P1 Security
Priority One Security

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 Kacer, Philippe Langlois

P1 Security 2017

Introduction

Martin Kacer, Core Network Security Researcher

Philippe Langlois, Security Researcher & CEO P1 Security

P1 Security (http://www.p1sec.com) is dedicated to providing top security products and services for high-expertise security areas.

P1 Labs is the research department of P1 Security.
Conducting research on many subjects related to telecom systems and protocols, mobile apps and platforms, embedded systems, Core Network protocols, etc.

Introduction

Open-source SigFW

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

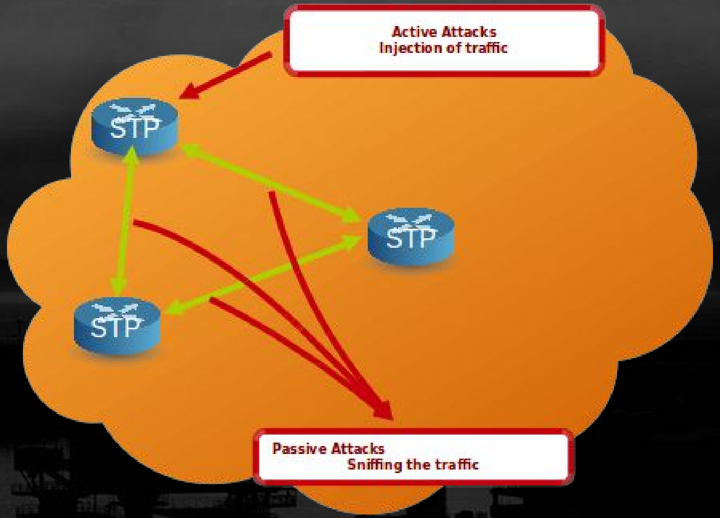
The open-source SigFW should be considered as **reference implementation** and **research project** but **without any warranty** and it is not carrier grade solution.

Motivation of this work

Motivation for this work

Background

Telecom networks (SS7, IPX) are the key infrastructure transmitting **subscribers’ locations, metadata and communication content**



These networks are vulnerable to both **active signalling attacks** and to **passive eavesdropping attacks**

Motivation for this work

Current status

On conferences and publicly in past, most time the attacks were covered

There is a lack of public defense solutions

There is intensive work at the GSMA level (trade body that represents the interests of mobile operators worldwide) and by telecom and private security companies

But there is lack of open-source and affordable tools to improve the security on a wide scale

Some specifications are written

But take a lot of time to become mainstream, if ever adopted.

Motivation for this work

Will SS7 be phased-out soon?

Let's evaluate this…

- The circuit switched voice service could be replaced by VoLTE (4G) with IMS home routed architecture, but such deployment requires VoLTE capable devices and VoLTE networks with a similar radio coverage compared to 2G & 3G. So before an operator decides to shut-down both 2G and 3G networks, **all the home subscribers should be VoLTE enabled.**
- **And still, haven't we forgot the inbound-roamers**?
- Moreover, similar pitfalls as in SS7 are still present in GRX/IPX networks used for mobile data for GTP protocol and in 4G for Diameter protocol.

…so maybe not so soon

Motivation for this work

The Signal App and other mobile encryption Apps

Not covering everything …

- Subscribers are not always attached to mobile data
- Not all users are using it, so there is still fallback to standard Voice and SMS
- The location data could be present in signalling in 4G over Diameter and for 3G, 2G in SS7. This will also apply for VoLTE.
- Most of the time A2P SMS are delivered over SS7

… there is still need to protect the signalling

Motivation for this work

Main Goal

Try to improve telecom security on a wide scale and try to provide solution to address vulnerabilities in SS7 and IPX networks.

Secure messages against advanced attacks:

- Spoofing
- Interception
- Illegal injection

Decision to try make a difference in the World:

- Humbly, with what we can
- Even if P1 Security is doubling size every Year
- Still small (compared to huge Telecom and Mobile giants)

Current status

Current status

SS7 / Sigtran stack overview

Decoding is done from lower layers to upper layers

Filtering should be performed based on decoding different layers

Current status

Major Core Network Elements overview

STP - Like router, but capable also doing filtering/screening

HLR/HSS - Home Location Register. The main database storing home subscribers, profiles, authentication information and locations.

MSC/VLR - switching center connecting the circuits and VLR is storing the subscriber profile received from HLR in HPLMN or VPLMN. Every subscriber in 2G/3G is served by some MSC/VLR.

SGSN - Like MSC/VLR, but for mobile data. Creating data tunnel (GTP) to home GGSN to allow internet connectivity. Every subscriber in 2G/3G with enabled data is served by some SGSN.

SMSC - SMS center. Storing and delivering SMS messages.

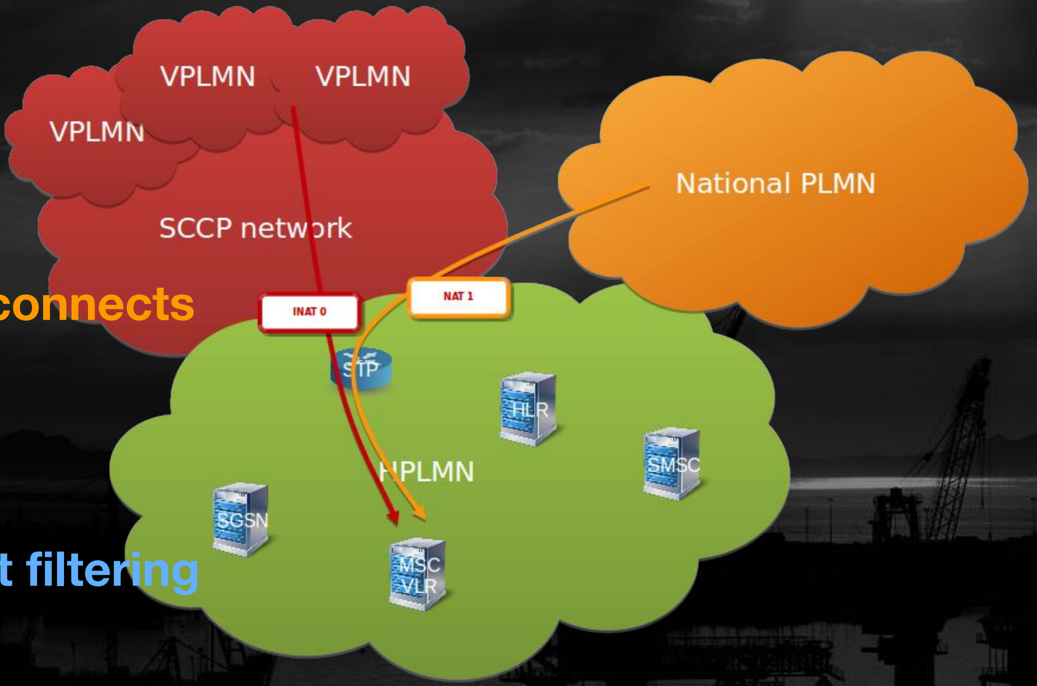
Other - IN, GGSN, PCRF, IVR, P-GW, S-GW, MME, IMS, ...

Current status

Perimeters of SS7 overview

INAT 0: International interconnects (higher risk)

NAT 1: National interconnects (possibly lower risk)



=> There is a different filtering for these perimeters

Current status

SS7 messages categories

“Category” is just a naming indicating a group of similar messages. For messages in the same category the same protection logic could be implemented. Mainly the message direction is important to decide to which category a message belongs. The normal call flows and normal use of the message are 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 it’s 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 require 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).

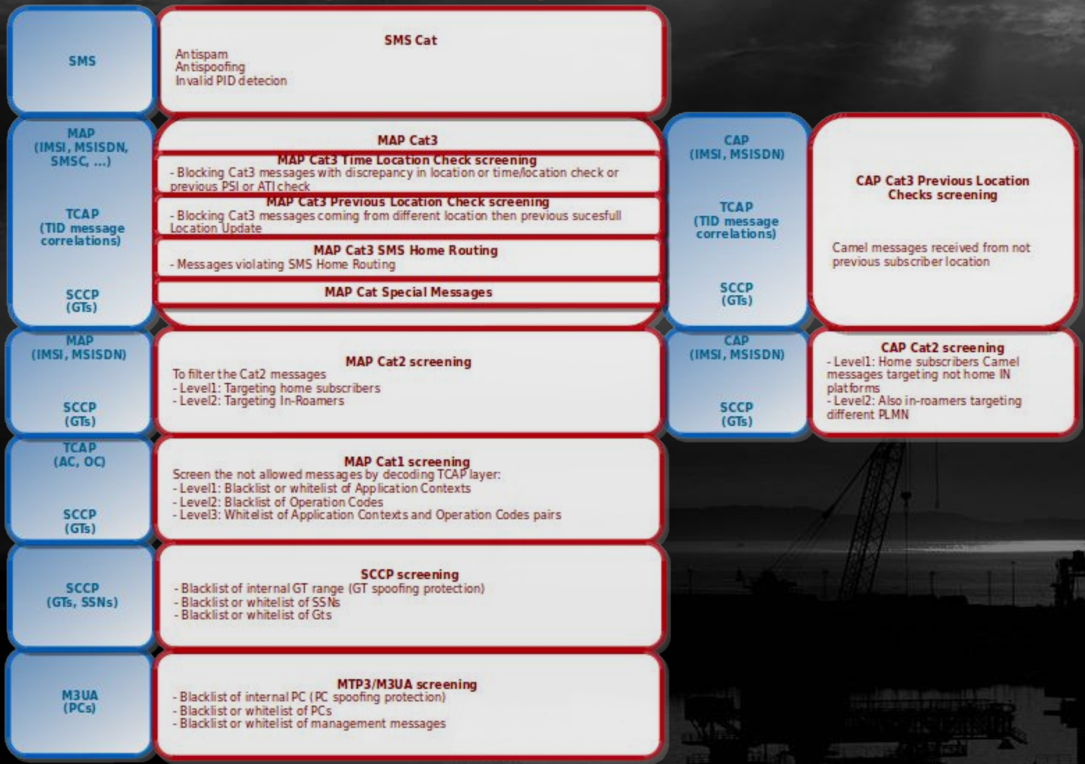
CAP Category 3 messages are Camel messages which should be allowed for outbound-roamers from VPLMN towards HPLMN.

Current status

And we can then create protocol matrices for SS7 …. and also for Diameter and GTP

Current status

SS7 screening categories grouped by protocol layers



Current status

Currently available solutions

Signalling Firewalls

Focused on protecting home network (HPLMN) and filtering illegal traffic originating from different PLMN countries implementing mainly GSMA recommendations

Currently commercial only

Filtering inside the Network elements

Depending on the vendor's capabilities

IDS and monitoring

Current status

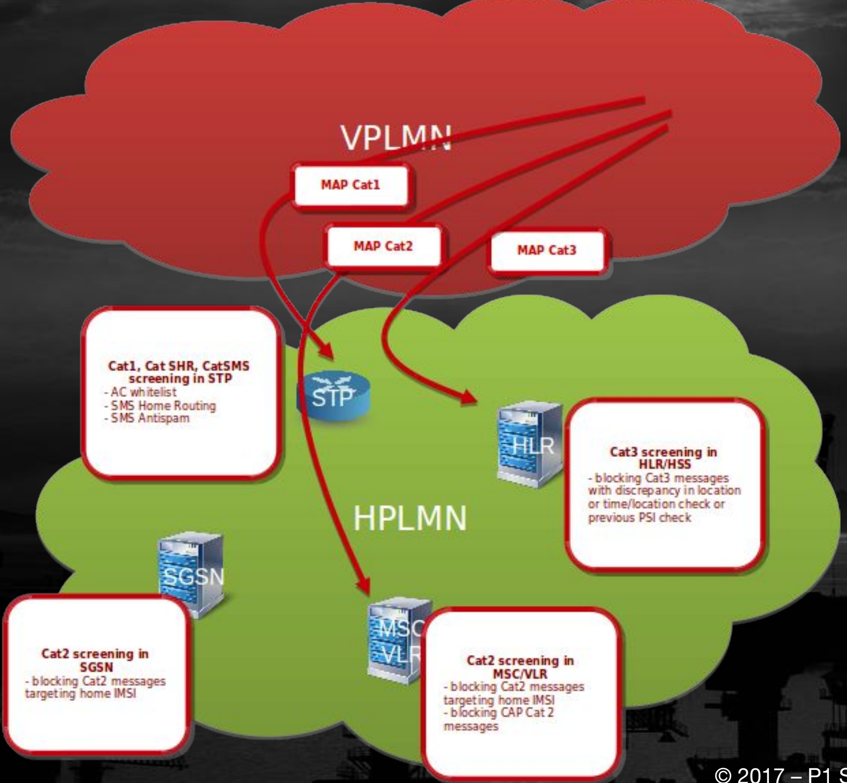
Possible SS7 filtering by existing infrastructure without FW

Better then no filtering

No easy path to enable message confidentiality and integrity protection

Every network element should protect itself

Most STPs can provide Cat1, Cat0 protections



Current status

Conclusion

All these filtering categories more strictly validate the signalling messages according to 3GPP specification and the context of their use.

But no authenticity, integrity protection and confidentiality

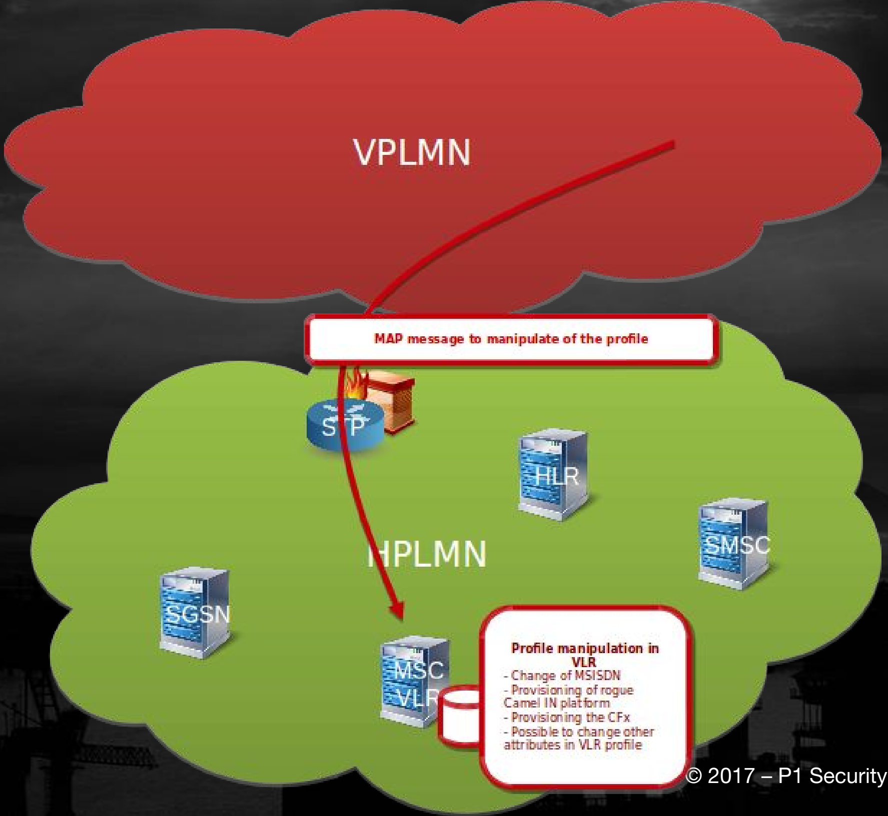
Advanced signalling attacks

Advanced signalling attacks

Category 2 attack example
VLR profile manipulation

In Cat2 there is also manipulation of VLR profile

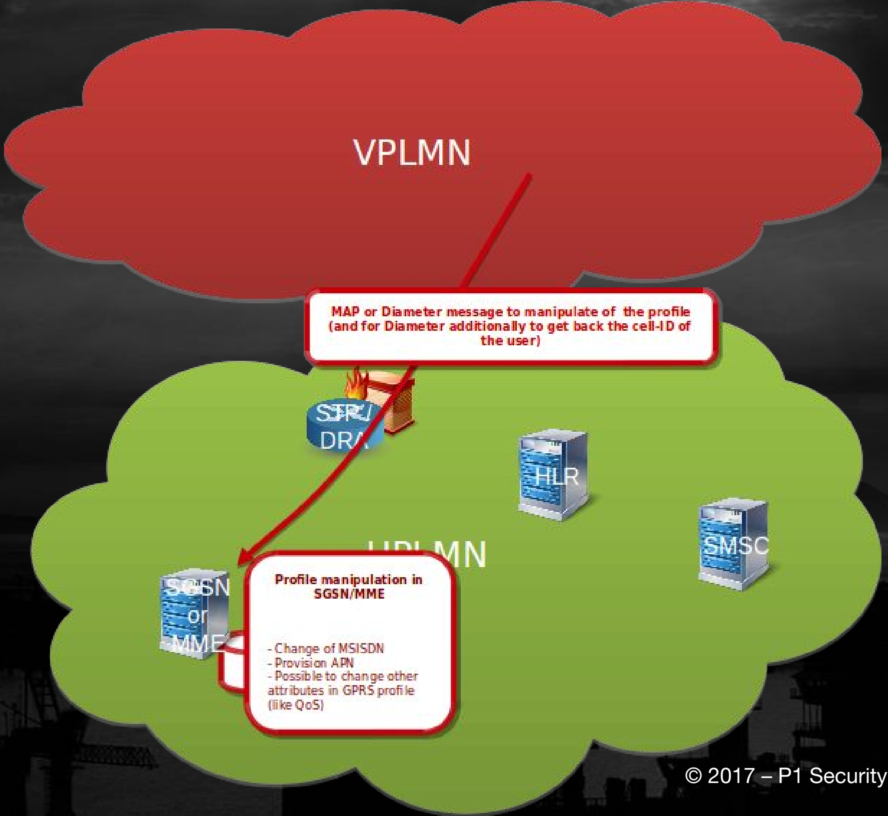
- MSISDN, SS, Camel trigger points manipulation, etc



Category 2 attack example
GPRS/LTE profile manipulation

In Cat2 there is also manipulation of GRPS profile in SGSN or MME.

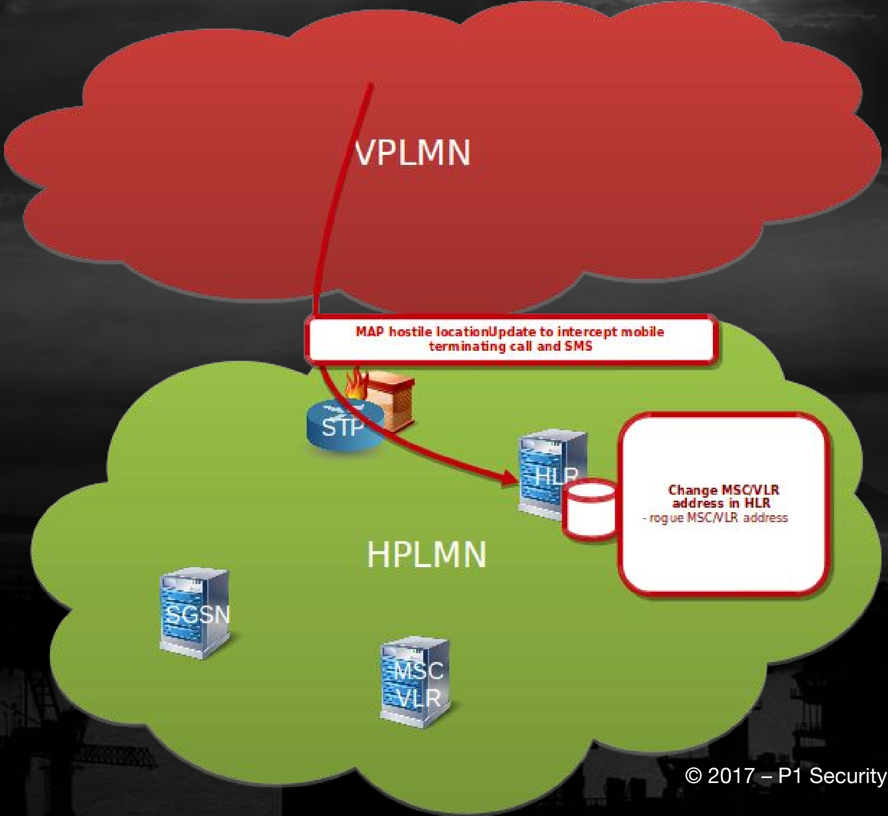
- Accessing the private APNs if there is no AAA used to authenticate APN



Category 3 attack example
Hostile Location Update

In Cat3 there is also manipulation of GRPS profile in SGSN or MME.

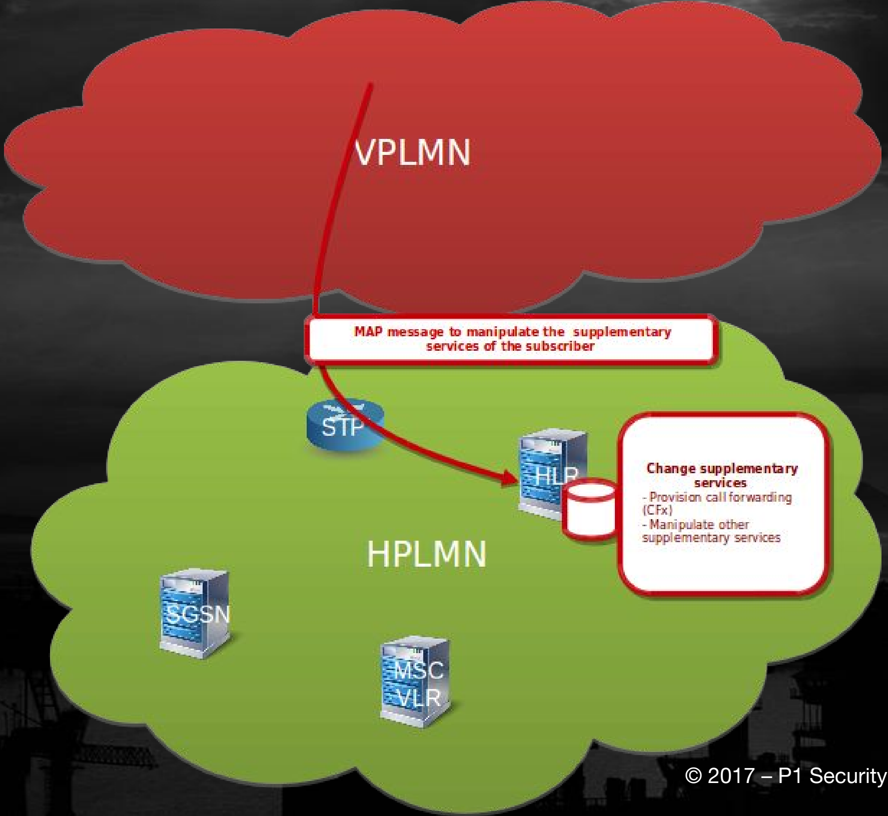
- MT SMS and MT Call interception or targeted MT DoS



Category 3 attack example
Register/Activate SS

In Cat3 there is also manipulation of SS.

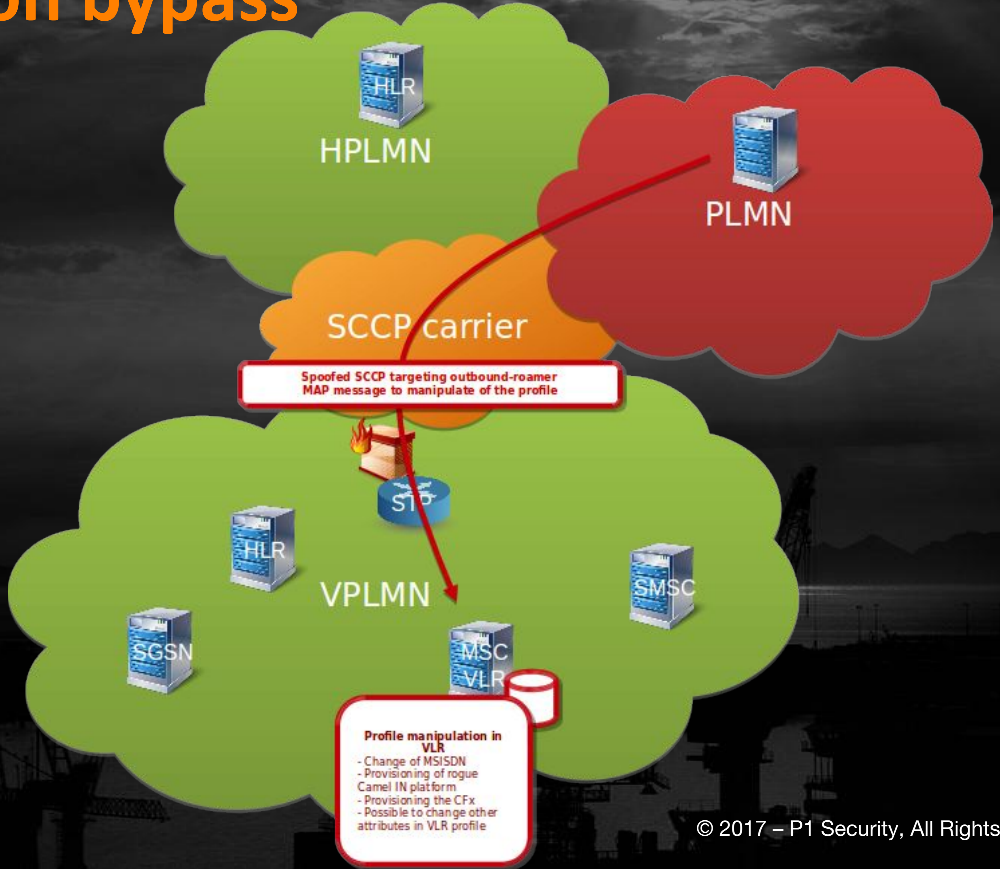
- CFx frauds, SMS forwarding and other



Advanced signalling attacks
Category 2 protection bypass

Outbound-roamer in VPLMN:

Attack targeting outbound-roamers with Cat2 messages with spoofed calling GT.

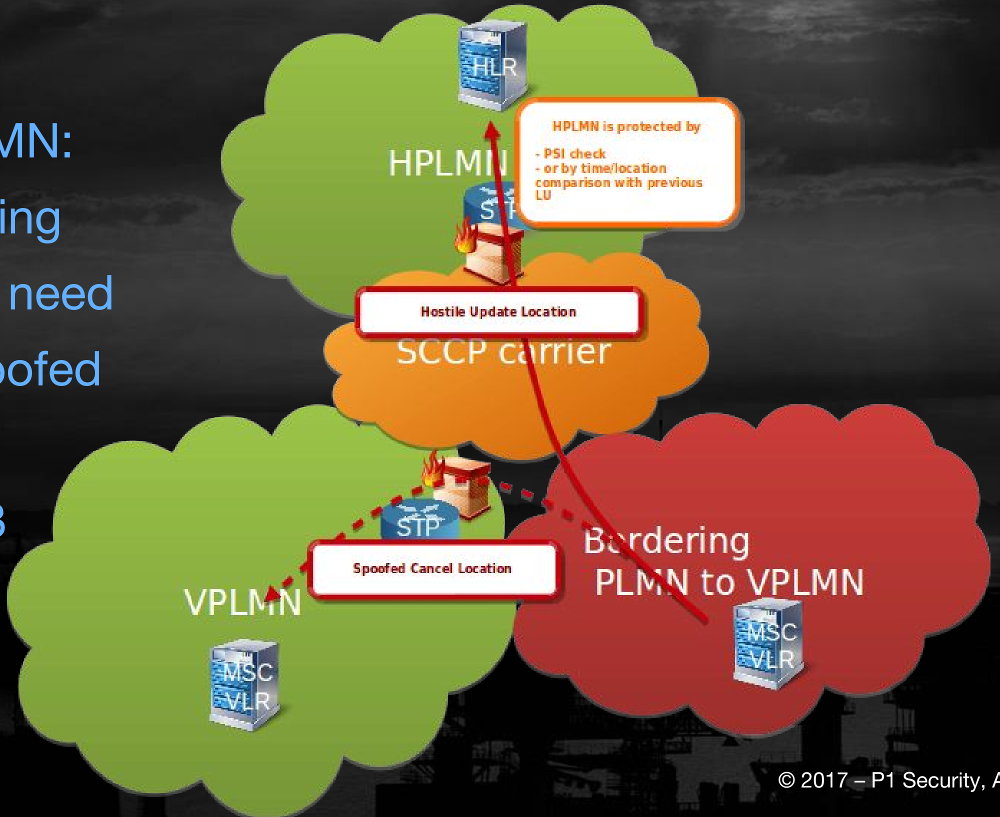


Advanced signalling attacks
Category 3 protection bypass

Outbound-roamer in VPLMN:

Attacker first performing hostile LocationUpdate (if need could use additionally spoofed Cancel Location)

After performing Cat3 messages.



Advanced signalling attacks
MITM

Man In The Middle traffic manipulation:

- Access into SS7 network by MITM in SCTP
- Possibility to inject traffic
- ISD/profile modification
- Authentication vectors modification (RES, IK, CK, AUTN)
- Possibility to modify the Result messages

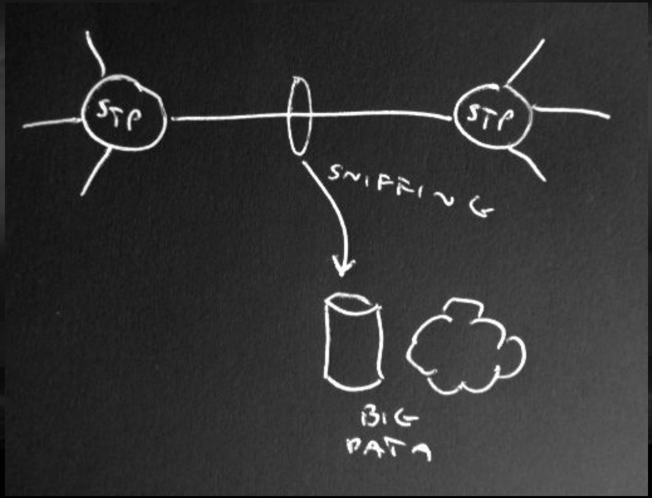
```
SCTP (RFC 3257)

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).
```

Advanced signalling attacks

Passive Attacks

Mass collection of signalling data including mainly:

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

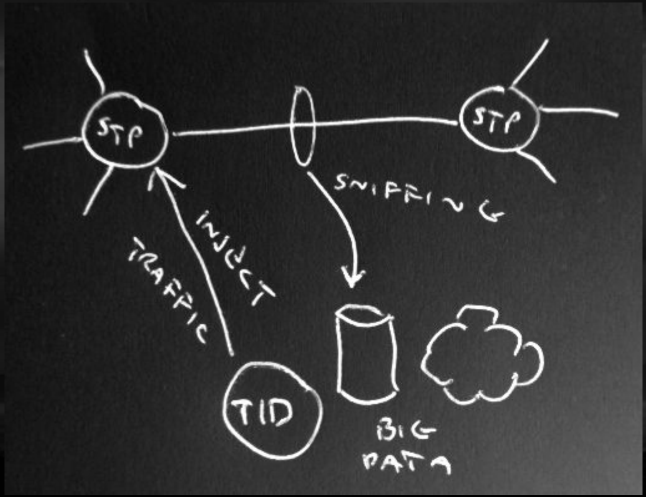


Advanced signalling attacks

Combining Passive and Active Attacks (MoTS)

By knowing the TCAP TID in real time and exact user location it could lead to more sophisticated attacks.

- 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



Advanced signalling attacks

Malformed messages

There are various ways of manipulating and malforming the messages.

Various effects and possible impacts on the network.

Could lead to DoS or Exploitation

Even DoS of the whole network

Advanced signalling attacks

Conclusion

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

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)*

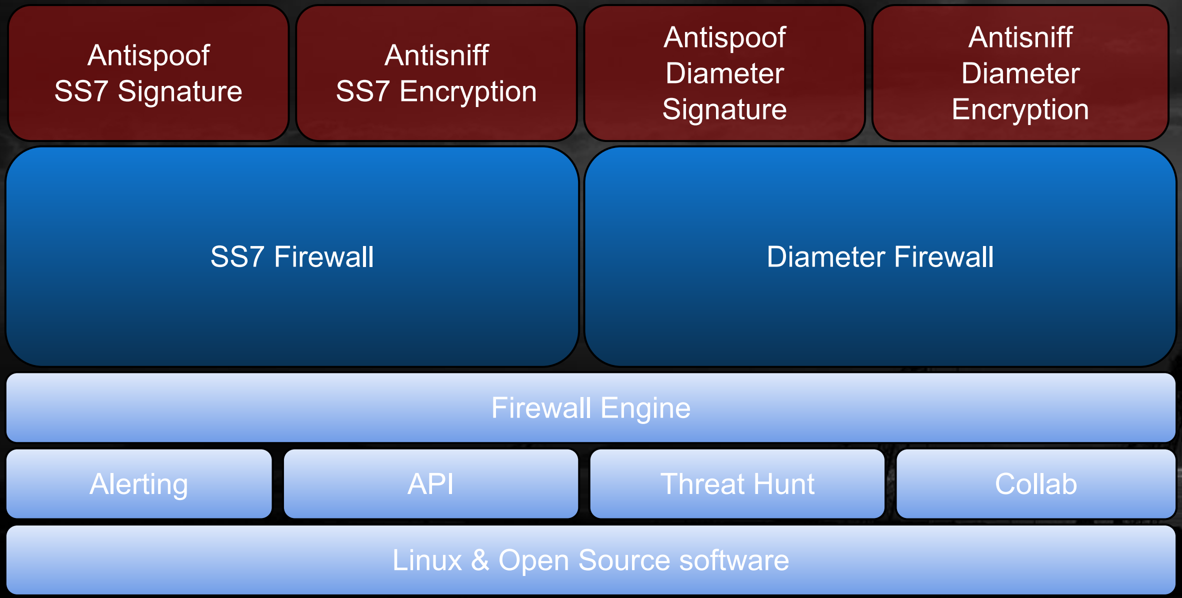
Messages can be integrity protected carrying signature. *(the current work is using proprietary implementation*

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

SigFW
Open SS7 Firewall

Open SS7 Firewall

Well positioned for signaling security enhancement



Future:
GTP, IMS/SIP
…

Features of Open SS7 Firewall

SS7 FW functionalities:

- 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

Maven™

Restcomm

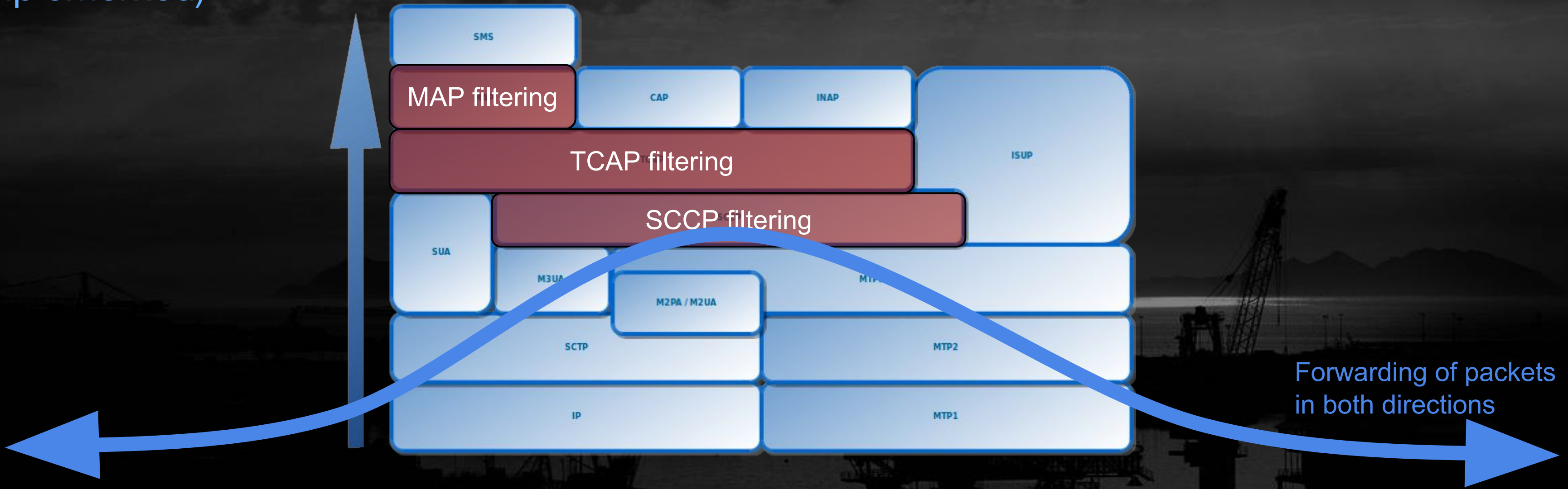
Build in Java Maven
Using free Telestax Mobicent/Restcomm jSS7
License AGPLv3

Open SS7 Firewall

Architecture

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

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

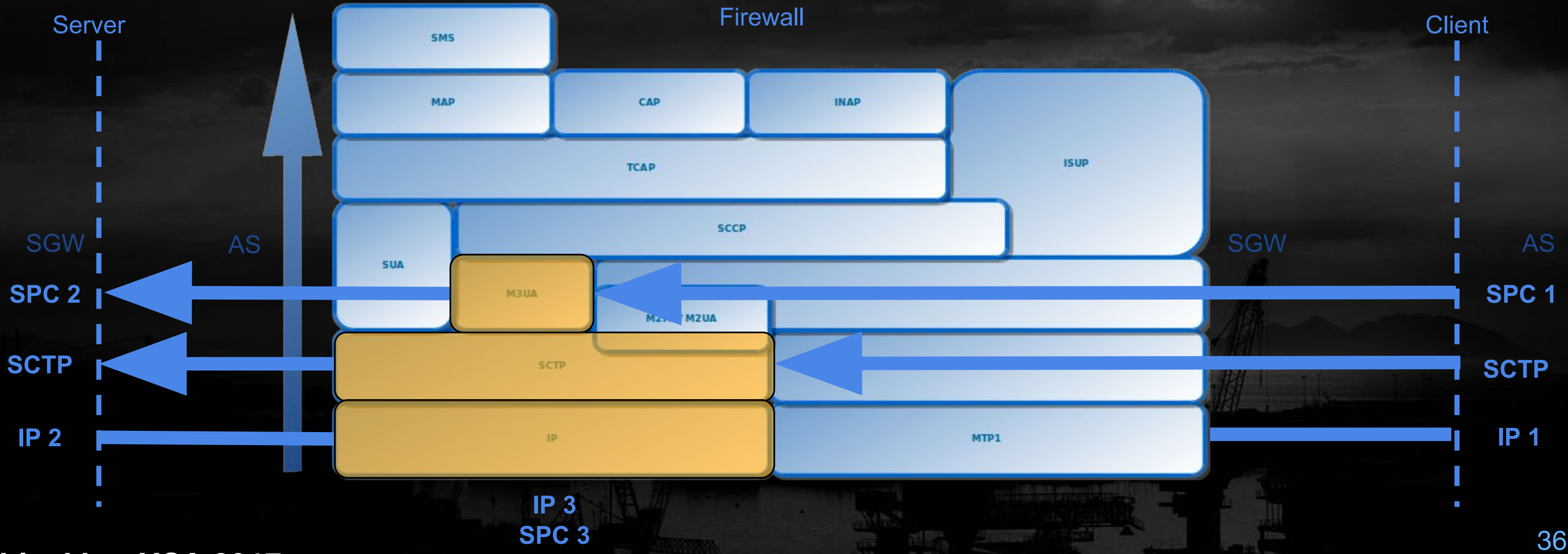


Open SS7 Firewall

Architecture

Firewall is acting like M3UA server and M3UA client, without having SCCP GT.

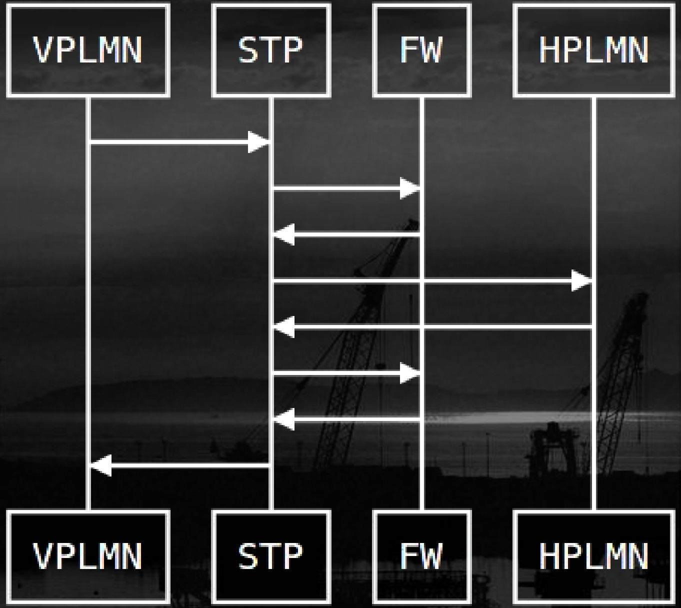
Below is an illustration of the direction of links and associations establishment.



Open SS7 Firewall

Deployment

Loopback on STP towards the FW



Open SS7 Firewall

Progressive deployment & support in networks

Can be used by individual Network Element owner (e.g. HLR owner, SMSC owner)

Not the whole network == progressive introduction

Could provide protection of the individual Network Elements

Allows to deploy FW in limited scope to protect just select network elements, parts of the networks or individual links.

Open SS7 Firewall

APIs

- Signaling Filter Push API *(Manage Firewall Rules)*
- Signaling Message Evaluation API *(Message evaluation with external IDS signalling system)*
- mThreat API *(to report the detected attacks)*

APIs allows to manage the FW and integrate it with other systems.

Open SS7 Firewall

Config

JSON syntax
(Compatible with P1 PTM IDS)

IP, SCTP, M3UA configuration

Firewall filtering rules

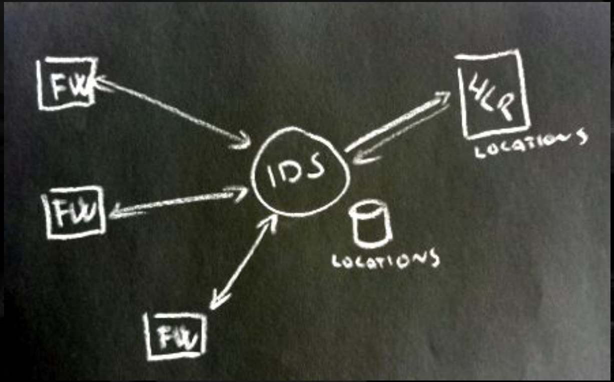
Config is periodically saved to store the changes


```
"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",
```

Open SS7 Firewall

Signaling Messages Evaluation API

- FW forwards the SCCP message to the Signalling IDS
- Signalling IDS responds back with the result (allow/filter message)
- FW performs the filtering action
- By this integration no need for the FW to contain it's 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

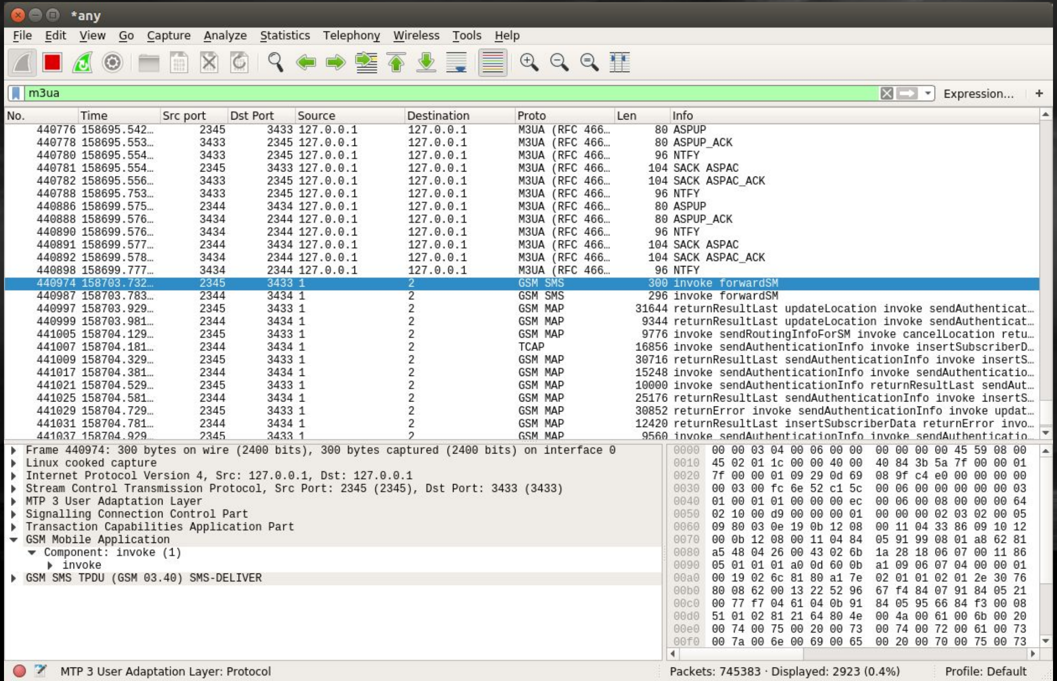


Open SS7 Firewall

Open SS7 FW Passive Mode

Example of replayed traffic on localhost

“Passive mode”



Open SS7 FW Passive Mode

Passive mode is implemented in VM in the following way:

1. tshark live capture to Json EK
2. SS7ClientLiveInput is reading sccp_raw from named pipe and forwarding it to FW
3. SS7FW performs the filtering
4. SS7Server receives the unfiltered traffic

Passive mode can enable to evaluate and validate the FW without deploying it actively into the network

Open SS7 Firewall

SS7 Encryption / Signatures

Current version is also capable of

- **Signing/Verifying 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 or Signatures could be optionally enabled.

Open SS7 Firewall

SS7 Encryption Config

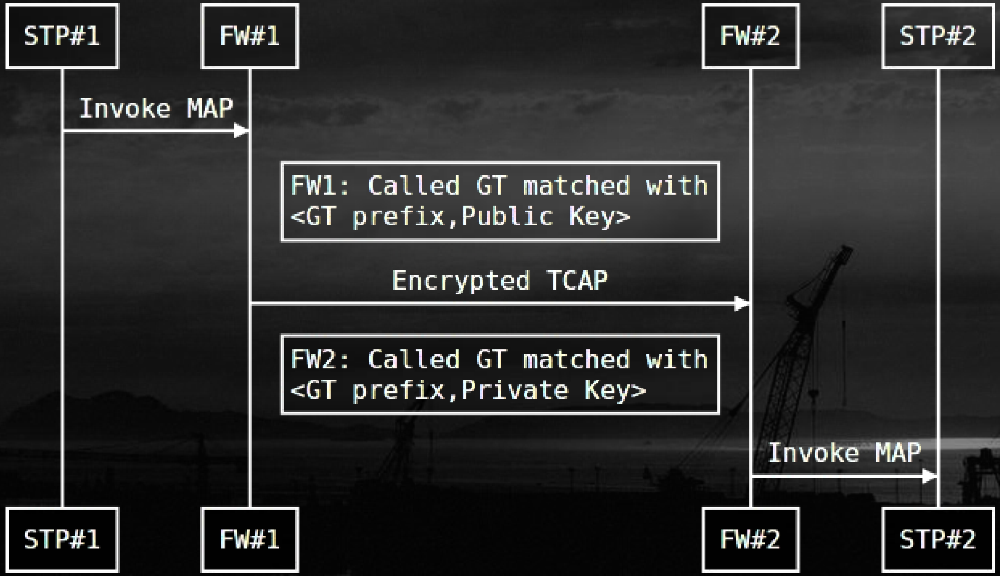
```
"encryption_rules": {
                    "called_gt_encryption": [
                        {
                            "called_gt": "0*",
                            "public_key":
"MIGfMA0GCSqGSIb3DqEBAQUAA4GNADCBiQKBgQCm/PAsXOj7cjirJsQsiIfHauFNLwBIuM1brkUm3aVXeraDIej2BWXmWlKMmX/FRZh4Qhe9mUy6YgwTO8PndWdMDRWMw8vvXJ
FI7HPJpsNfcBykefSqhr5X4h6HyQr63V8O0U5PtgCBuVoyuOFIj84WFwaLuajHiQgps7NOloeH1WIDAQAB"
                        }
                    ],
                    "called_gt_decryption": [
                    ],
                },
                "signature_rules": {
                    "calling_gt_verify": [
                    ],
```

GT prefix defining the PLMN where this public key should be used

Public Key Encoded by Base64

Open SS7 Firewall

SS7 Encryption Flow



Open SS7 Firewall

SS7 Encryption Algorithm

1. The whole TCAP layer is encrypted

2. The following payload is created:
   ```
   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 only RSA/ECB/PKCS1Padding is used in the code

4. Timestamp is verified after decryption to prevent attacks replay

Open SS7 Firewall

SS7 Encryption Example

25890	2017-04-04…	2344	3434	11111111111	VLR (Visitor Location Re…	00000000000	HLR (H…	111111…	GSM MAP	226	0000003b	invoke processUnstructuredSS-R…
25891	2017-04-04…	2345	3433	11111111111	HLR (Home Location Regis…	00000000000	VLR (V…	111111…	GSM MAP	222	0000003c	invoke unstructuredSS-Request
25893	2017-04-04…	2349	3439	11111111111,11…	HLR (Home Location Regis…	00000000000,…	VLR (V…		TCAP	614		XUDT (Message reassembled) XUD…
25895	2017-04-04…	2344	3434	11111111111	HLR (Home Location Regis…	00000000000	VLR (V…	111111…	GSM MAP	226	0000003c	invoke unstructuredSS-Request
25897	2017-04-04…	2345	3433	11111111111	HLR (Home Location Regis…	00000000000	VLR (V…	111111…	GSM MAP	210	0000003d	invoke unstructuredSS-Notify
• 25899	2017-04-04…	2349	3439	11111111111	HLR (Home Location Regis…	00000000000	VLR (V…		SCCP (…	362		XUDT (Message reassembled)
• 25903	2017-04-04…	2349	3439	11111111111	HLR (Home Location Regis…	00000000000	VLR (V…		TCAP	170		XUDT (Message reassembled)
25904	2017-04-04…	2344	3434	11111111111	HLR (Home Location Regis…	00000000000	VLR (V…	111111…	GSM MAP	218	0000003d	invoke unstructuredSS-Notify
25905	2017-04-04…	2345	3433	11111111111	gsmSCF (MAP) or IM-SSF (…	00000000000	HLR (H…		GSM MAP	206	0000003e	invoke anyTimeSubscriptionInte…
25909	2017-04-04…	2345	3433	11111111111	HLR (Home Location Regis…	00000000000	MSC (M…		GSM MAP	182	0000003f	invoke informServiceCentre
25910	2017-04-04…	2349	3439	11111111111	HLR (Home Location Regis…	00000000000	MSC (M…		SCCP (…	362		XUDT (Message reassembled)
25913	2017-04-04…	2349	3439	11111111111	HLR (Home Location Regis…	00000000000	MSC (M…		TCAP	170		XUDT (Message reassembled)
25914	2017-04-04…	2344	3434	11111111111	HLR (Home Location Regis…	00000000000	MSC (M…		GSM MAP	190	0000003f	invoke informServiceCentre
25915	2017-04-04…	2345	3433	11111111111	HLR (Home Location Regis…	00000000000	MSC (M…		GSM MAP	190	00000040	invoke alertServiceCentre
25917	2017-04-04…	2349	3439	11111111111,11…	HLR (Home Location Regis…	00000000000,…	MSC (M…		TCAP	486		XUDT (Message reassembled) XUD…
25919	2017-04-04…	2344	3434	11111111111	HLR (Home Location Regis…	00000000000	MSC (M…		GSM MAP	198	00000040	invoke alertServiceCentre
25921	2017-04-04…	2345	3433	11111111111	gsmSCF (MAP) or IM-SSF (…	00000000000	HLR (H…		GSM MAP	206	00000041	invoke anyTimeModification
25925	2017-04-04…	2345	3433	11111111111	HLR (Home Location Regis…	00000000000	VLR (V…		GSM MAP	190	00000042	invoke readyForSM
25926	2017-04-04…	2349	3439	11111111111	HLR (Home Location Regis…	00000000000	VLR (V…		SCCP (…	362		XUDT (Message reassembled)
25929	2017-04-04…	2349	3439	11111111111	HLR (Home Location Regis…	00000000000	VLR (V…		TCAP	170		XUDT (Message reassembled)
25930	2017-04-04…	2344	3434	11111111111	HLR (Home Location Regis…	00000000000	VLR (V…		GSM MAP	194	00000042	invoke readyForSM
25931	2017-04-04…	2345	3433	11111111111	VLR (Visitor Location Re…	00000000000	HLR (H…		GSM MAP	194	00000043	invoke purgeMS
25933	2017-04-04…	2349	3439	11111111111,11…	VLR (Visitor Location Re…	00000000000,…	HLR (H…		TCAP	486		XUDT (Message reassembled) XUD…
25935	2017-04-04…	2344	3434	11111111111	VLR (Visitor Location Re…	00000000000	HLR (H…		GSM MAP	198	00000043	invoke purgeMS
25937	2017-04-04…	2345	3433	11111111111	MSC (Mobile Switching Ce…	00000000000	MSC (M…		GSM MAP	186	00000044	invoke prepareHandover
25941	2017-04-04…	2345	3433	11111111111	MSC (Mobile Switching Ce…	00000000000	MSC (M…		GSM MAP	182	00000045	invoke prepareSubsequentHandov…
25943	2017-04-04…	2345	3433	11111111111	HLR (Home Location Regis…	00000000000	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:
▸ 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 bb f6   ........ {
0010  63 ae ad 1b 57 ea dd 52  4f ce a7 de 55 e5 7e b3   c...W..R O
0020  8d cf ed be a5 83 2b b8  de 32 db ed 07 be 8b 6a   ......+. .
0030  b6 ad 3e 31 62 20 9b 7f  50 8e 27 97 c7 e8 0c 0f   ..>1b .. P
0040  ff d3 c2 9b 4f a3 83 0b  7d 3f c4 03 3d 34 3b 73   ....O... }
0050  45 ac 96 44 0f b4 55 7b  db 3a 7f 3d 5c c6 a3 ed   E..D..U{ .
0060  82 ab 38 30 a9 46 2d 6b  6c 48 8d e6 7e cd 3e 20   ..80.F-k l
0070  6f d0 f0 f5 ee 1a 31 f6  cc 11 81 94 d3 2c 56 ff   o.....1. .
0080  03 09 5f ee 3d 69 0c e7  9c 5a 60 81 01 77 47 82   .._.=i.. .
0090  f5 e6 47 98 1d c1 f1 61  d8 c7 ac fa f3 58 dd bf   ..G....a .
00a0  ce 40 fb d6 1d 12 b3 fc  80 e0 c0 88 ca a1 93 41   .@...... .
```

Open SS7 Firewall

SCCP UDT / XUDT

XUDT messages has been seen on the previous slide.

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

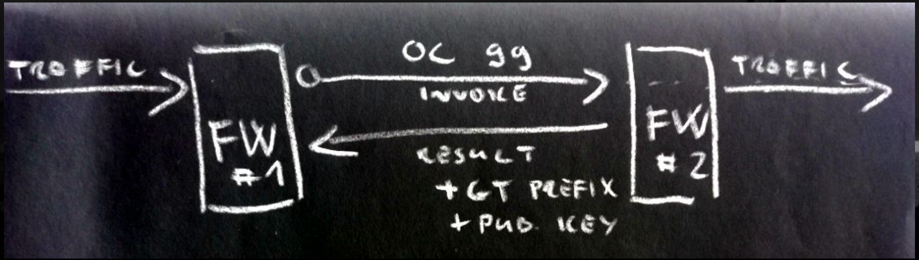
After decryption on the other end the message is again reconstructed into UDT message.

Open SS7 Firewall

SS7 Encryption Autodiscovery

Feature to enable encryption autodiscovery

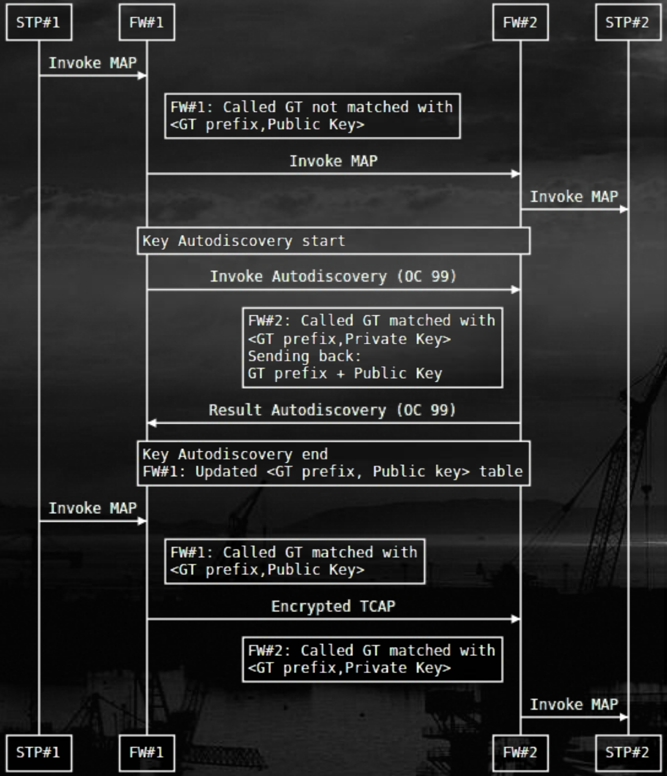
1. The FW #1 will send a MAP Invoke (New OpCode 99) for destinations with no known Public Keys
2. If there FW #2 is in the path, it processes the Invoke and sends Result (including GT prefix and Public Key)
3. FW #1 config is updated with gathered Public Keys



Autodiscover should enable easier key management

Open SS7 Firewall

SS7 Encryption Flow - autodiscovery



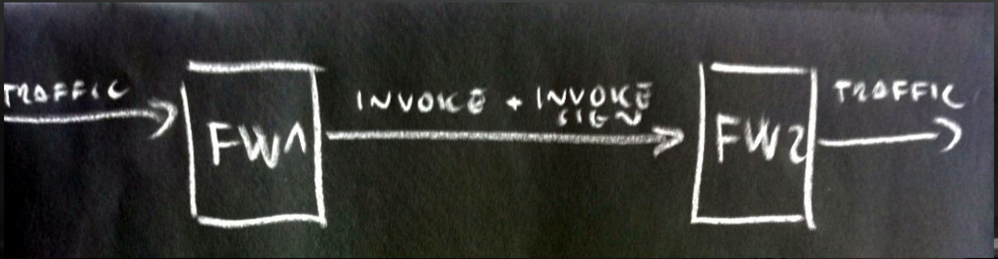
Open SS7 Firewall

TCAPsec comparison

	Current Encryption	TCAPsec
Last update	2017	3GPP Rel.7 2006
History	Reuse email security principles	Evolved from MAPsec (Mapsec was deprecated)
Type of encryption	Asymmetric Integrity and Confidentiality is independent (like in email security: signature and encryption)	Symmetric AES Point to Point Protection Mode 1 and 2 (with Confidentiality)
Key exchange	Public Keys Autodiscovery	Manual exchange between every 2 operators

Open SS7 Firewall
SS7 Signature

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



Open SS7 Firewall

SS7 Signature Algorithm

1. Only TCAP Begins are signed

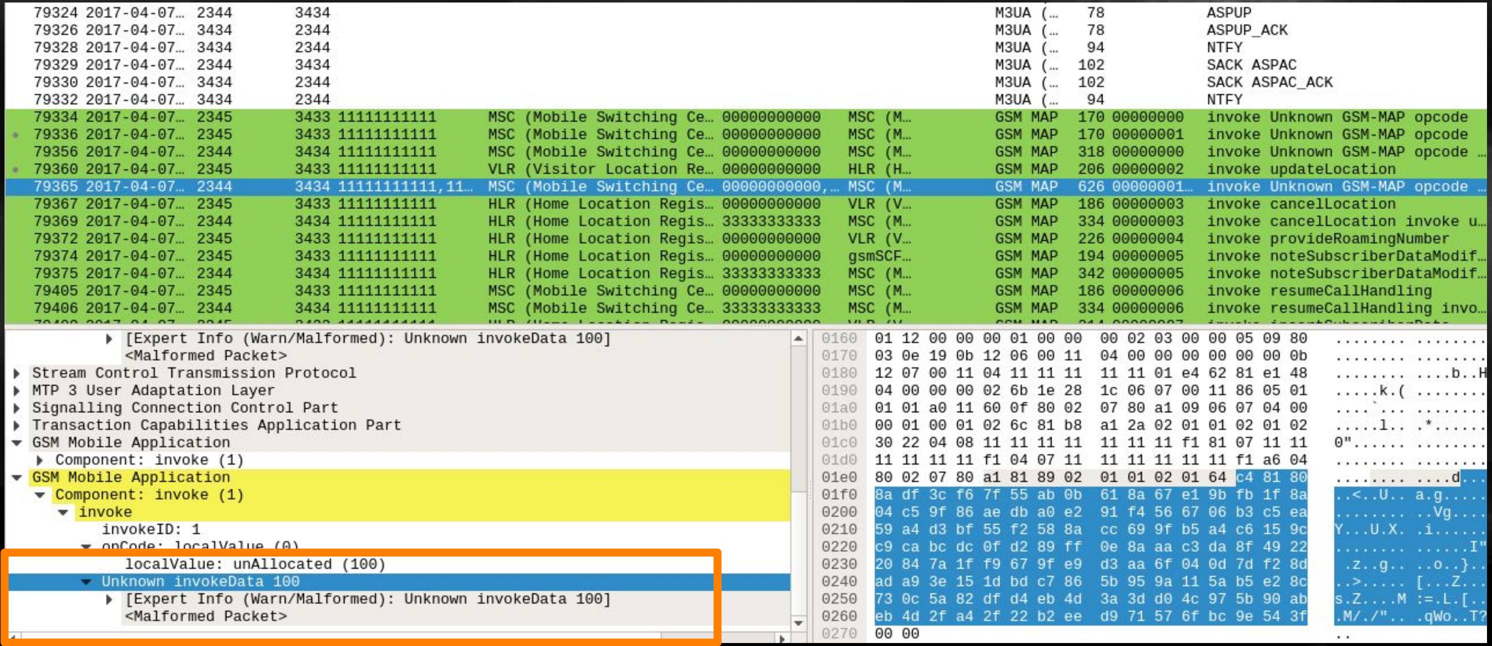
2. Check if the TCAP already contains some TCAP Invoke signature component. If not, sign it.

3. TCAP signature component will contain:
   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)

Open SS7 Firewall

SS7 Signature

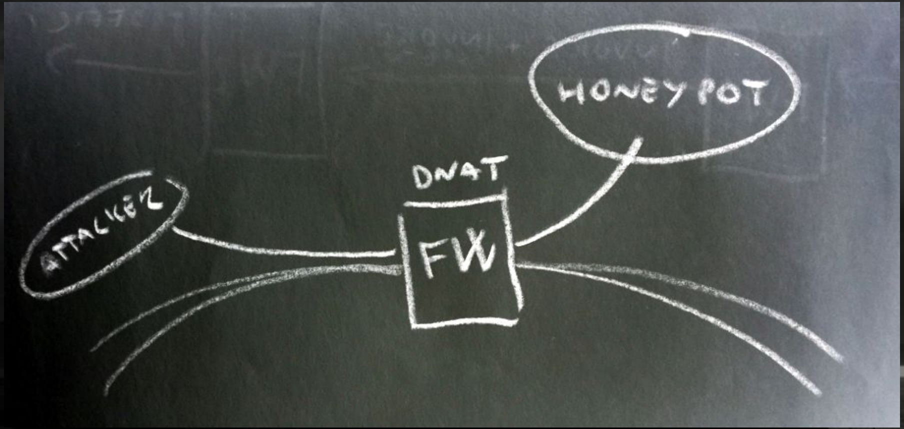


TCAP Signature in Additional Invoke with new OpCode

Open SS7 Firewall

DNAT to Honeypot

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

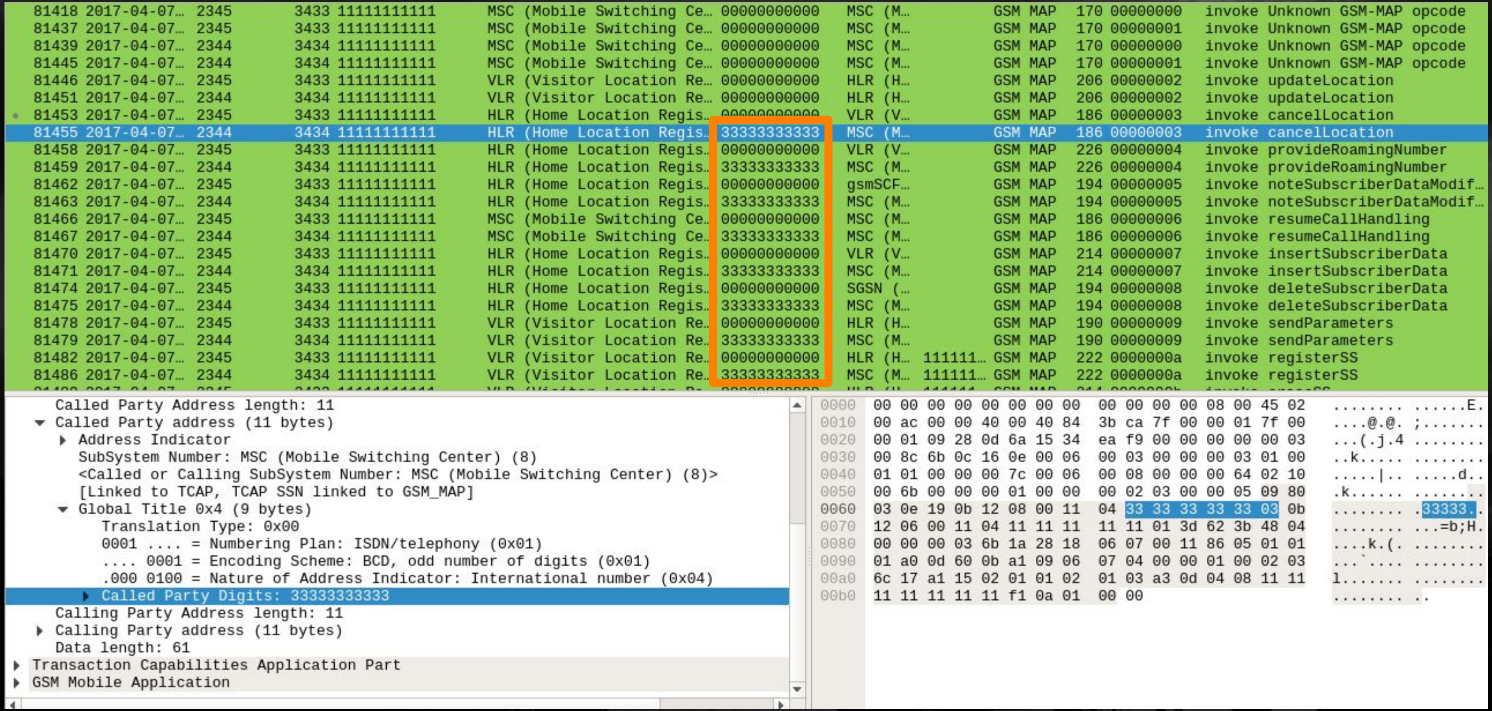


FW supported actions: DROP_SILENTLY, DROP_WITH_SCCP_ERROR, DNAT_TO_HONEYPOT, ALLOW

Open SS7 Firewall

DNAT to Honeypot

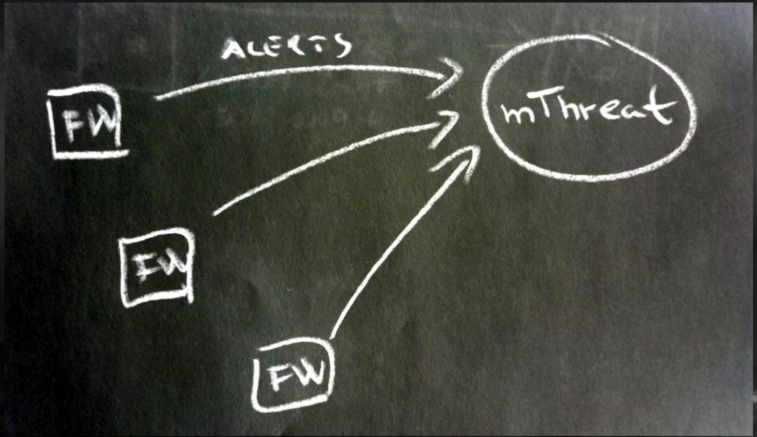
DNAT to new called GT



Open SS7 Firewall

mThreat

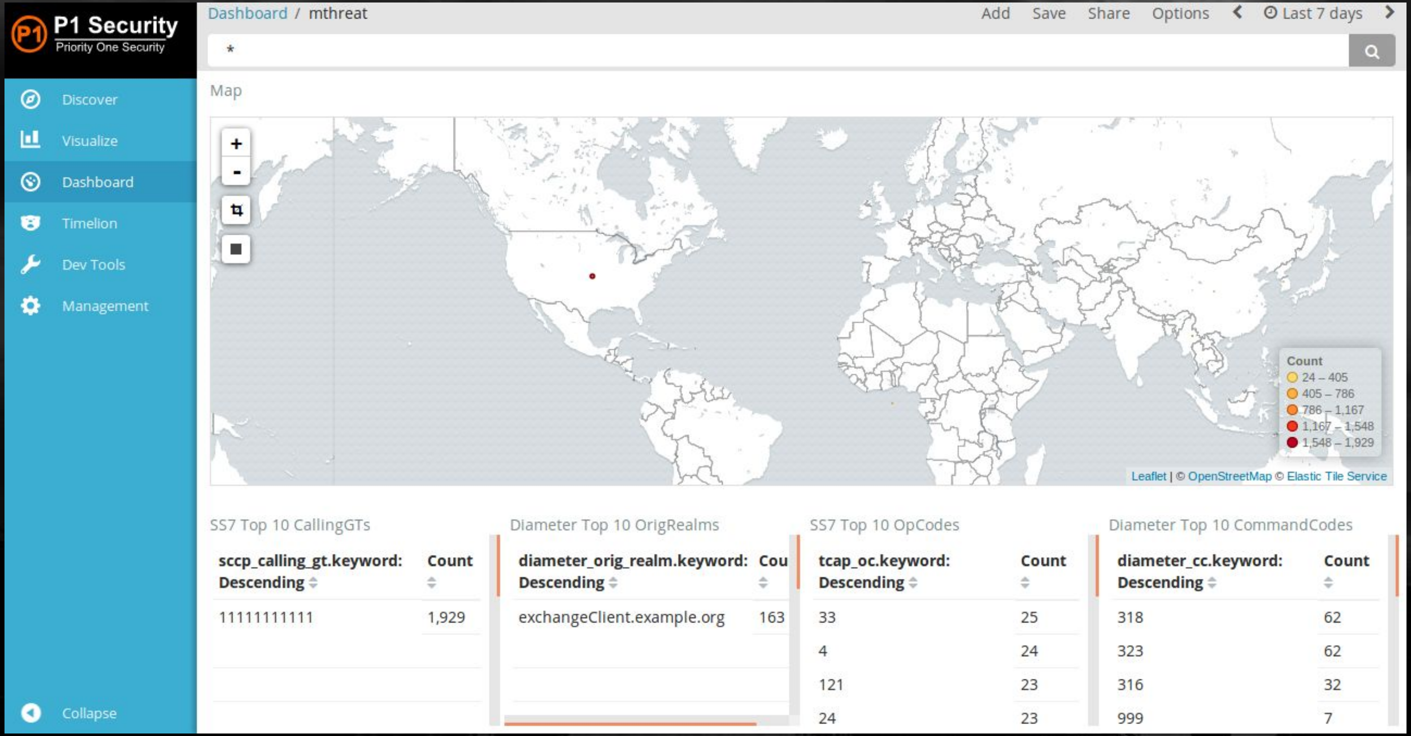
Every firewalled event can be optionally sent to mThreat over mThreat API



FW Alerts can be anonymized and reported to central Threat intelligence

Open SS7 Firewall

mThreat Example



SigFW
Open Diameter Firewall

Open Diameter Firewall
Overview

The source code contains also a **Diameter Firewall** with similar capabilities.

To address Diameter security is mainly important for 4G and 5G.

In Diameter, message spoofing brings additional vulnerability, because of Route-Records AVPs, the attacker can get back Diameter Answers to spoofed Origin-Host and Origin-Realm messages.

Open Diameter Firewall

Features of Open Diameter Firewall

Diameter FW functionalities:

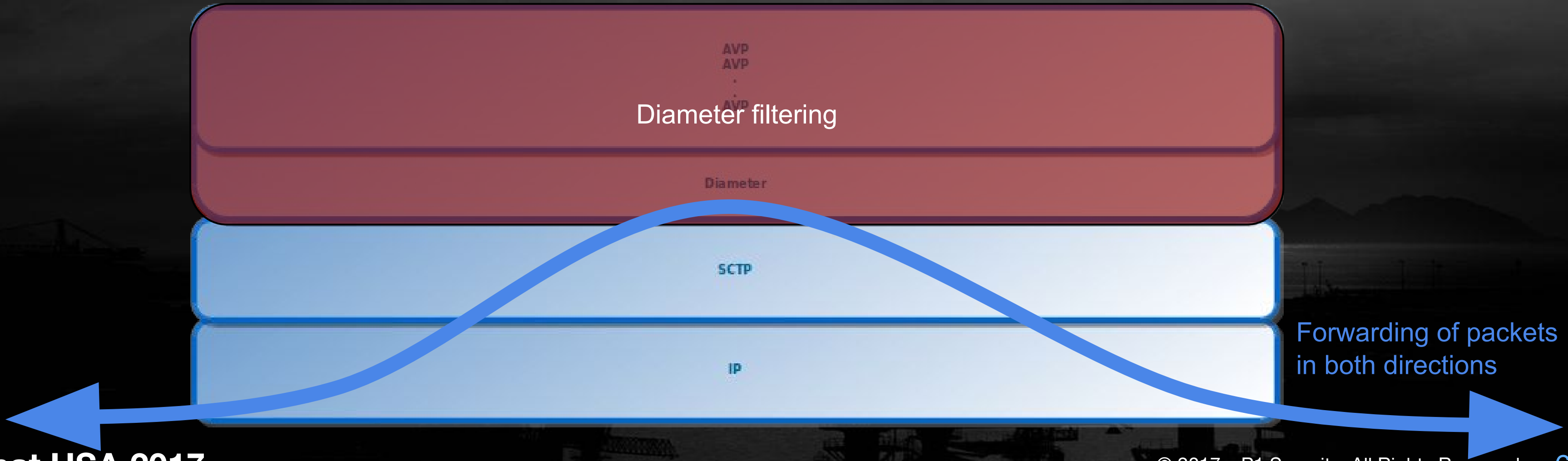
- Open Diameter encryption and signing of the Diameter messages, including auto encryption setup
- Diameter host and realms blacklists (Category 0)
- Diameter Command Code blacklists and Realm whitelist (Category 1)
- Diameter firewall rules (Category 2)
- Signalling IDS integration (for Category 3 and 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

Build in Java Maven
Using free Telestax
Mobicent/Restcomm
jDiameter
License AGPLv3

Open Diameter Firewall
Architecture

Frames are forwarded on SCTP layer

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



Open Diameter Firewall

Architecture

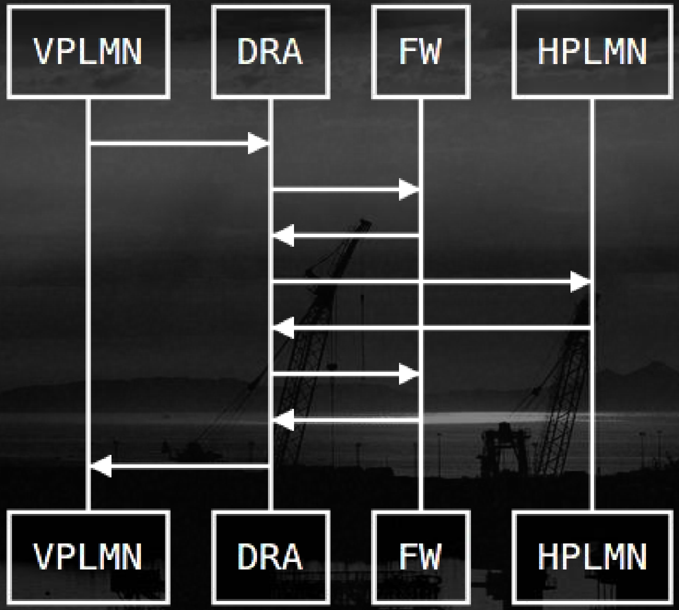
Firewall is acting like SCTP server and SCTP client, without having Diameter Address. The Diameter CER, DWR, DPR or forwarded.

Below is an illustration of the direction of links and associations establishment.

Open Diameter Firewall

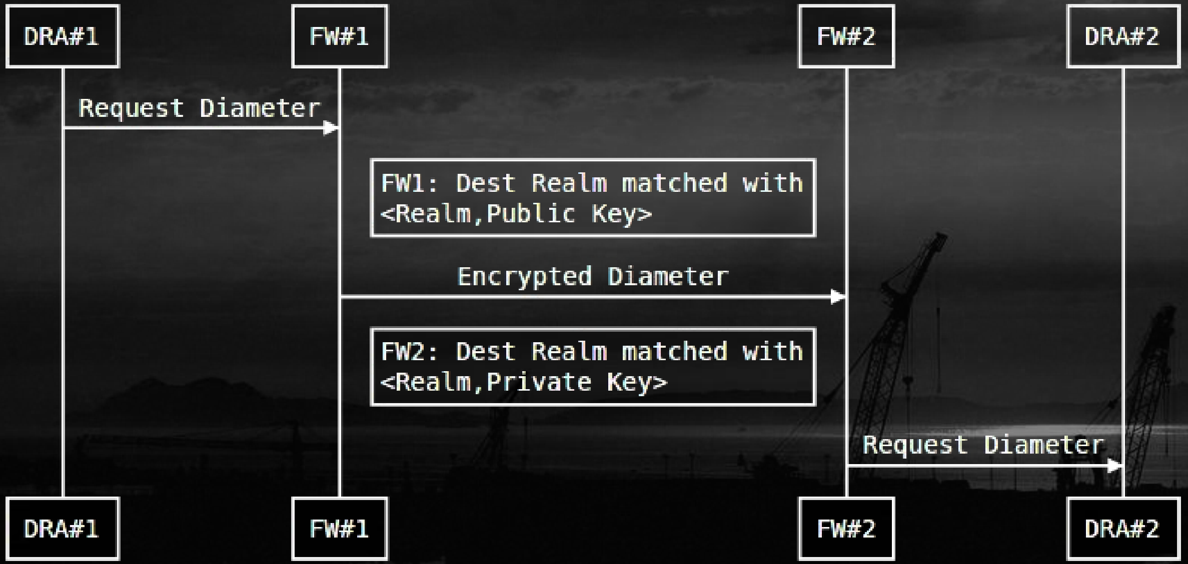
Deployment

Loopback on DRA towards the FW



Open Diameter Firewall

Diameter Encryption flow



Open Diameter Firewall

Diameter Encryption Algorithm

1. Encryption is at the Diameter AVP level

2. AVPs required for IPX carriers are unencrypted (mainly host, realm, route)

3. The following payload is encrypted 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 only RSA/ECB/PKCS1Padding is used in the code.

5. Timestamp is verified after decryption to prevent replay attacks

Open Diameter Firewall

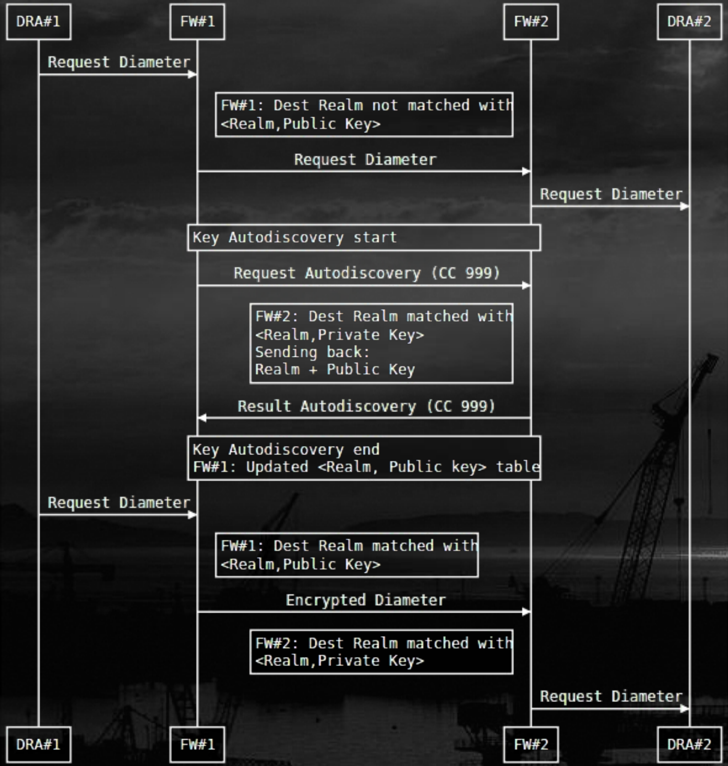
Diameter Encryption Example

```
147 62.930384208  127.0.0.1        127.0.0.1        DIAMET…   462 cmd=3GPP-Notify Request(323) flags=R--- appl=3GPP S6a/S6d(16777251) 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(16777251) 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(16777251) 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(16777251) 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(16777251) h2h=4a49277d e2e…
159 62.966246812  127.0.0.1        127.0.0.1        DIAMET…  1514 SACK cmd=3GPP-Update-Location Request(316) flags=R--- appl=3GPP S6a/S6d(16777251) h2h=4a49277d e2e…
164 62.985854473  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=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(16777251) h2h=4a49277d e2e=…
168 62.992970061  127.0.0.1        127.0.0.1        DIAMET…  1418 SACK cmd=3GPP-Update-Location Answer(316) flags=--E- appl=3GPP S6a/S6d(16777251) 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(16777251) h2h=4a49277d e2e=…
186 92.995232305  127.0.0.1        127.0.0.1        DIAMET…   142 cmd=Device-Watchdog Request(280) flags=R--- appl=Diameter Common Messages(0) h2h=4a492fc2 e2e=70b0…
187 92.996785046  127.0.0.1        127.0.0.1        DIAMET…   142 cmd=Device-Watchdog Request(280) flags=R--- appl=Diameter Common Messages(0) h2h=4a492fc2 e2e=70b0…
188 92.998244255  127.0.0.1        127.0.0.1        DIAMET…   142 cmd=Device-Watchdog Request(280) flags=R--- appl=Diameter Common Messages(0) h2h=4a492fc2 e2e=70b0…
189 92.999627596  127.0.0.1        127.0.0.1        DIAMET…   166 SACK cmd=Device-Watchdog Answer(280) flags=---- appl=Diameter Common Messages(0) h2h=4a492fc2 e2e=…
190 93.000873609  127.0.0.1        127.0.0.1        DIAMET…   166 SACK cmd=Device-Watchdog Answer(280) flags=---- appl=Diameter Common Messages(0) h2h=4a492fc2 e2e=…
191 93.002105486  127.0.0.1        127.0.0.1        DIAMET…   166 SACK cmd=Device-Watchdog Answer(280) flags=---- appl=Diameter Common Messages(0) h2h=4a492fc2 e2e=…
```

```
▸ Flags: 0x80, Request
  Command Code: 316 3GPP-Update-Location
  ApplicationId: 3GPP S6a/S6d (16777251)
  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=3650b1097190e791b8d03f53f9eb1afffbdc34cc5afdb80f...
▸ AVP: Origin-Host(264) l=59 f=-M- val=[redacted]
▸ AVP: Unknown(1100) l=136 f=--- val=79d158db1690d26781fffbcbf040ec7023dabc728bf5556a...
▸ 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=7290d5360b1fcecbf95fe43055cb5f49ef273f3ce8a43b04...
```

Open Diameter Firewall

Diameter Encryption Autodiscovery



Open Diameter Firewall

Diameter Signature Algorithm

1. Only Diameter Requests are signed

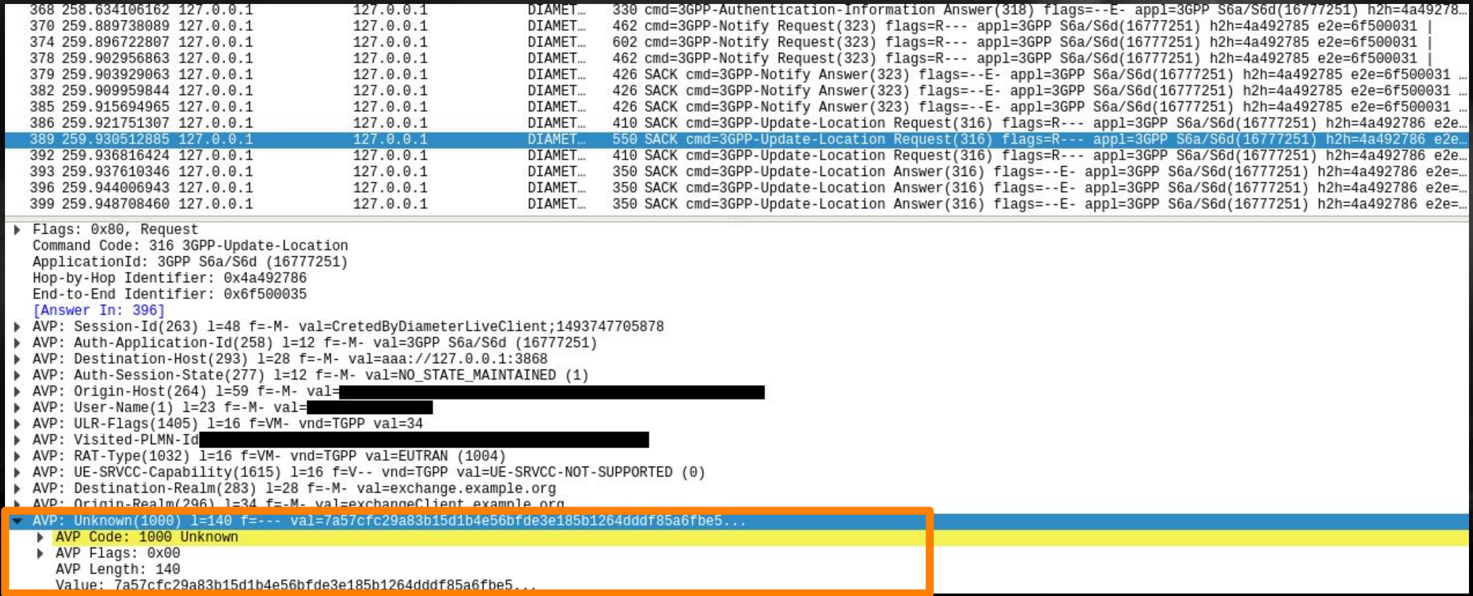
2. Check if the Diameter message already contains some Diameter signature AVP. If not, sign it.

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)`

Open Diameter Firewall

Diameter Signature



Diameter Signature in Additional AVP with new Code

SigFW

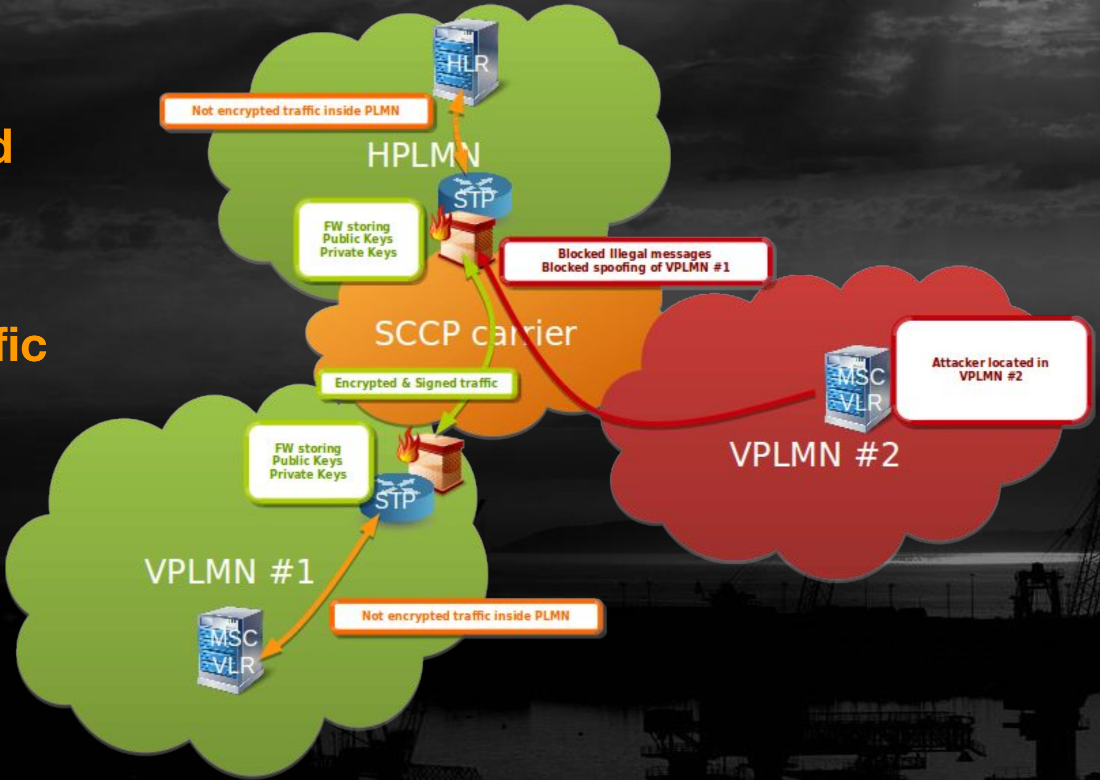
Open Source SS7/Diameter firewall

Use cases

SigFW - Open Source SS7/Diameter firewall

Use cases

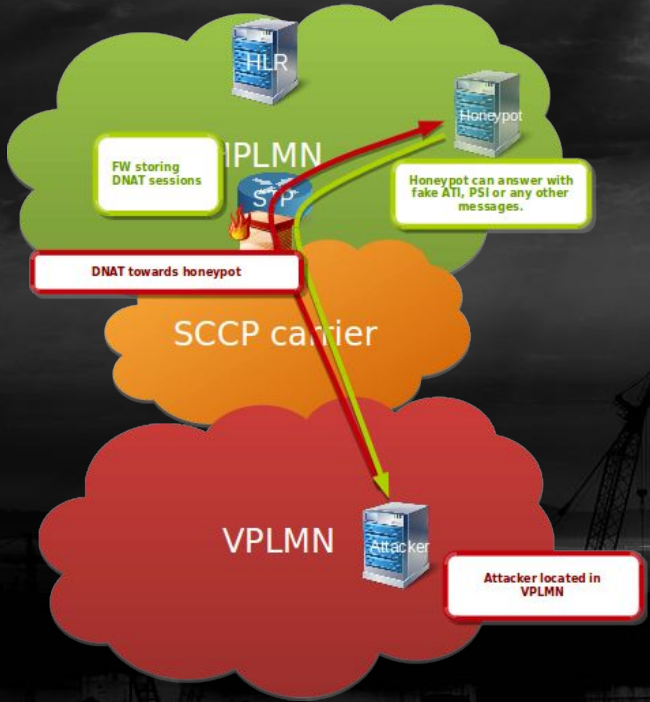
Encrypted and integrity protected signalling traffic



SigFW - Open Source SS7/Diameter firewall

Use cases

DNAT of attacker to honeypot



Related Open Source contributions

Related Open Source contributions
tshark + Elasticsearch

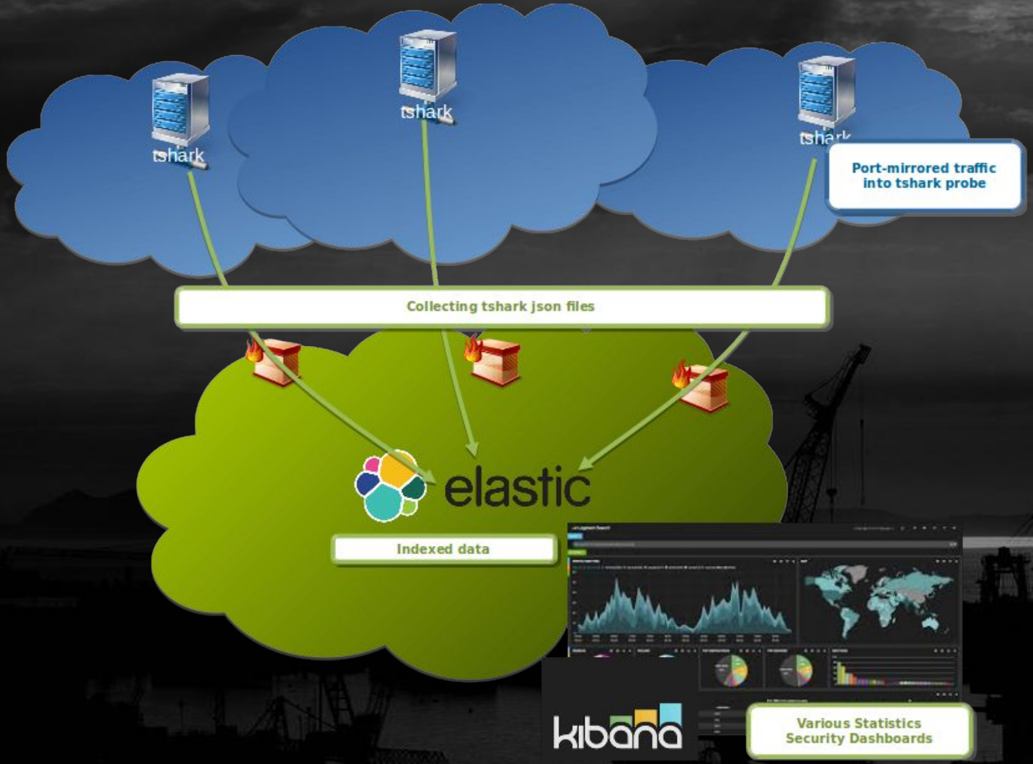
Could be used as light monitoring and analytics solution

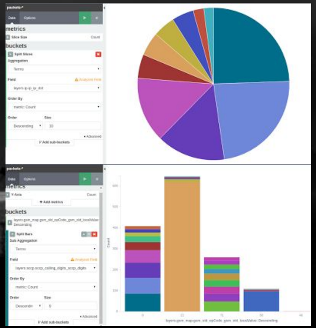
Easier detected signalling attacks targeting HPLMN users could be monitored directly in Kibana dashboard

Also applicable to different technology domains

Related Open Source contributions

tshark + Elasticsearch





Conclusion

Source code

Source code is available at

https://github.com/P1sec/SigFW

For more details also read BlackHat whitepaper

Conclusion
VM

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

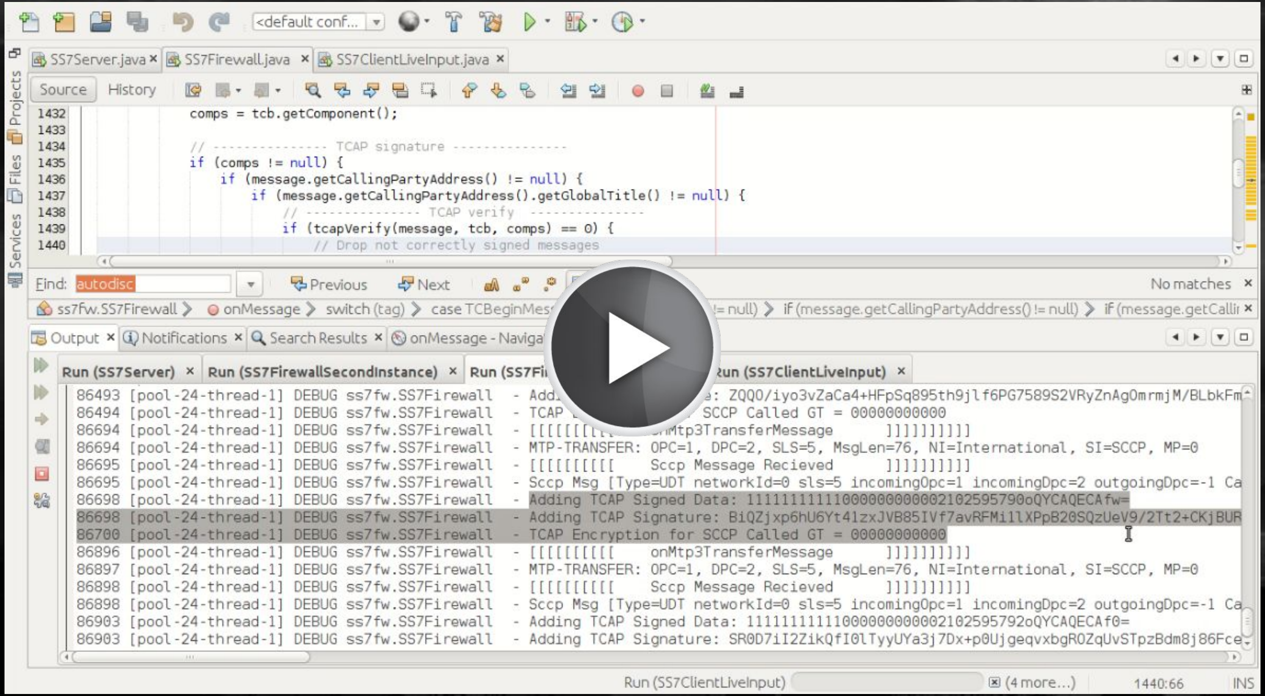
Follow up / Next steps

- Review by security researchers
- Review by industry
- Possible wider adoption
- Move towards standardization of the used extensions
- Better and multiple encryption models

We would like to encourage everyone to contribute.

Conclusion

Video Example



https://github.com/P1sec/SigFW/blob/master/docs/running_from_netbeans.gif

Work done thanks to

Special thanks to:

Open Source projects

- Telestax jSS7, jDiameter
- Wireshark
- Elastic (ELK)

Conclusion

Q&A

- Thank you

Takeaways

- **Open-source SS7/Diameter firewall**
  (the reference implementation and the research project)
- **Encryption and integrity protection of the signalling**
- **mThreat reporting and forwarding the attacker to honeypot**