias Engel, CCC 2015
- [16] About SS7 (Signalling System Seven) in 60 Minutes, SR Labs, 2016

10 Annex

10.1 SS7FW VM readme

Signalling firewall and monitoring appliance

Interfaces:

```
enp0s3 - management (SSH, Web)
enp0s8 - signalling (SS7FW could be reconfigured here)
enp0s9 - passive signalling (port-mirrored traffic)
```

To access Kibana:

```
http://<host>:5601/
```

To access API

```
https://<host>:8443/ss7fw_api/1.0/get_status
```

To check if services are running:

```
sudo service tshark_to_ss7fw status
sudo service tshark_to_ek status
sudo service ss7fw status
sudo service ss7server status
sudo service ss7client status
```

To replay the pcap on passive interface:

```
sudo tcpreplay --intf1=enp0s9 sigtran.pcap
```

Description:

By default the SS7FW is in passive mode.
Tshark is capturing traffic on enp0s9 and pushing into ElasticSearch.
Second instance of tshark is pushing capture into named pipe of SS7FW.
The SS7FW consist of ss7client, ss7firewall, ss7server. ss7client replay the captured traffic from enp0s9 towards ss7firewall and ss7server on localhost.

SS7FW is located in /opt/ss7fw/

Before first run or if the IP has changed, modify /etc/kibana/kibana.yml"

To access logs:

```
tail -f /opt/ss7fw/ss7fw/ss7fw.ss7fw-core_jar_1.0.0-SNAPSHOT/ss7fw.log
```

10.2 SS7FW configuration example

```
{
  "operator_configuration": {
    "Home_GT_prefixes_comment": "# Identification of HPLMN network, used to identify incoming and outgoing
traffic of HPLMN",
    "Home_GT_prefixes": [
      "0"
    ],
    "Home_IMSI_prefixes_comment": "# Identification Home IMSI range for HPLMN network, used to identify home
subscribers",
```

```
"Home_IMSI_prefixes": [
  "111111"
],
"sigfw_configuration": {
  "ss7fw_configuration_comment": "# Signalling Firewall configuration. Because of dynamic updates, the
sigfw.json.last is periodically created on filesystem.",

  "sctp_comment": "# Sctp configuration part of Signalling Firewall",
  "sctp": {
    "sctp_management_name": "sctp_mgmt",
    "sctp_server": [
      {
        "server_name": "sctp_server",
        "host_address": "127.0.0.1",
        "port": "3433"
      }
    ],
    "sctp_server_association": [
      {
        "peer_address": "127.0.0.1",
        "peer_port": "2345",
        "server_name": "sctp_server",
        "assoc_name": "sctp_from_client_to_firewall"
      }
    ],
    "sctp_association": [
      {
        "host_address": "127.0.0.1",
        "host_port": "2344",
        "peer_address": "127.0.0.1",
        "peer_port": "3434",
        "assoc_name": "sctp_from_firewall_to_server"
      }
    ]
  },
  "m3ua": {
    "m3ua_comment": "# M3UA configuration part of Signalling Firewall",

    "m3ua_server": {
      "m3ua_management_name": "m3ua_server_mgmt",
      "as_name": "RAS1",
      "asp_name": "RASP1",
      "sctp_assoc_name": "sctp_from_client_to_firewall",
      "remote_pc": ["1"]
    },
    "m3ua_client": {
      "m3ua_management_name": "m3ua_client_mgmt",
      "as_name": "AS1",
      "asp_name": "ASP1",
      "sctp_assoc_name": "sctp_from_firewall_to_server",
      "remote_pc": ["2"]
    }
  },
  "firewall_rules": {
    "firewall_rules_comment": "# Firewall filtering rules configuration",

    "firewall_policy_comment": "# Allowed value is one from: DROP_SILENTLY, DROP_WITH_SCCP_ERROR,
DNAT_TO_HONEYPOT, ALLOW",
    "firewall_policy": "DROP_WITH_SCCP_ERROR",

    "sccp": {
      "sccp_comment": "# SCCP firewall rules",
      "calling_gt_whitelist": [
        "4*"
      ],
      "calling_gt_blacklist": [
        "1000000000",
        "222*"
      ]
    },
    "tcap": {
      "tcap_comment": "# TCAP Cat1 firewall rules",
      "oc_blacklist": [
        "5",
        "6",
        "9",
        "16",
        "20",
        "21",

```

```
        "22",
        "24",
        "25",
        "26",
        "27",
        "28",
        "29",
        "30",
        "31",
        "32",
        "33",
        "34",
        "35",
        "39",
        "40",
        "41",
        "42",
        "43",
        "50",
        "51",
        "52",
        "55",
        "58",
        "62",
        "65",
        "68",
        "69",
        "71",
        "72",
        "76",
        "77",
        "78",
        "79",
        "80",
        "81",
        "82",
        "83",
        "84",
        "85",
        "86",
        "109",
        "110",
        "111",
        "112",
        "113",
        "114",
        "115",
        "116",
        "117",
        "118",
        "119",
        "120",
        "121",
        "122",
        "123",
        "124",
        "125",
        "126"
    ]
},
"map": {
    "map_comment": "# MAP Cat2 firewall rules",
    "cat2_oc_blacklist": [
        "3",
        "4",
        "7",
        "8",
        "70"
    ]
},
"lua": {
    "lua_comment": "# LUA Blacklist firewall rules. Currently supported LUA variables are:
sccp_calling_gt, sccp_called_gt, tcap_oc, tcap_ac, tcap_tag, map_imsi, map_msisdn",
    "blacklist_rules": [
        "sccp_called_gt == '2222222222'",
        "sccp_calling_gt == '1111111111' and tcap_oc == '59'"
    ]
},
"ids": {
```

```
    "ids_comment": "# IDS API. After evaluating internal firewall rules, the external IDS
system can be used to check message (e.g. Cat3). If not required remove this ids json block from config.",
    "ids_api_type_comment": "# Type of connector. Currently supported only REST",
    "ids_api_type": "REST",
    "ids_servers": [
        {
            "host":
"https://localhost:8443/ss7fw_api/1.0/eval_sccp_message_in_ids",
            "username": "user",
            "password": "password"
        }
    ],
    "mthreat": {
        "mthreat_comment": "# mThreat API. If the message matches internal firewall or IDS rules,
then the firewall can report the event in anonymized way to mThreat. If not required remove this mthreat json block from
config.",
        "mthreat_api_type_comment": "# Type of connector. Currently supported only REST",
        "mthreat_api_type": "REST",
        "mthreat_salt_comment": "# Change the salt value for unique anonymization",
        "mthreat_salt": "XVm4AoKrkciscgEcX",
        "mthreat_servers": [
            {
                "host":
"https://51.15.148.211:8444/mthreat_api/1.0/send_ss7_alert_to_mthreat",
                "username": "contact@plsec.com",
                "password": "contact@plsec.com"
            }
        ]
    },
    "honeypot": {
        "honeypot_comment": "# Honeypot configuration. Only used if firewall policy is
DNAT_TO_HONEYPOT",
        "sccp_gt_comment": "# The firewall after detecting the message will perform DNAT to the
following GT.",
        "sccp_gt": "333333333333",
        "dnat_session_expiration_timeout_comment": "# After matching the firewall or IDS rules,
the firewall will apply DNAT for calling GT for the defined number of seconds",
        "dnat_session_expiration_timeout": "30"
    },
    "encryption_rules": {
        "encryption_rules_comment": "# TCAP encryption. NTP synchronization of FW instance is required to
work this properly. If autodiscovery is enabled the public keys are added dynamically. Public and private keys are Base64
encoded.",
        "called_gt_encryption_comment": "# Should include json block with {called_gt, public_key}. For
example of config see sigfw_1.json or sigfw_2.json.",
        "called_gt_encryption": [
        ],
        "called_gt_decryption_comment": "# Should include json block with {called_gt, public_key, private}.
For example of config see sigfw_1.json or sigfw_2.json.",
        "called_gt_decryption": [
        ],
        "autodiscovery_comment": "# When enabled the Firewall will try to retrieve public key for unknown
destinations by sending MAP Invoke with OpCode 99.",
        "autodiscovery": "true"
    },
    "signature_rules": {
        "signature_rules_comment": "# TCAP signing. NTP synchronization of FW instance is required to work
this properly. Public and private keys are Base64 encoded.",
        "calling_gt_verify_comment": "# Should include json block with {calling_gt, public_key}. For
example of config see sigfw_1.json or sigfw_2.json.",
        "calling_gt_verify": [
        ],
        "calling_gt_signing_comment": "# Should include json block with {calling_gt, public_key,
private_key}. For example of config see sigfw_1.json or sigfw_2.json.",
        "calling_gt_signing": [
        ]
    }
}
```


10.3 DiameterFW configuration example

```
{
  "operator_configuration": {
    "Home_IMSI_prefixes_comment": "# Identification Home IMSI range for HPLMN network, used to identify home
subscribers",
    "Home_IMSI_prefixes": [
      "111111"
    ],
    "Home_Diameter_Realm_list_comment": "Operator Diameter Internal Realm list, used to identify incoming and
outgoing traffic of HPLMN",
    "Home_Diameter_Realm_list": [
      "exchange.example.org"
    ]
  },
  "sigfw_configuration": {
    "sctp": {
      "sctp_management_name": "sctp_mgmt",
      "sctp_server": [
        {
          "server_name": "sctp_server",
          "host_address": "127.0.0.1",
          "port": "3869"
        }
      ],
      "sctp_server_association": [
        {
          "peer_address": "127.0.0.1",
          "peer_port": "13868",
          "server_name": "sctp_server",
          "assoc_name": "sctp_from_client_to_firewall"
        }
      ],
      "sctp_association": [
        {
          "host_address": "127.0.0.1",
          "host_port": "13869",
          "peer_address": "127.0.0.1",
          "peer_port": "3868",
          "assoc_name": "sctp_from_firewall_to_server"
        }
      ]
    },
    "firewall_rules": {
      "firewall_rules_comment": "# Firewall filtering rules configuration",
      "firewall_policy_comment": "# Allowed value is one from: DROP_SILENTLY, DROP_WITH_DIAMETER_ERROR,
DNAT_TO_HONEYPOT, ALLOW",
      "firewall_policy": "DNAT_TO_HONEYPOT",
      "diameter": {
        "origin_realm_blacklist": [
          "blacklisted.example.org"
        ],
        "application_id_whitelist": [
          "0",
          "16777251"
        ],
        "command_code_blacklist": [
          "8388620",
          "8388622"
        ],
        "cat2_command_code_blacklist": [
          "317",
          "319",
          "329"
        ]
      },
      "lua": {
        "lua_comment": "# LUA Blacklist firewall rules. Currently supported LUA variables are:
diameter_orig_host, diameter_orig_realm, diameter_dest_host, diameter_dest_realm, diameter_cc, diameter_ai, diameter_imsi,
diameter_msisdn",
        "blacklist_rules": [
          "diameter_orig_realm == 'exchangeClient.example.org'",
          "diameter_orig_realm == 'exchangeClientB.example.org'"
        ]
      },
      "ids": {

```

```

        "ids_comment": "# IDS API. After evaluating internal firewall rules, the external IDS
system can be used to check message (e.g. Cat3). If not required remove this ids json block from config.",

        "ids_api_type_comment": "# Type of connector. Currently supported only REST",
        "ids_api_type": "REST",
        "ids_servers": [
            {
                "host":
"https://localhost:8443/diameterfw_api/1.0/eval_diameter_message_in_ids",
                "username": "user",
                "password": "password"
            }
        ],

        "mthreat": {
            "mthreat_comment": "# mThreat API. If the message matches internal firewall or IDS rules,
then the firewall can report the event in anonymized way to mThreat. If not required remove this mthreat json block from
config.",

            "mthreat_api_type_comment": "# Type of connector. Currently supported only REST",
            "mthreat_api_type": "REST",
            "mthreat_salt_comment": "# Change the salt value for unique anonymization",
            "mthreat_salt": "XVm4AoKrkicsgEcx",
            "mthreat_servers": [
                {
                    "host":
"https://51.15.148.211:8444/mthreat_api/1.0/send_diameter_alert_to_mthreat",
                    "username": "contact@plsec.com",
                    "password": "contact@plsec.com"
                }
            ],

            "honeypot": {
                "honeypot_comment": "# Honeypot configuration. Only used if firewall policy is
DNAT_TO_HONEYPOT",

                "diameter_host_comment": "# The firewall after detecting the message will perform DNAT to
the following Diameter address.",
                "diameter_host": "127.0.0.1",
                "diameter_realm": "honeypot.example.org",

                "dnat_session_expiration_timeout_comment": "# After matching the firewall or IDS rules,
the firewall will apply DNAT for Diameter address for the defined number of seconds",
                "dnat_session_expiration_timeout": "30"
            }
        },
        "encryption_rules": {
            "destination_realm_encryption": [
            ],
            "destination_realm_decryption": [
            ],
            "autodiscovery": "true"
        },
        "signature_rules": {
            "origin_realm_verify": [
            ],
            "origin_realm_signing": [
            ]
        }
    }
}

```

10.4 SS7FW API specification

10.4.1 Provisioning FW rules API

Title	sccp_calling_gt_blacklist_add
URL	Configurable
Method	GET
URL Params	Required @MatrixParam("gt") String gt
Data Params	gt String GlobalTitle
Success Response	SS7FW should return String "Successful"
Error Response	SS7FW should return not specified string
Sample Call	<code>https://XXX.XXX.XXX.XXX:8443/ss7fw_api/1.0/sccp_calling_gt_blacklist_add;gt=11111*</code>
Notes	@GET @Consumes("text/plain") @Produces("text/plain") @Path("sccp_calling_gt_blacklist_add") public String sccp_calling_gt_blacklist_add(@MatrixParam("gt") String gt);

Title	sccp_calling_gt_blacklist_remove
URL	Configurable
Method	GET
URL Params	Required @MatrixParam("gt") String gt
Data Params	gt String GlobalTitle
Success Response	SS7FW should return String "Successful"
Error Response	SS7FW should return not specified string
Sample Call	<code>https://XXX.XXX.XXX.XXX:8443/ss7fw_api/1.0/sccp_calling_gt_blacklist_remove;gt=11111*</code>
Notes	@GET @Consumes("text/plain") @Produces("text/plain")

	<pre>@Path("sccp_calling_gt_blacklist_remove") public String sccp_calling_gt_blacklist_remove(@MatrixParam("gt") String gt);</pre>
--	--

Title	sccp_calling_gt_blacklist_list
URL	Configurable
Method	GET
URL Params	
Data Params	
Success Response	<p>SS7FW should return String containing the SCCP GTs</p> <p>Example: 10000000000 222*</p>
Error Response	SS7FW should return empty string
Sample Call	<pre>https://XXX.XXX.XXX.XXX:8443/ss7fw_api/1.0/sccp_calling_gt_blacklis t_list</pre>
Notes	<pre>@GET @Consumes("text/plain") @Produces("text/plain") @Path("sccp_calling_gt_blacklist_list") public String sccp_calling_gt_blacklist_list();</pre>

Title	tcap_oc_blacklist_add
URL	Configurable
Method	GET
URL Params	<p>Required @MatrixParam("oc") int oc</p>
Data Params	oc

	String OpCode
Success Response	SS7FW should return String "Successful"
Error Response	SS7FW should return not specified string
Sample Call	<code>https://XXX.XXX.XXX.XXX:8443/ss7fw_api/1.0/tcap_oc_blacklist_add;oc=71</code>
Notes	<pre>@GET @Consumes("text/plain") @Produces("text/plain") @Path("tcap_oc_blacklist_add") public String tcap_oc_blacklist_add(@MatrixParam("oc") int oc);</pre>

Title	tcap_oc_blacklist_remove
URL	Configurable
Method	GET
URL Params	Required @MatrixParam("oc") int oc
Data Params	oc String OpCode
Success Response	SS7FW should return String "Successful"
Error Response	SS7FW should return not specified string
Sample Call	<code>https://XXX.XXX.XXX.XXX:8443/ss7fw_api/1.0/tcap_oc_blacklist_remove;oc=71</code>
Notes	<pre>@GET @Consumes("text/plain") @Produces("text/plain") @Path("tcap_oc_blacklist_remove") public String tcap_oc_blacklist_remove(@MatrixParam("oc") int oc);</pre>

Title	tcap_oc_blacklist_list
URL	Configurable
Method	GET

URL Params	
Data Params	
Success Response	SS7FW should return String containing the OCs Example: 109 110 111 112
Error Response	SS7FW should return empty string
Sample Call	<code>https://XXX.XXX.XXX.XXX:8443/ss7fw_api/1.0/tcap_oc_blacklist_list</code>
Notes	<pre>@GET @Consumes("text/plain") @Produces("text/plain") @Path("tcap_oc_blacklist_list") public String tcap_oc_blacklist_list();</pre>

Title	map_cat2_oc_blacklist_add
URL	Configurable
Method	GET
URL Params	Required @MatrixParam("oc") int oc
Data Params	oc String OpCode
Success Response	SS7FW should return String "Successful"
Error Response	SS7FW should return not specified string
Sample Call	<code>https://XXX.XXX.XXX.XXX:8443/ss7fw_api/1.0/map_cat2_oc_blacklist_add;oc=3</code>
Notes	<pre>@GET @Consumes("text/plain") @Produces("text/plain") @Path("map_cat2_oc_blacklist_add") public String map_cat2_oc_blacklist_add(@MatrixParam("oc") int oc);</pre>

Title	map_cat2_oc_blacklist_remove
URL	Configurable
Method	GET
URL Params	Required @MatrixParam("oc") int oc
Data Params	oc String OpCode
Success Response	SS7FW should return String "Successful"
Error Response	SS7FW should return not specified string
Sample Call	<code>https://XXX.XXX.XXX.XXX:8443/ss7fw_api/1.0/map_cat2_oc_blacklist_remove;oc=3</code>
Notes	<pre>@GET @Consumes("text/plain") @Produces("text/plain") @Path("map_cat2_oc_blacklist_remove") public String map_cat2_oc_blacklist_remove(@MatrixParam("oc") int oc);</pre>

Title	map_cat2_oc_blacklist_list
URL	Configurable
Method	GET
URL Params	
Data Params	
Success Response	SS7FW should return String containing the OCs Example: 3 4 7
Error Response	SS7FW should return empty string
Sample Call	<code>https://XXX.XXX.XXX.XXX:8443/ss7fw_api/1.0/map_cat2_oc_blacklist_list</code>

Notes	<pre>@GET @Consumes("text/plain") @Produces("text/plain") @Path("map_cat2_oc_blacklist_list") public String map_cat2_oc_blacklist_list();</pre>
-------	---

Title	map_cat2_oc_blacklist_list
URL	Configurable
Method	GET
URL Params	
Data Params	
Success Response	<p>SS7FW should return String containing the SS7FW status</p> <p>Example:</p> <pre>Jetty Server Status = STARTED Jetty Date = Date: Fri, 21 Apr 2017 07:41:24 GMT Jetty URI = = https://127.0.1.1:8443/ SCTP Associations Name = sctp_from_client_to_firewall Details = Association [name=sctp_from_client_to_firewall, associationType=SERVER, ipChannelType=SCTP, hostAddress=null, hostPort=0, peerAddress=127.0.0.1, peerPort=2345, serverName=sctp_server, extraHostAddress=[]] isStarted = true isConnected = true Name = sctp_from_firewall_to_server Details = Association [name=sctp_from_firewall_to_server, associationType=CLIENT, ipChannelType=SCTP, hostAddress=127.0.0.1, hostPort=2344, peerAddress=127.0.0.1, peerPort=3434, serverName=null, extraHostAddress=[]] isStarted = true isConnected = true SCTP Servers = {Server [name=sctp_server, started=true, hostAddress=127.0.0.1, hostPort=3433, ipChannelType=SCTP, acceptAnonymousConnections=false, maxConcurrentConnectionsCount=0, associations(anonymous does not included)={sctp_from_client_to_firewall, }, extraHostAddress=[]]} M3UA Server Name = RAS1 isConnected = true isUp = true M3UA Server Route = {1:-1:-1=org.mobicens.protocols.ss7.m3ua.impl.RouteAsImpl@36f8d985} M3UA Client Name = AS1 isConnected = true isUp = true M3UA Client Route = {2:-1:-1=org.mobicens.protocols.ss7.m3ua.impl.RouteAsImpl@48e54e27} SCCP M3UA User Parts = {0=rsrp=1 rsp-flag=0 mask=0 rsp-prohibited=false rscpp-prohibited=false, 1=rsrp=2 rsp-flag=0 mask=0 rsp-prohibited=false rscpp-prohibited=false} OS Available processors (cores): statistics 8 Free memory (bytes): 133897184 Maximum memory (bytes): 3711959040 Total memory available to JVM (bytes): 470286336 File system root: Total space (bytes): 53424500736 Free space (bytes): 38042247168</pre>

	<pre> Usable space (bytes): 35304783872 Network Display name: interfaces Name: vboxnet0 InetAddress: /fe80:0:0:0:800:27ff:fe00:0%vboxnet0 InetAddress: /192.168.56.1 Display name: wlp5s0 Name: wlp5s0 InetAddress: /fe80:0:0:0:5c77:8239:7d6b:5b16%wlp5s0 InetAddress: /192.168.1.62 Display name: lo Name: lo InetAddress: /0:0:0:0:0:0:1%lo InetAddress: /127.0.0.1 </pre>
Error Response	SS7FW should return empty string or String with error message
Sample Call	<code>https://XXX.XXX.XXX.XXX:8443/ss7fw_api/1.0/get_status</code>
Notes	<pre> @GET @Consumes("text/plain") @Produces("text/plain") @Path("get_status") public String get_status(); </pre>

10.4.2 Evaluation API

Title	eval_sccp_message_in_ids
URL	Configurable
Method	GET
URL Params	Required @MatrixParam("sccp_raw")
Data Params	sccp_raw The sccp_raw parameter is hex string of SCCP layer. (e.g. in tshark sccp_raw in json output)
Success Response	IDS should return String "1" on alert detection
Error Response	IDS should return String "0" if the message is not alert.
Sample Call	<code>https://XXX.XXX.XXX.XXX:8443/ss7fw_api/1.0/eval_sccp_message_in_ids;sccp_raw=aabbccddeeff</code>
Notes	<pre> @GET @Consumes("text/plain") @Produces("text/plain") </pre>

```
@Path("eval_sccp_message_in_ids")
public String eval_sccp_message_in_ids(@MatrixParam("sccp_raw") String
sccp_raw);
```

10.4.3 mThreat API

Title	send_ss7_alert_to_mthreat
URL	Configurable
Method	POST
URL Params	Required String alert
Data Params	<p>String alert</p> <p>The alert is JSON containing currently in API v1.0 the following variables. But in future the mThreat will support also more values and also Diameter should be supported.</p> <pre>String sccp_calling_gt = ""; String sccp_called_gt = ""; String tcap_oc = ""; String tcap_ac = ""; String map_imsi = ""; String map_msisdn = "";</pre> <p>Example of JSON</p> <pre>{"map_msisdn":"0016883DA7B9FAFD9BD9BE3E7FD4171A5058E4E3","tcap_tag":"2", "tcap_oc":"82","map_imsi":"0016883DA7B9FAFD9BD9BE3E7FD4171A5058E4E3","sc cp_calling_gt":"1111111111","tcap_ac":["0, 4, 0, 0, 1, 0, 1, 2"],"sccp_called_gt":"000000000000"}</pre>
Success Response	mThreat should return String "1"
Error Response	mThreat could return String "0" in case of some failure or other not specified string.
Sample Call	<code>https://XXX.XXX.XXX.XXX:8444/mthreat_api/1.0/send_ss7_alert_to_mthreat</code>
Notes	<pre>@POST @Consumes("text/plain") @Produces("text/plain") @Path("send_ss7_alert_to_mthreat") public Response send_ss7_alert_to_mthreat(String alert)</pre>