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
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
“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-roamer).

MAP Cat3 messages are messages which should be allowed towards HPLMN from own subscribers in roaming (outbound-roamer) 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

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMS</td>
<td>Antispoofing, Inbound PDP detection</td>
</tr>
<tr>
<td>MAP (IMSI, HSISDN, SMSC...)</td>
<td>MAP Cat3 Time Lengthen Check screening, MAP Cat4 Previous Location Check screening</td>
</tr>
<tr>
<td>TCAP (TID message correlations)</td>
<td>MAP Cat3 SMS Home Routing, MAP Cat Special Messages</td>
</tr>
<tr>
<td>SCCP (Gb)</td>
<td>MAP Cat22 screening</td>
</tr>
<tr>
<td>MAP (IMSI, HSISDN)</td>
<td>MAP Cat33 screening, MAP Cat34 screening, MAP Cat35 screening</td>
</tr>
<tr>
<td>SCCP (Gb)</td>
<td>SCCP screening</td>
</tr>
<tr>
<td>TCAP (AC, DC)</td>
<td>SCCP (Gb)</td>
</tr>
<tr>
<td>SCCP (GTP, SSN4)</td>
<td>MTP3/MEIA (PCi)</td>
</tr>
</tbody>
</table>

- MAP Cat3 Time Lengthen Check screening: Blocking Cat3 messages with a discrepancy in location or time of location check or previous 3GPP GTP checks.
- MAP Cat4 Previous Location Check screening: Blocking Cat3 messages coming from different location than previous successful location update.
- MAP Cat3 SMS Home Routing: Messages violating SMS Home Routing.
- MAP Cat Special Messages: MAP Cat special messages.
- MAP Cat22 screening: To filter the Cat22 messages.
- MAP Cat33 screening: Screen the not allowed messages by decoding TCAP layer.
- MAP Cat34 screening: MAP Cat34 screening.
- MAP Cat35 screening: MAP Cat35 screening.
- SCCP screening: Blacklist of internal GT range (GT spoofing protection).
- MTP3/MEIA (PCi) screening: Blacklist of internal PCi (PCi spoofing protection).
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
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

Profile manipulation in VLR
- Change of MSISDN
- Provisioning of e.g. Camel IR platform
- Provisioning the CTP
- Possible to change other attributes in VLR profile
Advanced signalling attacks
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
In Cat3 there is also manipulation of GRPS profile in SGSN or MME.

- MT SMS and MT Call interception or targeted MT DoS
Advanced signalling attacks
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.
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 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.*
SigFW
Open SS7 Firewall
Open SS7 Firewall
Well positioned for signaling security enhancement

- Antispoof SS7 Signature
- Antisniff SS7 Encryption
- Antispoof Diameter Signature
- Antisniff Diameter Encryption

Future:
- GTP, IMS/SIP
...

SS7 Firewall
Diameter Firewall

Firewall Engine
Alerting
API
Threat Hunt
Collab

Linux & Open Source software
Open SS7 Firewall

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 honeypotting
- 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

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
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 Firewall

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": "MIGfMA0GCSqGSIb3DQEBADCBiQKBgQCm/PAsXOj7cjrJaQsIfHauFNlwrIuMlbrkUm3aVXeraDsiej2BWXmWlKxMxy/FR2h4Qhe9mUy6YgwT08PdMDRWMx8vxxJFI7HPJpsNfcBykefSghr5x4h6HyQr63V800U5PtgCBuVoyuOFIj84WFwlaLuajHiQgps7NOloeh1WlIDAQAB"
        }
    ],
    "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 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

<table>
<thead>
<tr>
<th>Frame 25803: 170 bytes on wire (1366 bits), 178 bytes captured (1366 bits) on interface 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet II, Src: 80:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)</td>
</tr>
<tr>
<td>Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1</td>
</tr>
<tr>
<td>Stream Control Transmission Protocol, Src Port: 2583 (2583), Dst Port: 1538 (1538)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Transaction Capabilities Application Part</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message fragments (264 bytes): #25893(229), #25900(35)</td>
</tr>
</tbody>
</table>
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

- Invoke MAP
  - Key Autodiscovery start
    - Invoke Autodiscovery (OC 69)
      - Key Autodiscovery end
        - FW1: Updated <Gi prefix, Public key> table
  - FW1: Called GT matched with <Gi prefix, Public Key>
    - FW2: Called GT matched with <Gi prefix, Private Key>
      - Sending back GT prefix + Public Key
      - Result Autodiscovery (OC 99)
      - Invoke MAP
  - FW2: Called GT matched with <Gi prefix, Private Key>
      - Sending back GT prefix + Private Key
      - Result Autodiscovery (OC 99)
      - Encrypt TCAP
        - FW1: Called GT matched with <Gi prefix, Public Key>
          - Invoke MAP

© 2017 – P1 Security, All Rights Reserved
## Open SS7 Firewall

### TCAPsec comparison

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

<table>
<thead>
<tr>
<th>Time</th>
<th>Start Time</th>
<th>End Time</th>
<th>Source</th>
<th>Destination</th>
<th>Port</th>
<th>Protocol</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017-04-07</td>
<td>2345</td>
<td>2345</td>
<td>3433 1111111111</td>
<td>MSC (Mobile Switching Ce.)</td>
<td>0000000000</td>
<td>MSC (M.</td>
<td>GSN MAP</td>
</tr>
<tr>
<td>2017-04-07</td>
<td>2344</td>
<td>2344</td>
<td>3433 1111111111</td>
<td>MSC (Mobile Switching Ce.)</td>
<td>0000000000</td>
<td>MSC (M.</td>
<td>GSN MAP</td>
</tr>
<tr>
<td>2017-04-07</td>
<td>2344</td>
<td>2344</td>
<td>3433 1111111111</td>
<td>MSC (Mobile Switching Ce.)</td>
<td>0000000000</td>
<td>MSC (M.</td>
<td>GSN MAP</td>
</tr>
<tr>
<td>2017-04-07</td>
<td>2344</td>
<td>2344</td>
<td>3433 1111111111</td>
<td>VLR (Visitor Location Reg.)</td>
<td>0000000000</td>
<td>VLR (H.</td>
<td>GSN MAP</td>
</tr>
<tr>
<td>2017-04-07</td>
<td>2344</td>
<td>2344</td>
<td>3433 1111111111</td>
<td>VLR (Visitor Location Reg.)</td>
<td>0000000000</td>
<td>VLR (H.</td>
<td>GSN MAP</td>
</tr>
<tr>
<td>2017-04-07</td>
<td>2344</td>
<td>2344</td>
<td>3433 1111111111</td>
<td>VLR (Visitor Location Reg.)</td>
<td>0000000000</td>
<td>VLR (H.</td>
<td>GSN MAP</td>
</tr>
<tr>
<td>2017-04-07</td>
<td>2344</td>
<td>2344</td>
<td>3433 1111111111</td>
<td>VLR (Visitor Location Reg.)</td>
<td>0000000000</td>
<td>VLR (H.</td>
<td>GSN MAP</td>
</tr>
<tr>
<td>2017-04-07</td>
<td>2344</td>
<td>2344</td>
<td>3433 1111111111</td>
<td>VLR (Visitor Location Reg.)</td>
<td>0000000000</td>
<td>VLR (H.</td>
<td>GSN MAP</td>
</tr>
<tr>
<td>2017-04-07</td>
<td>2344</td>
<td>2344</td>
<td>3433 1111111111</td>
<td>VLR (Visitor Location Reg.)</td>
<td>0000000000</td>
<td>VLR (H.</td>
<td>GSN MAP</td>
</tr>
</tbody>
</table>

### DNAT to new called GT

<table>
<thead>
<tr>
<th>Time</th>
<th>Start Time</th>
<th>End Time</th>
<th>Source</th>
<th>Destination</th>
<th>Port</th>
<th>Protocol</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017-04-07</td>
<td>2345</td>
<td>2345</td>
<td>3433 1111111111</td>
<td>HLR (Home Location Reg.)</td>
<td>333333333333</td>
<td>HLR (M.</td>
<td>GSN MAP</td>
</tr>
<tr>
<td>2017-04-07</td>
<td>2344</td>
<td>2344</td>
<td>3433 1111111111</td>
<td>HLR (Home Location Reg.)</td>
<td>333333333333</td>
<td>HLR (M.</td>
<td>GSN MAP</td>
</tr>
<tr>
<td>2017-04-07</td>
<td>2344</td>
<td>2344</td>
<td>3433 1111111111</td>
<td>HLR (Home Location Reg.)</td>
<td>333333333333</td>
<td>HLR (M.</td>
<td>GSN MAP</td>
</tr>
<tr>
<td>2017-04-07</td>
<td>2344</td>
<td>2344</td>
<td>3433 1111111111</td>
<td>HLR (Home Location Reg.)</td>
<td>333333333333</td>
<td>HLR (M.</td>
<td>GSN MAP</td>
</tr>
<tr>
<td>2017-04-07</td>
<td>2344</td>
<td>2344</td>
<td>3433 1111111111</td>
<td>HLR (Home Location Reg.)</td>
<td>333333333333</td>
<td>HLR (M.</td>
<td>GSN MAP</td>
</tr>
<tr>
<td>2017-04-07</td>
<td>2344</td>
<td>2344</td>
<td>3433 1111111111</td>
<td>HLR (Home Location Reg.)</td>
<td>333333333333</td>
<td>HLR (M.</td>
<td>GSN MAP</td>
</tr>
<tr>
<td>2017-04-07</td>
<td>2344</td>
<td>2344</td>
<td>3433 1111111111</td>
<td>HLR (Home Location Reg.)</td>
<td>333333333333</td>
<td>HLR (M.</td>
<td>GSN MAP</td>
</tr>
<tr>
<td>2017-04-07</td>
<td>2344</td>
<td>2344</td>
<td>3433 1111111111</td>
<td>HLR (Home Location Reg.)</td>
<td>333333333333</td>
<td>HLR (M.</td>
<td>GSN MAP</td>
</tr>
<tr>
<td>2017-04-07</td>
<td>2344</td>
<td>2344</td>
<td>3433 1111111111</td>
<td>HLR (Home Location Reg.)</td>
<td>333333333333</td>
<td>HLR (M.</td>
<td>GSN MAP</td>
</tr>
<tr>
<td>2017-04-07</td>
<td>2344</td>
<td>2344</td>
<td>3433 1111111111</td>
<td>HLR (Home Location Reg.)</td>
<td>333333333333</td>
<td>HLR (M.</td>
<td>GSN MAP</td>
</tr>
</tbody>
</table>

**Called Party Address Length: 11**

- **Called Party address (11 bytes)**
  - Address Indicator
    - Subsystem Number: MSC (Mobile Switching Center) (8)
    - Called or Calling SubSystem Number: MSC (Mobile Switching Center) (8)
      - [Linked to TCAP, TCAP 589 linked to GSN_MAP]
      - Global Title: 0x06 (9 bytes)
      - Translation Type: 0x06
      - 0001 ... - Numbering Plan: ISDN/telephony (0x01)
        - 0003 = Encoding Scheme: BCD, odd number of digits (0x01)
        - 0000 0000 = Nature of Address Indicator: International number (0x04)

- **Called Party Digits: 3833838382**

- **Calling Party Address Length: 11**
- **Calling Party address (11 bytes)**
- **Transaction Capabilities Application Part**
- **GSN Mobile Application**

---

**blackhat USA 2017**

© 2017 – P1 Security, All Rights Reserved
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 honeypotting
- 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
Frames are forwarded on SCTP layer
Filtering is possible up to the application layer (Diameter layer)
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

Request Diameter

FW1: Dest Realm matched with <Realm,Public Key>

Encrypted Diameter

FW2: Dest Realm matched with <Realm,Private Key>

Request Diameter
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 82.020884296 127.0.0.1 127.0.0.1 DIAMET. 482 cmd=3GPP-Notify Request(23) flags=- --- appl=3GPP S6a/S8d[16777251] h2h=4a403677c e2e=65fd05011
148 82.020884296 127.0.0.1 127.0.0.1 DIAMET. 1391 SACK cmd=3GPP-Notify Answer(32) flags=- --- appl=3GPP S6a/S8d[16777251] h2h=4a403677c e2e=65fd05011
151 82.020884296 127.0.0.1 127.0.0.1 DIAMET. 1394 SACK cmd=3GPP-Notify Answer (32) flags=- --- appl=3GPP S6a/S8d[16777251] h2h=4a403677c e2e=65fd05011
155 82.020884296 127.0.0.1 127.0.0.1 DIAMET. 426 SACK cmd=3GPP-Notify Answer(32) flags=- --- appl=3GPP S6a/S8d[16777251] h2h=4a403677c e2e=65fd05011
156 82.020884296 127.0.0.1 127.0.0.1 DIAMET. 410 SACK cmd=3GPP-Update-Location Request(315) flags=R --- appl=3GPP S6a/S8d[16777251] h2h=4a403677c e2e=65fd05011

Flags: Ox80, Request
Command Code: 316 3GPP-Update-Location
ApplicationId: 3GPP S6a/S6d (16777251)
Msg-Hy-Hop-Identifier: Ox4ad9277c
End-To-End-Identifier: Ox07800014

[Answer In: 198]

AVP: Session-ID(23) l=48 f=M- value=createdbydiameterliveclient;1493747598087

AVP: Destination-Host(23) l=28 f=M- value=37.0.0.1:13866

AVP: Unknown(100) l=115 f=M- value=09062d79b3e2e7115e34492170f0d3acc4d5355f8f

AVP: Origin-Realm(254) l=59 f=M- value=

AVP: Unknown(100) l=136 f=M- value=79405b0d9c60d20770ffbadf049e870e23d73958d7

AVP: Unknown(100) l=136 f=M- value=053f73dc99e2b21cbf6c3f4ee379956bb2

AVP: Unknown(100) l=136 f=M- value=0fa1c467e402c0608c3f4f65b20b336f1

AVP: Unknown(100) l=136 f=M- value=51f62a2af5316853fe8c5233692e2c47f8e8e8b7d9f

AVP: Unknown(100) l=136 f=M- value=51f62a2af5316853fe8c5233692e2c47f8e8e8b7d9f

AVP: Destination-Realm(23) l=28 f=M- value=exchange.example.org

AVP: Unknown(296) l=34 f=M- value=exchange.example.org

AVP: Unknown(100) l=136 f=M- value=7290d3298d0ff4c05d0f0e4b06547f49fe273f3ce8a43b48
Open Diameter Firewall
Diameter Encryption Autodiscovery

Request Diameter

FW#1: Dest Realm not matched with <Realm, Public Key>
Request Diameter

Key Autodiscovery start
Request Autodiscovery (CC 999)

FW#2: Dest Realm matched with <Realm, Private Key>
Sending back:
Realm = Public Key

Result Autodiscovery (CC 999)

Key Autodiscovery end
FW#1: Updated <Realm, Public key> table
Request Diameter

FW#1: Dest Realm matched with <Realm, Public Key>
Encrypted Diameter

FW#2: Dest Realm matched with <Realm, Private Key>
Request Diameter
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

blackhat USA 2017
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
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
Conclusion

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
Conclusion

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