:: Positive Technologies

Creating a fuzzer for telecom protocol

4G LTE case study

Sergey Mashukov

positive-tech.com

:: SS7 now

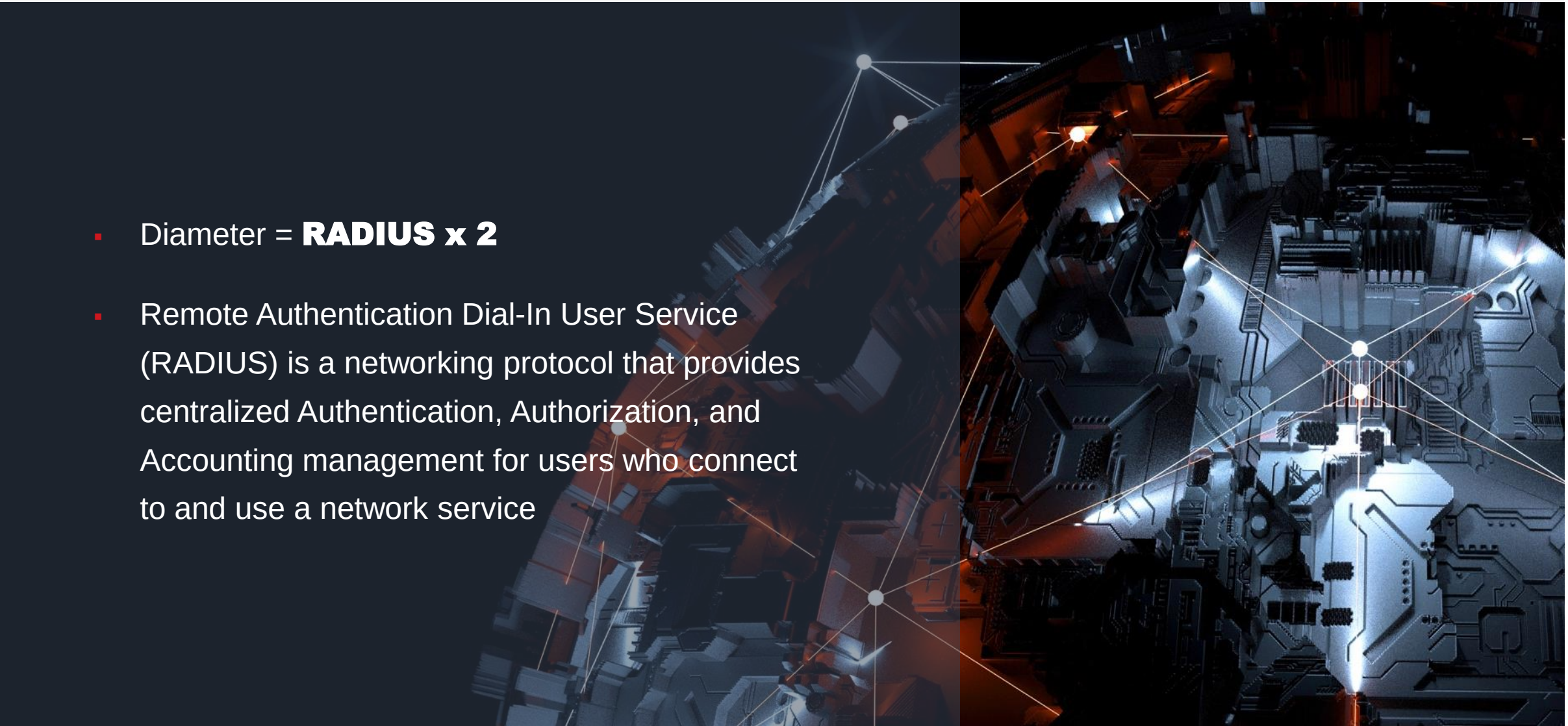
More than 50 different SS7 attacks:

- IMSI disclosure
- Location discovery
- Subscriber DoS
- SMS interception and spoofing
- Call interception
- Reading of Telegram and WhatsApp chats

:: Diameter now

	SS7	Diameter
Interception	+	+
Tracking	+	+
DoS on subscriber	+	+
DoS on network equipment	+	+
Fraud	+	+

:: Diameter

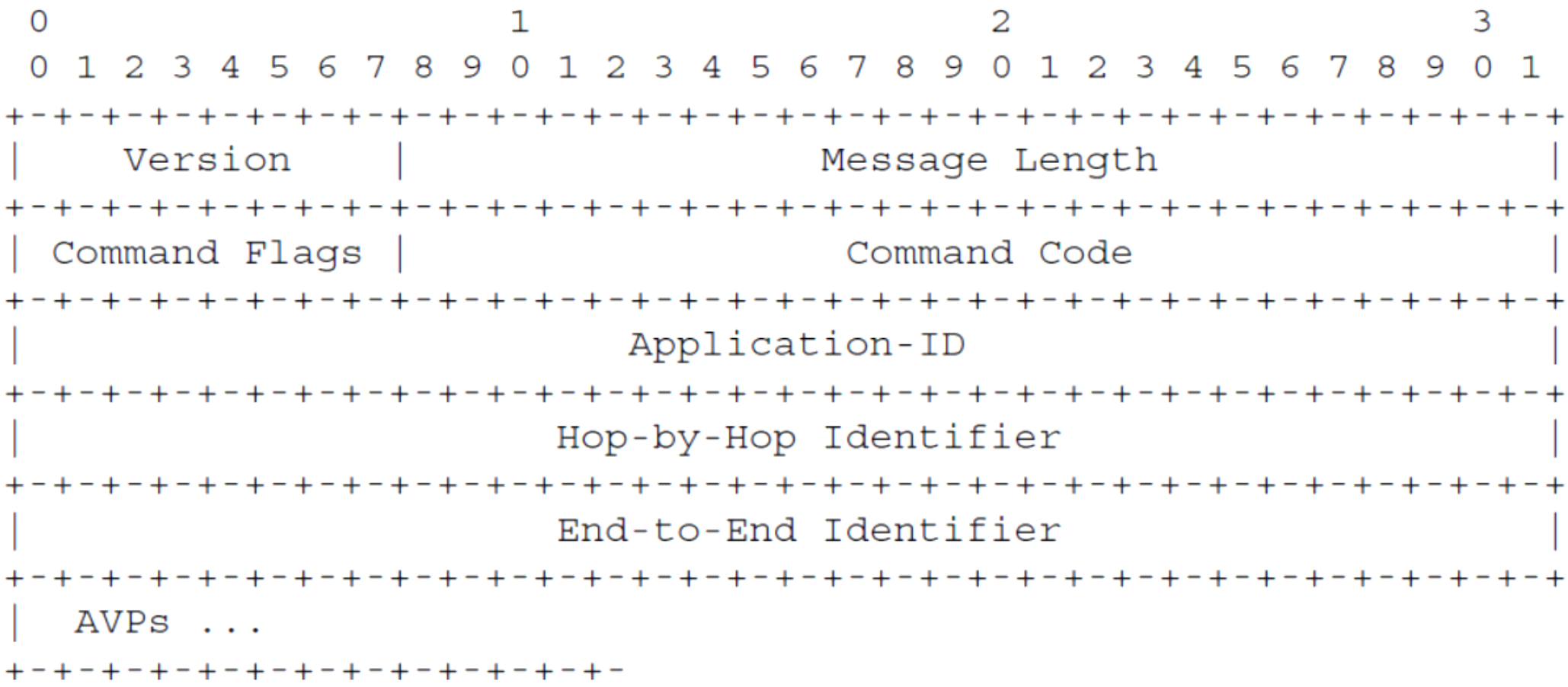


- Diameter = **RADIUS x 2**
- Remote Authentication Dial-In User Service (RADIUS) is a networking protocol that provides centralized Authentication, Authorization, and Accounting management for users who connect to and use a network service

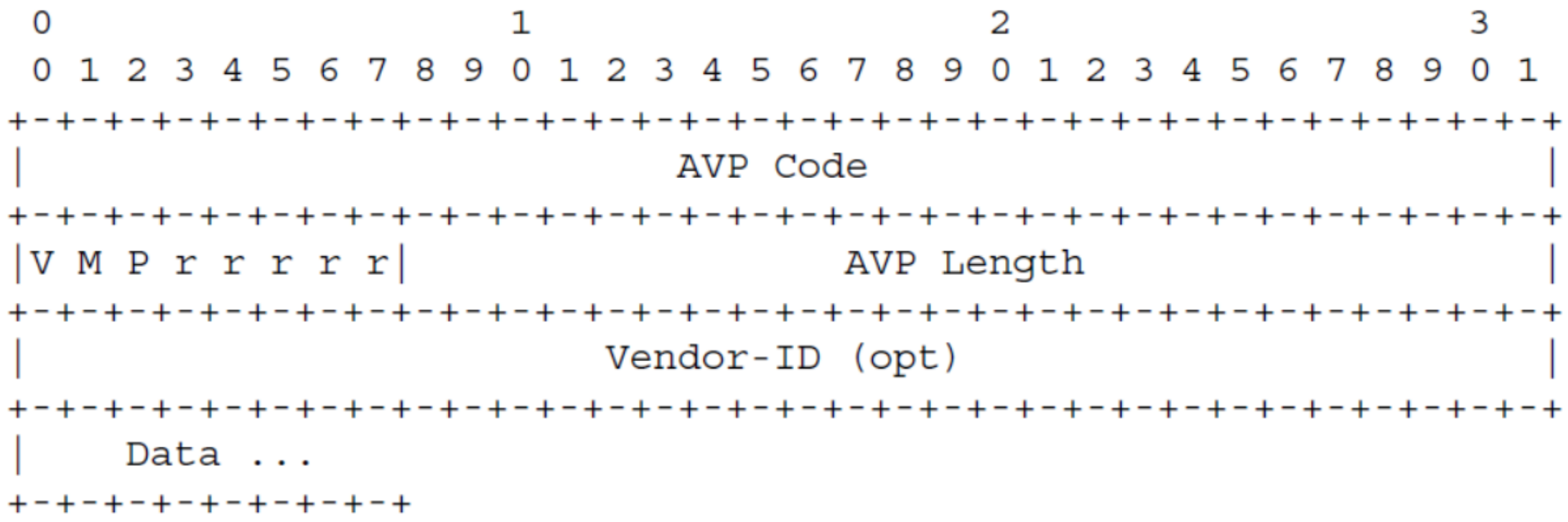
:: Diameter

- **Session-layer AAA protocol**
- **Cleartext**
- **Support for SCTP or TCP**
- **IPsec or TLS/DTLS for encryption**
- **Extensibility**
  (Diameter Base and Applications on top of it)

:: Diameter header



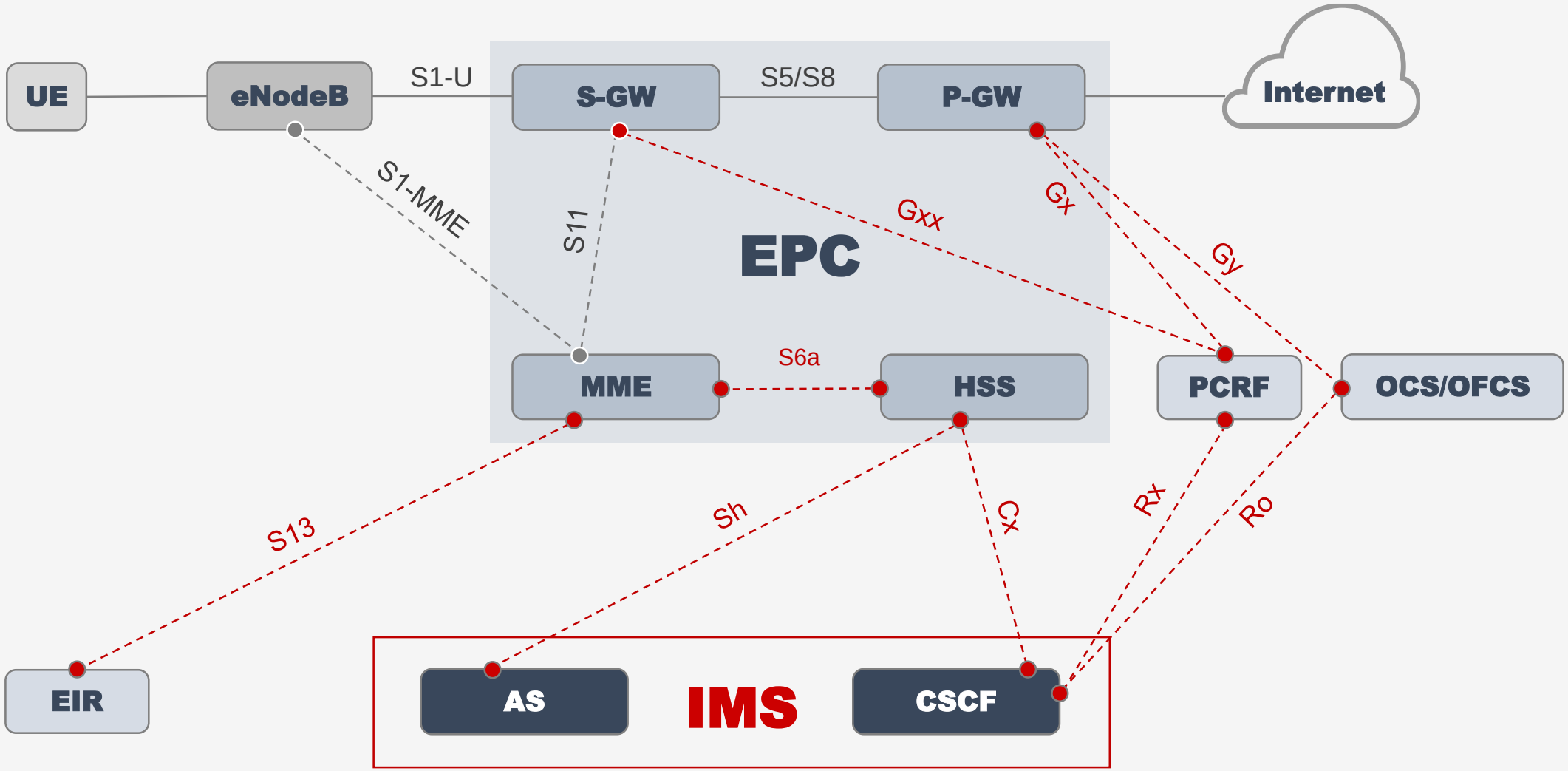
:: Diameter AVPs



:: Diameter message

```
Diameter Protocol
  Version: 0x01
  Length: 376
▷ Flags: 0xc0, Request, Proxyable
  Command Code: 319 3GPP-Insert-Subscriber-Data
  ApplicationId: 3GPP S6a/S6d (16777251)
  Hop-by-Hop Identifier: 0x01a0ceb5
  End-to-End Identifier: 0xf88301f1
▷ AVP: Session-Id(263) l=54 f=-M- val=epc.mnc001.mcc641.3gppnetwork.org;1538135923;0
▷ AVP: Vendor-Specific-Application-Id(260) l=32 f=-M-
▷ AVP: Auth-Session-State(277) l=12 f=-M- val=NO_STATE_MAINTAINED (1)
▷ AVP: Destination-Host(293) l=46 f=-M- val=mme1.epc.mnc001.mcc648.3gppnetwork.org
▷ AVP: Destination-Realm(283) l=41 f=-M- val=epc.mnc001.mcc648.3gppnetwork.org
▷ AVP: Origin-Host(264) l=45 f=-M- val=mme.epc.mnc001.mcc641.3gppnetwork.org
▷ AVP: User-Name(1) l=23 f=-M- val=648010000000001
▷ AVP: Origin-Realm(296) l=41 f=-M- val=epc.mnc001.mcc641.3gppnetwork.org
▷ AVP: IDR-Flags(1490) l=16 f=VM- vnd=TGPP val=8
▷ AVP: Subscription-Data(1400) l=32 f=VM- vnd=TGPP
```

:: Diameter in LTE

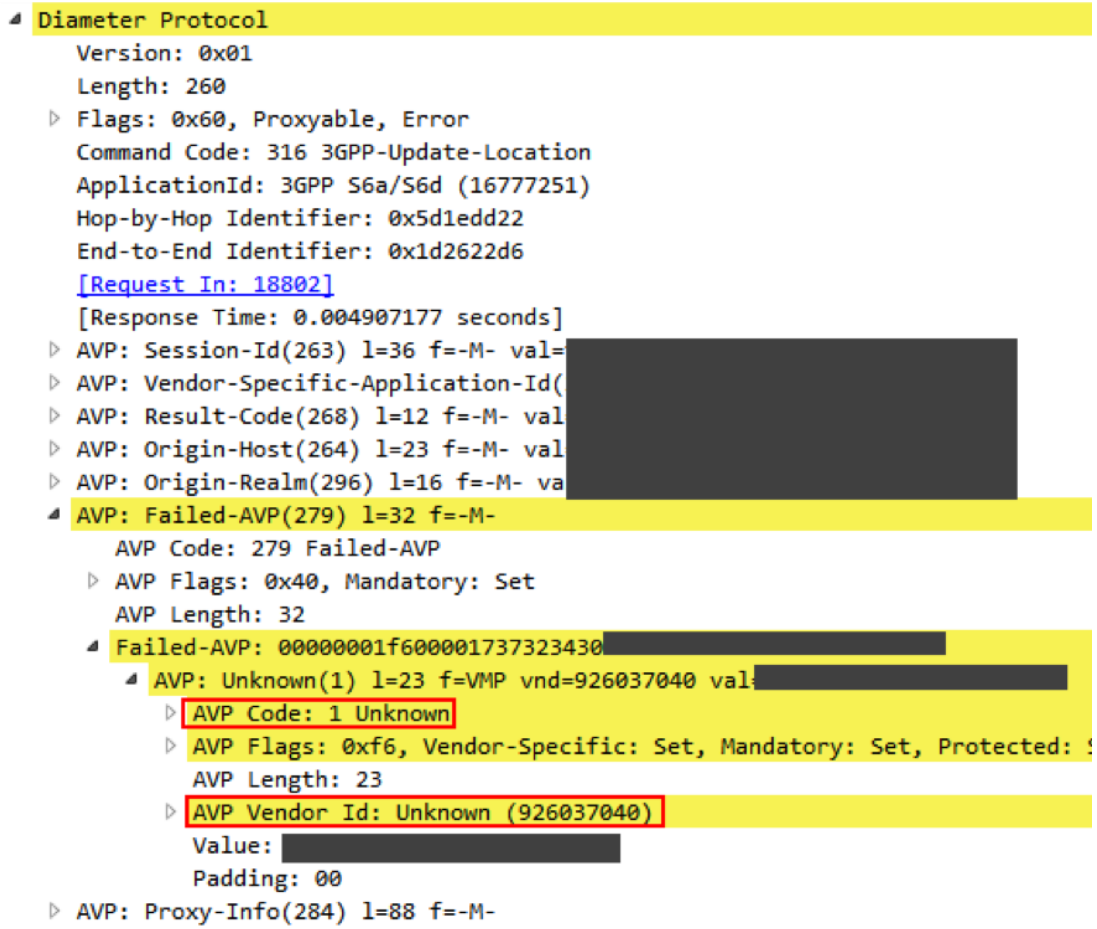


:: Fuzzing

According to Wikipedia:

"Fuzzing or fuzz testing is an automated software testing technique that involves providing invalid, unexpected, or random data as inputs to a computer program.

The program is then monitored for exceptions such as crashes, failing built-in code assertions, or potential memory leaks."



:: Fuzzing

- Software to perform the test
- A way to check and interpret the results

:: Fuzzing of telecom equipment

- Normally should be done by vendors, but often overlooked
- No access to hardware or code for security community => bugs are present
- In our experience with fuzzing assessments, more than half of tested equipment has vulnerabilities
- Bugs may lead to serious consequences

:: Correctly formed messages may cause the same impacts

- Outage on February 19, 2016
- More than 3.5 hours
- More than 1 million subscribers

:: RCE on host

- Vulnerabilities found during fuzzing may be exploited to perform **Remote Code Execution attacks**
- Successful RCE may lead to adversary **gaining control over the Network Element** to perform further attacks on this or other MNO networks

:: Where to test

:: Test lab usually is different from network

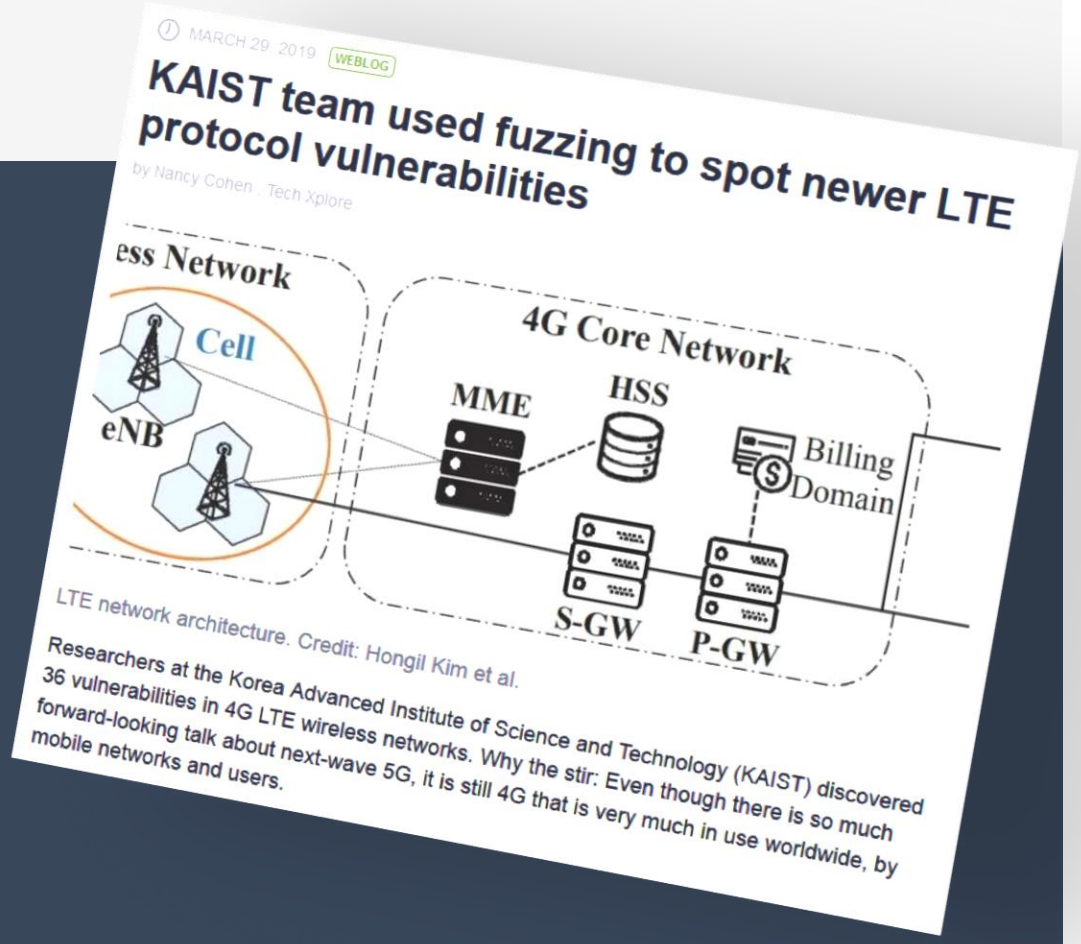
Different configuration of UE and Network Equipment:

- Non-standard identifiers are used
- Routing is always different

Nodes should be configured as in real life or else some cases won't be tested

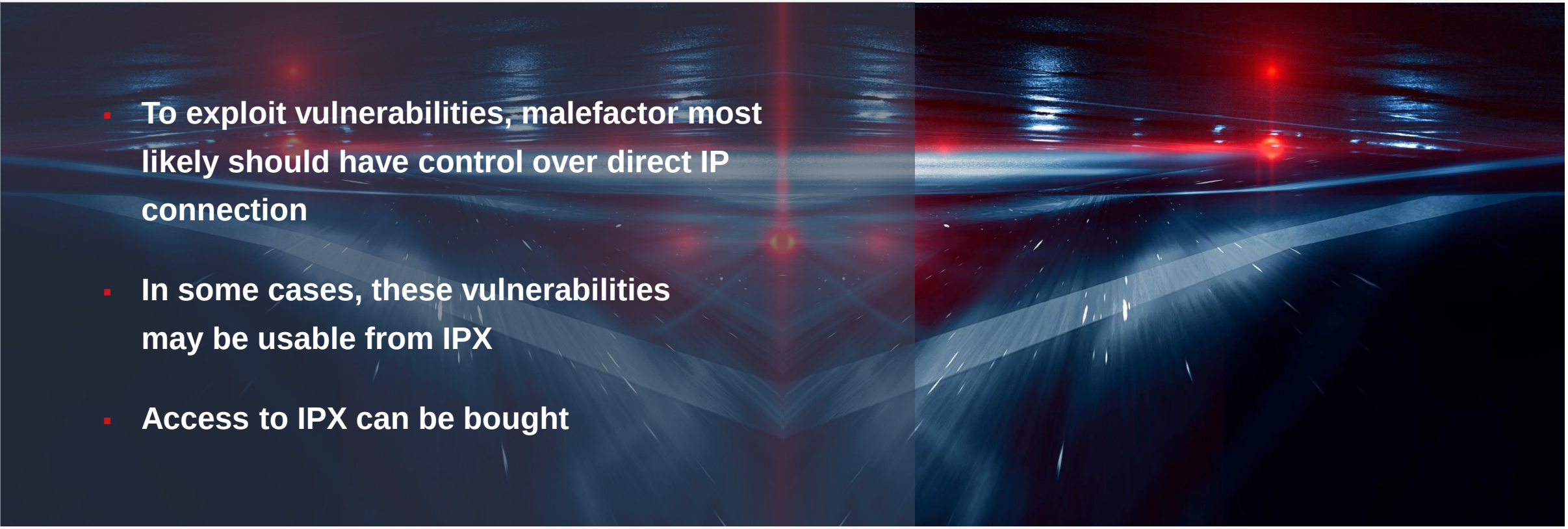
:: Recent Case

- "Touching the Untouchables: Dynamic Security Analysis of the LTE Control Plane" by KAIST
- 51 vulnerabilities were found
- Problems found through fuzzing may be overlooked by vendors
- Not all found problems may be exploitable in the wild



https://techxplore.com/news/2019-03-kaist-team-fuzzing-lte-protocol.html

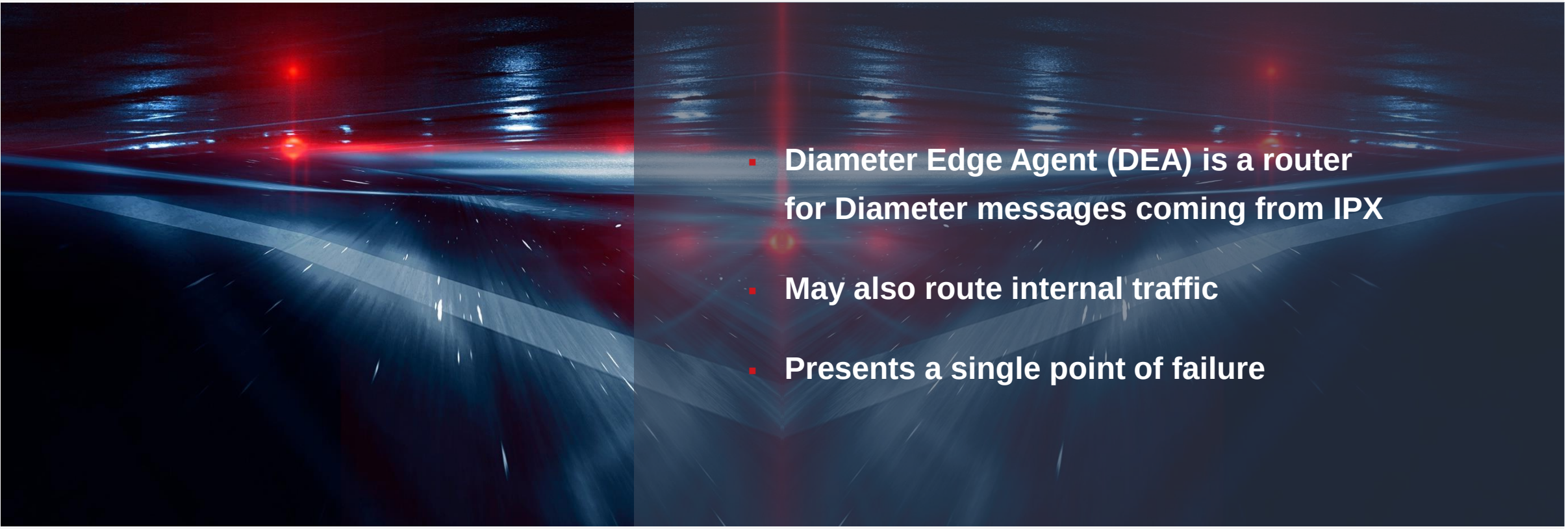
:: DEA as a single point of connection



- To exploit vulnerabilities, malefactor most likely should have control over direct IP connection
- In some cases, these vulnerabilities may be usable from IPX
- Access to IPX can be bought

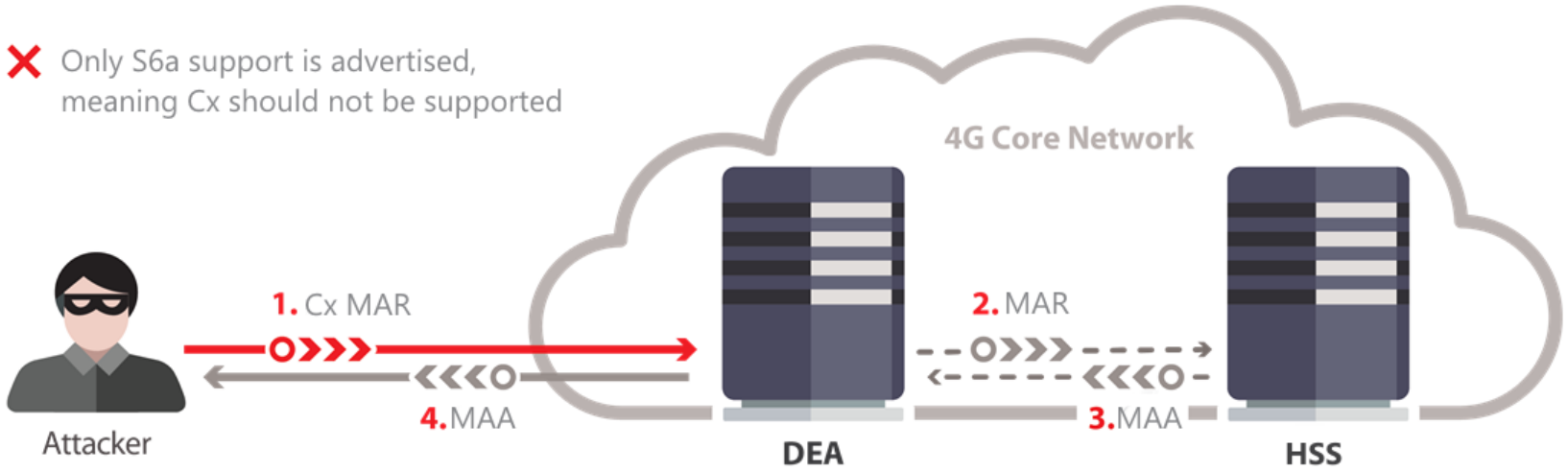
https://www.thedailybeast.com/you-can-spy-like-the-nsa-for-a-few-thousand-bucks

:: DEA as a single point of connection



- Diameter Edge Agent (DEA) is a router for Diameter messages coming from IPX
- May also route internal traffic
- Presents a single point of failure

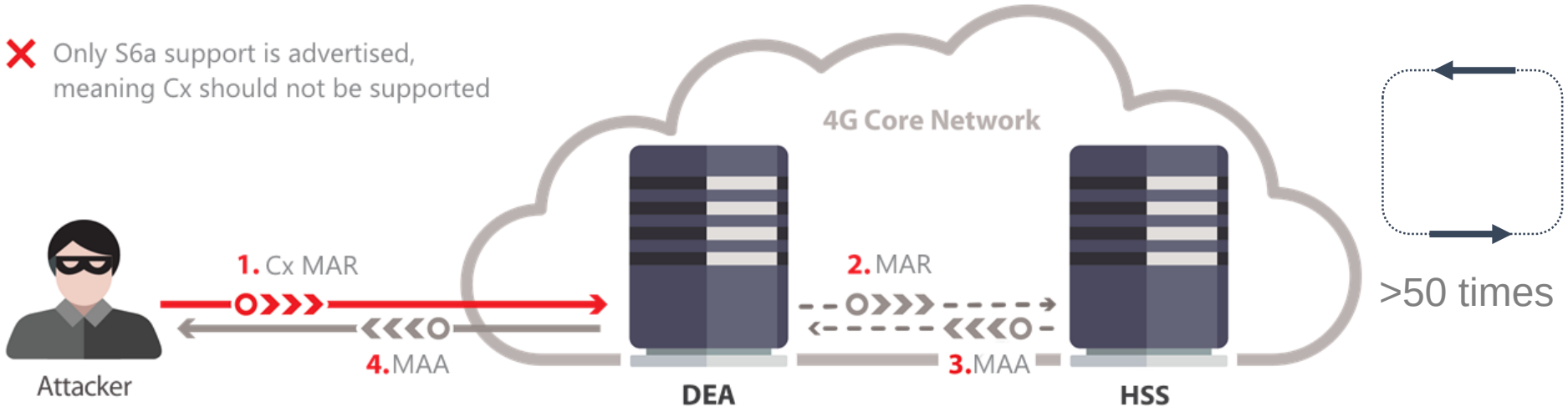
:: DEA ignoring configuration



Messages:

MAR – Multimedia-Auth-Request (Cx)
MAA – Multimedia-Auth-Answer (Cx)

:: Attack on HSS through the DEA



- Diameter connection is dropped on HSS
- Burst of messages
- Works both directly and through DEA

:: Why separate telecom fuzzer is needed

- Need to communicate with tested equipment through network
- Specific message structure and data types
- Having source code allows flexibility on-site if new functionality is needed

:: Existing protocol implementations

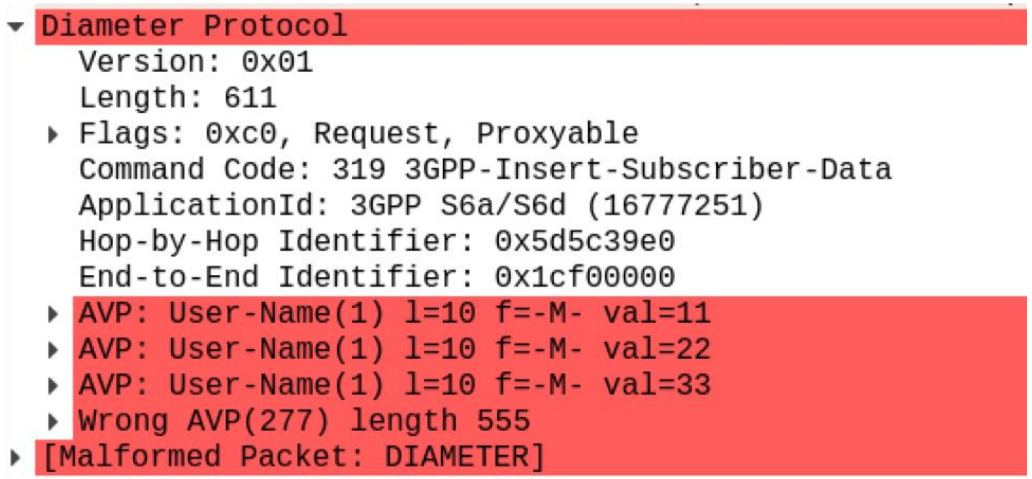
Problems with:

- Creating malformed messages
- Breaking correct message order
- Testing on Diameter Base level
  - Connection establishment
  - Answers

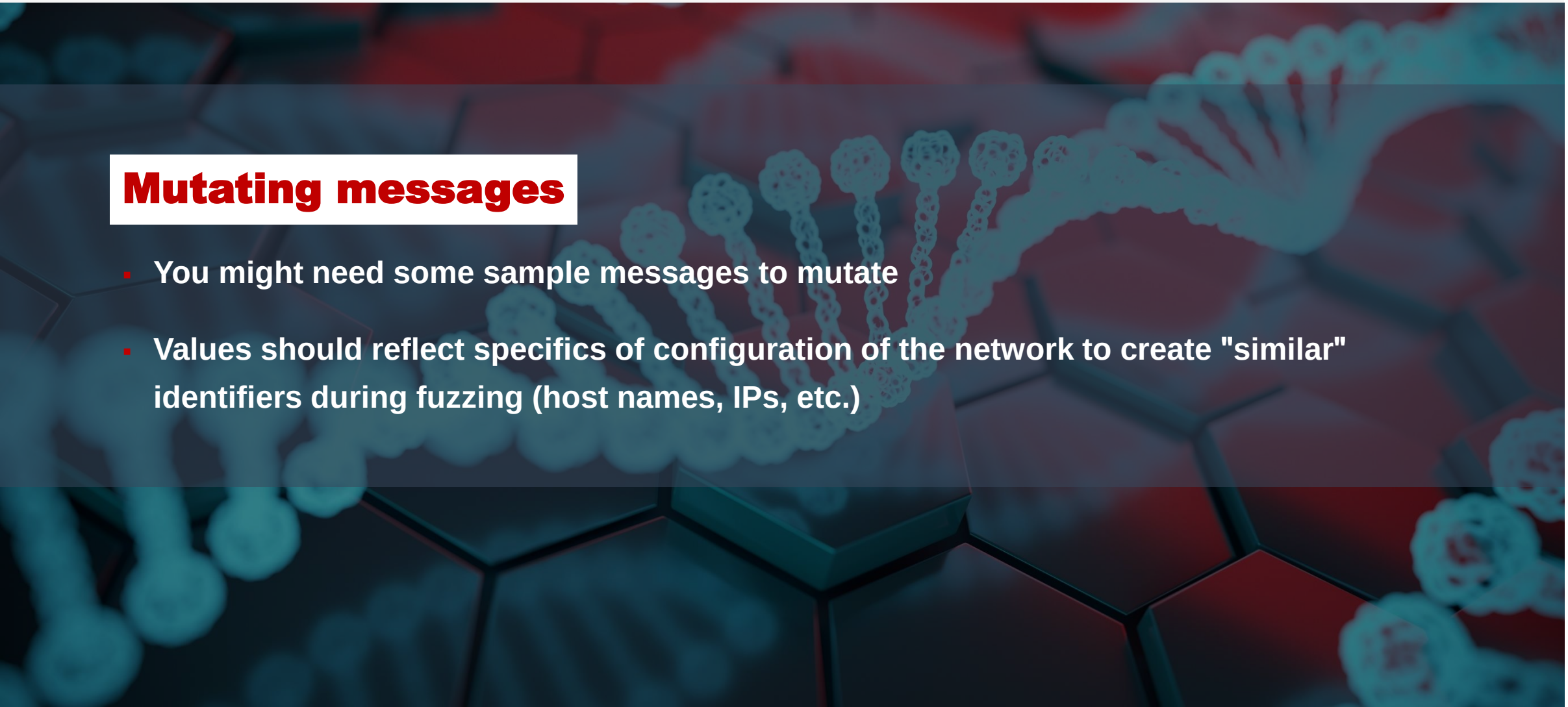
Usually commercial protocol stacks are not suited for fuzzing

:: How to fuzz

- **Two kinds of malformed messages:**
  - Wrong from encoding perspective
  - Wrong from semantics perspective (e.g., fields that should not be present in the message)
- **You need messages that are somewhat similar to "real" ones to cover both types**
- **You need to isolate the problem to report and fix it**



:: How to fuzz



Mutating messages

- You might need some sample messages to mutate
- Values should reflect specifics of configuration of the network to create "similar" identifiers during fuzzing (host names, IPs, etc.)

:: What kind of mutations to use

Mutating headers:

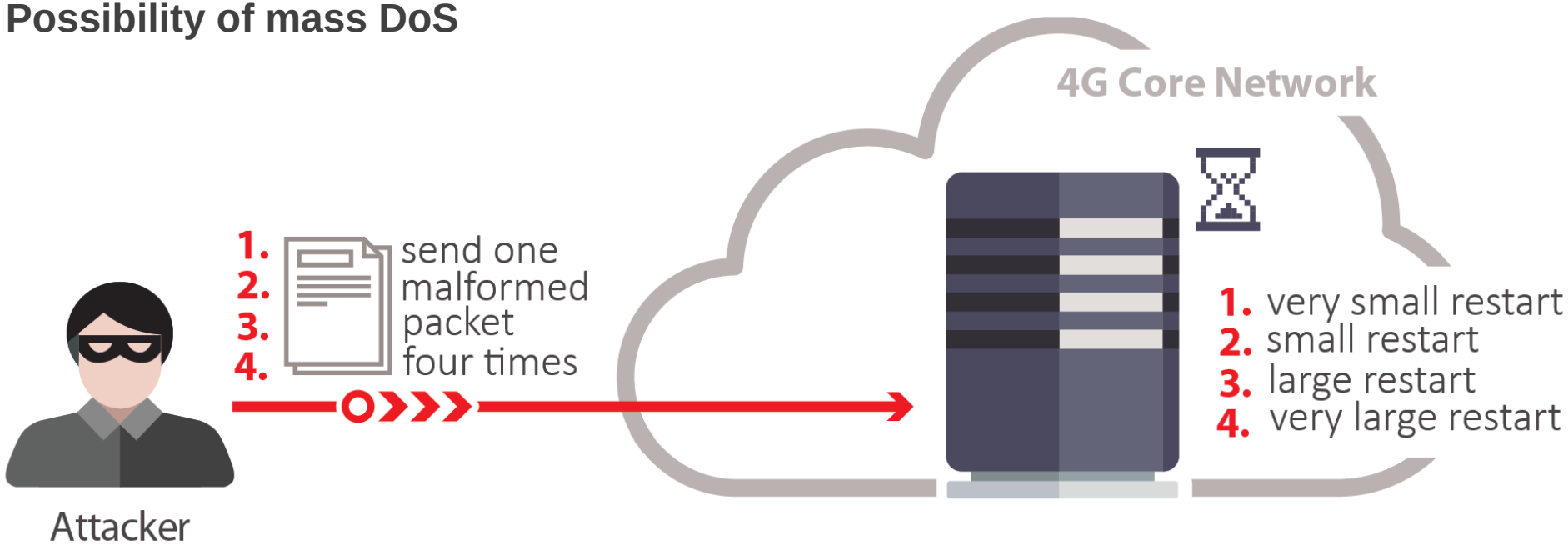
- Random bit flips
- Pre-set values

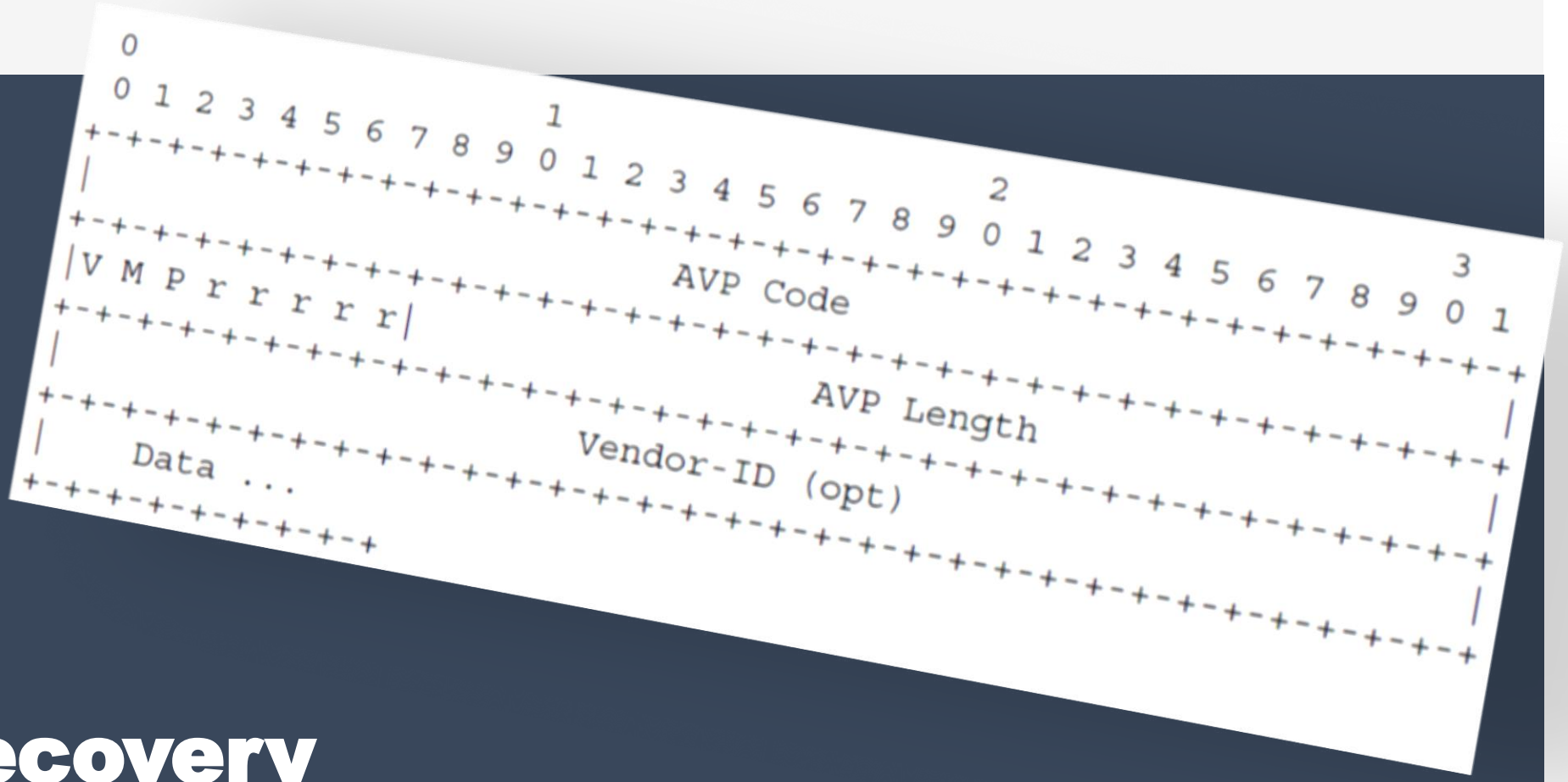
Mutating AVP values:

- Random bit flips
- Pre-set values specific to AVP type
- Random appends and removals
  for variable-length AVP types
- Changes in message structure for grouped AVPs

:: DoS: Recovery mechanism from attacker side

- MME restarts
- Problem in Diameter parsers
- Possibility of mass DoS





DoS: Recovery mechanism from attacker side

V-bit is used to determine whether the 4-byte Vendor-Id field should be present in the AVP header.

:: Blacklisting values

Adapt your fuzzer to the operating conditions:

- Test lab may be used for other tests
- It even may be connected to international network

Blacklisting of some values, since operators don't want to stop work in test lab (especially true for network elements doing routing such as Diameter Routing Agent)

:: Problems implementing the fuzzer

- It is almost impossible to do exhaustive testing due to possible number of combinations and extensibility of protocols
- So the faster we fuzz, the better

:: Speeding up the fuzzing process

Study the protocol to see where you can speed up fuzzing

Two kinds of mutations are possible for Diameter:

- Mutations that affect length of the message
  (length field in headers, data of variable-length AVPs or message structure)
- Mutations that don't affect the length

If length is not affected, there are no changes in binary structure and padding, so fuzzing may be performed on the encoded message

:: Mutations in length

- Updating all headers in the message appropriately
- Without changes to the length fields

:: DEA restarts

Nested AVPs with wrong AVP Length field in the parent

From:

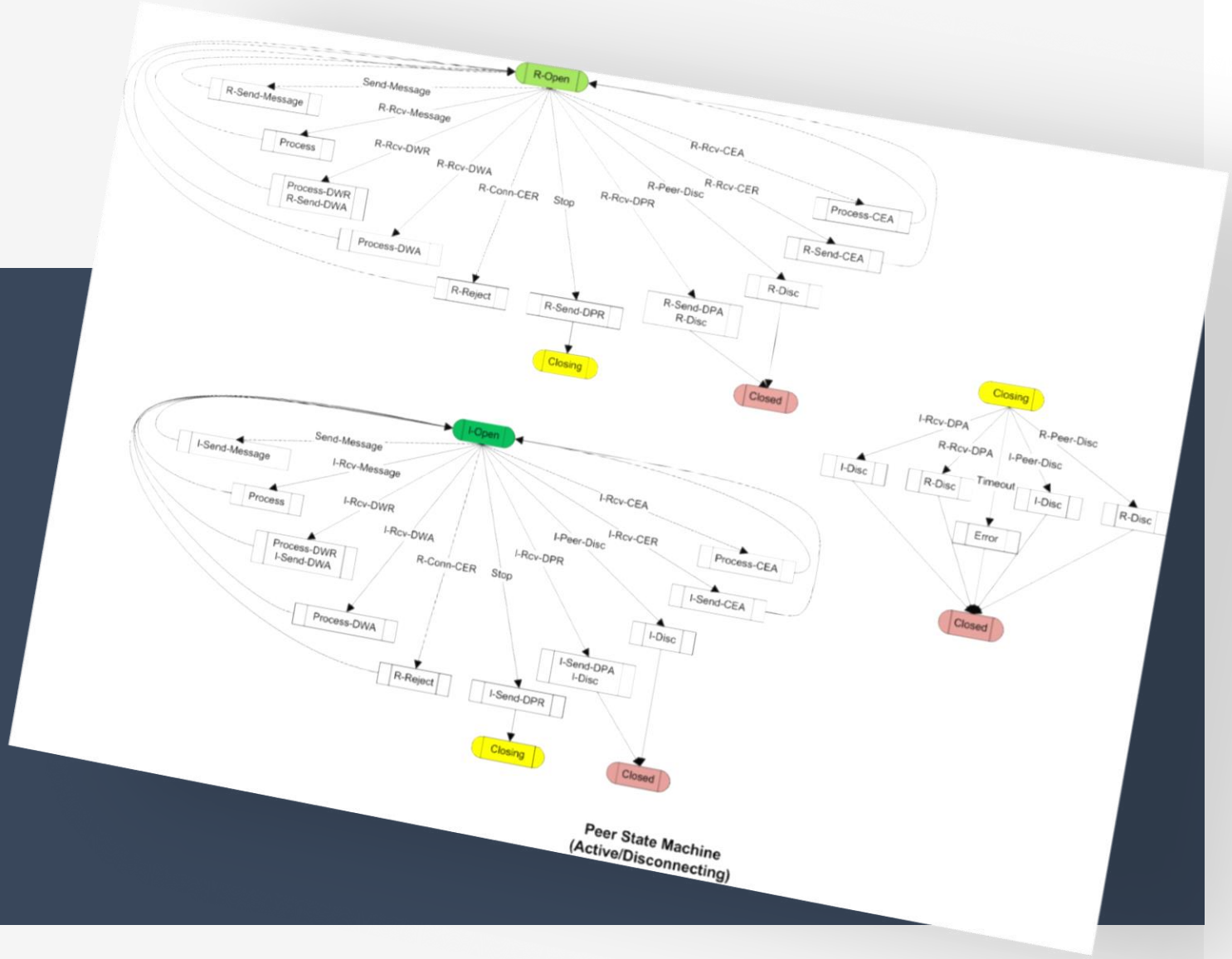
```
AVP: Vendor-Specific-Application-Id(260) l=32 f=-M-
  AVP Code: 260 Vendor-Specific-Application-Id
  AVP Flags: 0x40, Mandatory: Set
  AVP Length: 32
  Vendor-Specific-Application-Id: 0000010a4000000c000028af000001024000000c01000023
    AVP: Vendor-Id(266) l=12 f=-M- val=10415
    AVP: Auth-Application-Id(258) l=12 f=-M- val=3GPP S6a/S6d (16777251)
```

To:

```
AVP: Vendor-Specific-Application-Id(260) l=32 f=-M-
  AVP Code: 260 Vendor-Specific-Application-Id
  AVP Flags: 0x40, Mandatory: Set
  AVP Length: 0
  Vendor-Specific-Application-Id: 0000010a4000000c000028af000001024000000c01000023
    AVP: Vendor-Id(266) l=12 f=-M- val=10415
    AVP: Auth-Application-Id(258) l=12 f=-M- val=3GPP S6a/S6d (16777251)
```

:: Stateful checks

- Needed when you want to test handling of state machines
- Not very interesting when working for MNO, since most telecom Diameter interfaces don't use multi-message transactions



By Vingarzan – Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=10844303

:: Parsing error answers

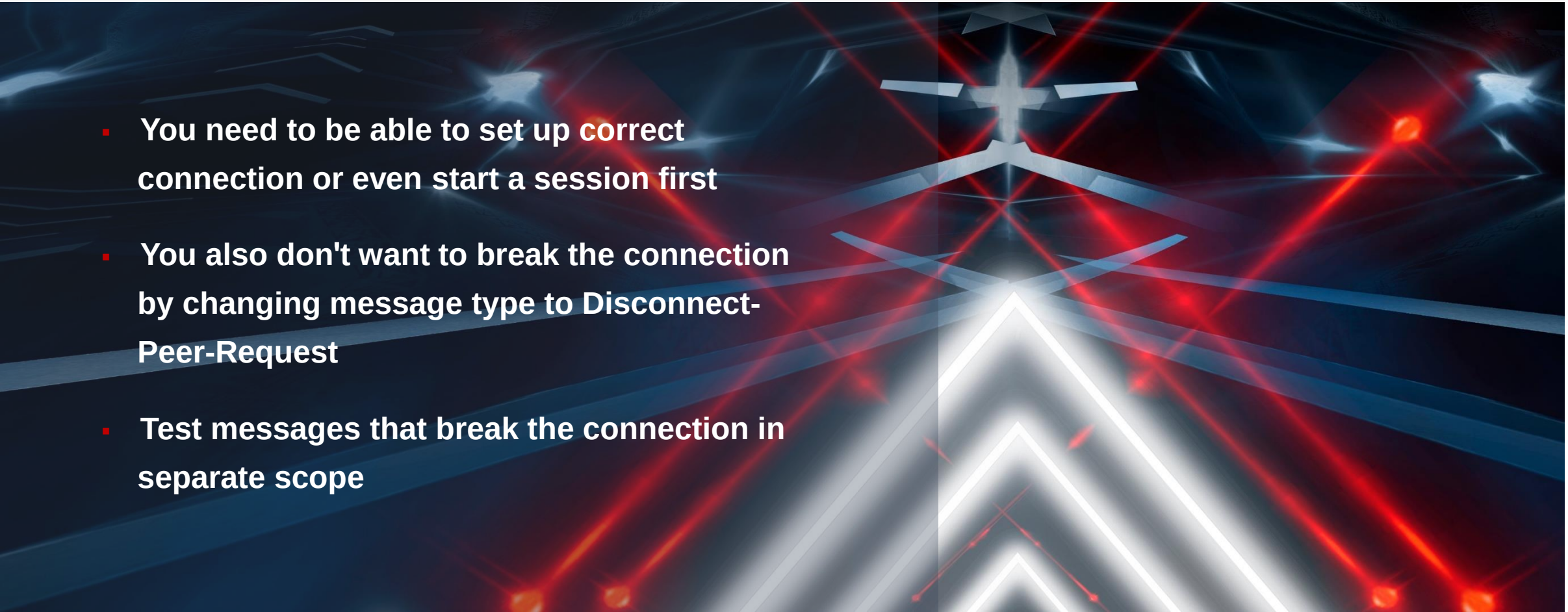
Your program may need to work both as fuzzer and as a legitimate peer

So you might **need to be able to parse the messages** that are coming back from the network

In answers, Failed-AVP may contain malformed data sent to the test node -> do not decode answer messages, or your own implementation might crash

It is better to have configurable parsing since sometimes you need to parse what is received and sometimes not

:: Handling connections and sessions



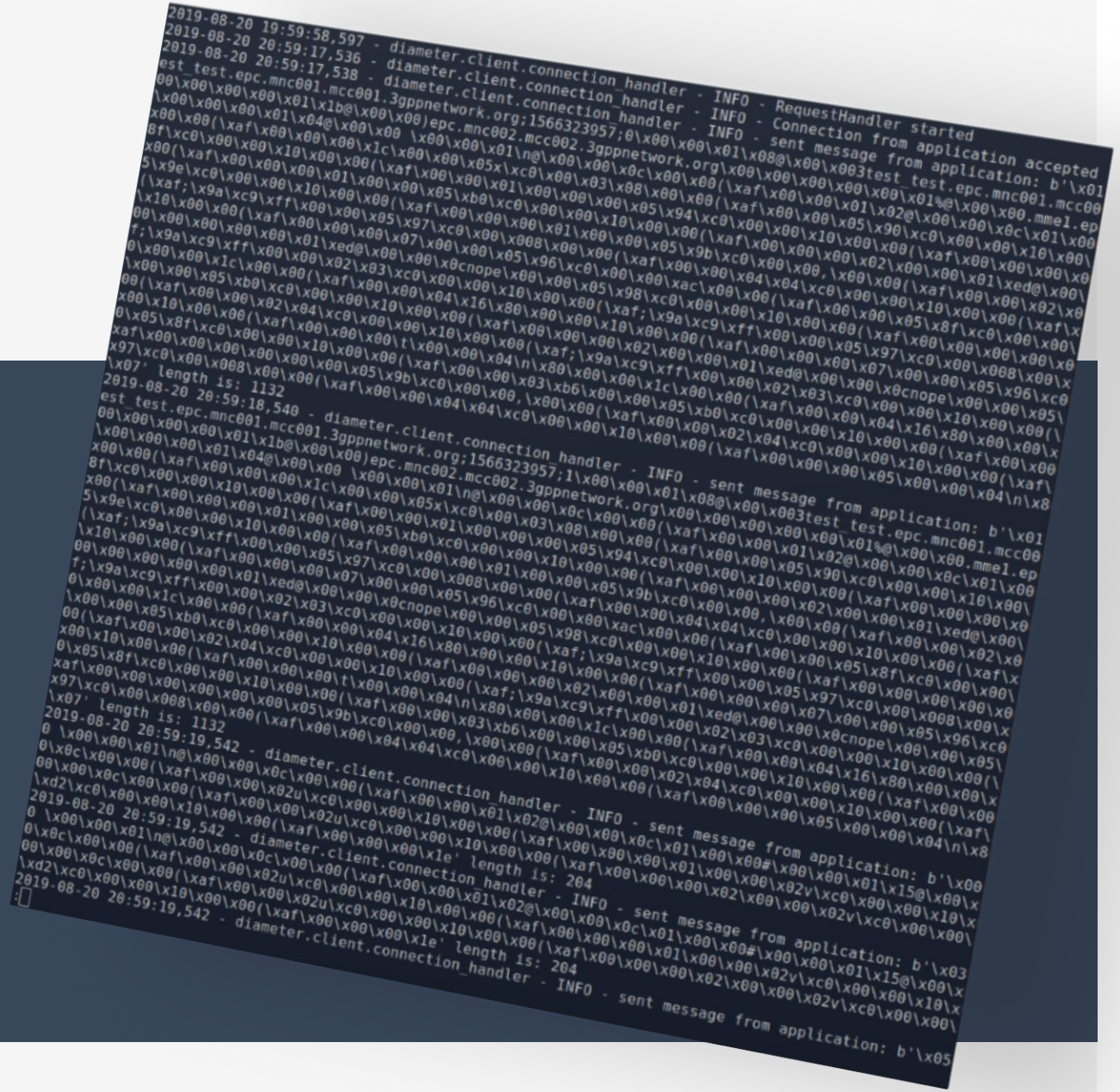
- You need to be able to set up correct connection or even start a session first
- You also don't want to break the connection by changing message type to Disconnect-Peer-Request
- Test messages that break the connection in separate scope

:: Problems implementing the fuzzer

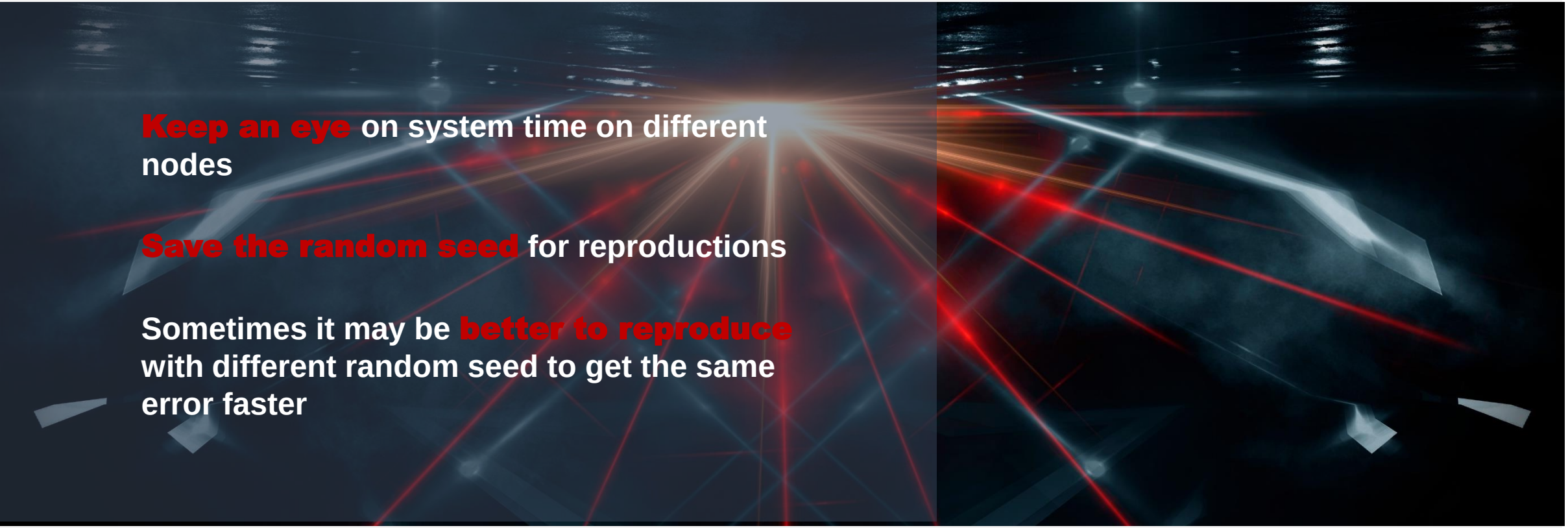
- To properly test answer messages or set up the session, you need to emulate requests at a certain rate
- You also need to update hop-by-hop and end-to-end
- Ask MNO for emulators if needed
- Create your own, it might help with development

:: Logging

- For errors you are at the mercy of the network element's own monitoring systems
- "Here is ssh to the node and two console commands. Enjoy!"
- Do log all sent messages



:: Reproducing issues



Keep an eye on system time on different nodes

Save the random seed for reproductions

Sometimes it may be **better to reproduce** with different random seed to get the same error faster

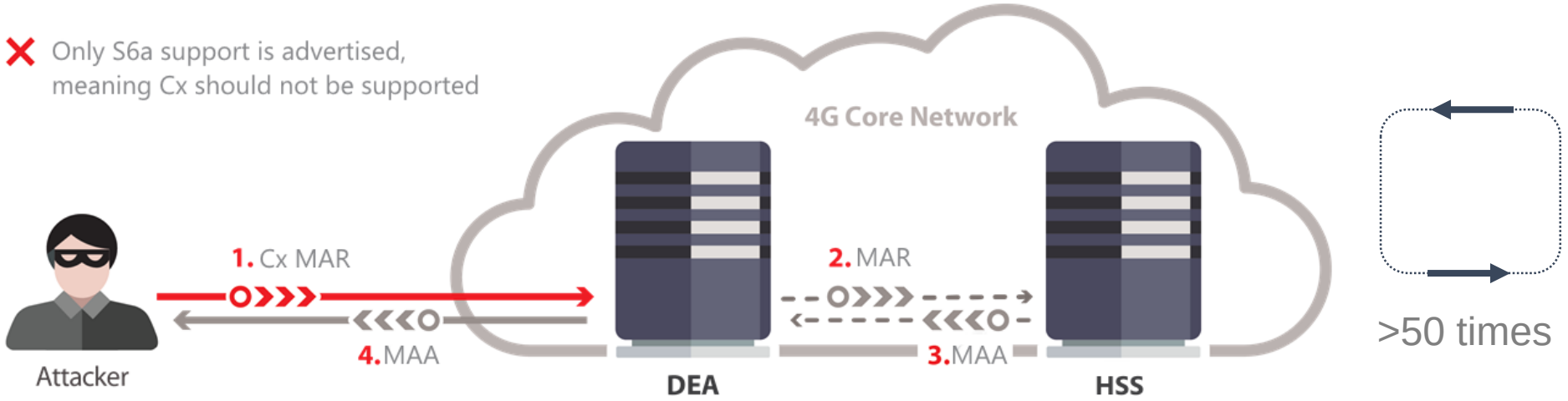
:: Reproducing issues

It is not enough to say **"something crashed"**

You need **working PoC**

Getting the PoC for Diameter is all about changing AVP content until you find the message causing the issue

:: Reproducing issues



- Reproduced during random changes in message structure
- Slowing down message sending rate "fixed" the issue
- Narrowing down types of mutations and then AVP content

:: Typical project

1. Initial contact

2. "We need this fast"

3. Several months of back and forth

4. "Get started next week"

5. Issues on-site (typhoon, public holidays not communicated, some tests can't be done in the lab, etc.)

6. Time for tests is shortened / it is not possible to do additional tests

:: How to deal with clients

- Ask for access to log systems and crash dumps
- Get access to hardware vendor's representatives (may experience pushback from them)
- Have enough time planned for investigation
- Don't concentrate on number of messages for each separate mutation to each information element — it is better to do more tests with different parts of message being altered

:: How to deal with clients

1. Present your results comprehensibly

2. Sometimes it may be very hard to evaluate the impact of the finding on the spot, ask vendor's representatives if possible

3. Have a working PoC for the issue

Takeaways

Decide if you need your own protocol implementation

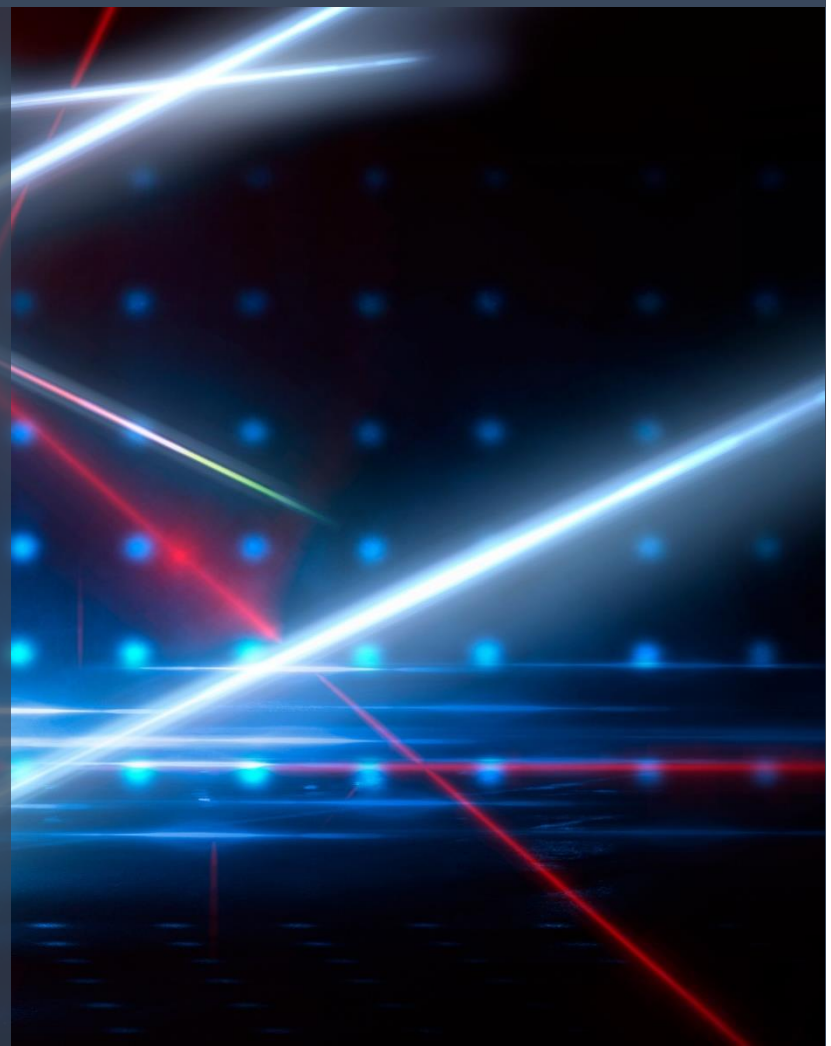
Adapt your fuzzer to the operating conditions

Study the protocol to see where you can speed up fuzzing

Check if stateful checks are interesting to your client

Avoid breaking connection when fuzzing

Parsing of answers should be configurable



:: Takeaways

Create programs to test your fuzzer and then use them as emulators

Plan how to deal with fault management systems beforehand

Log everything

Include "fudge factor" in schedule to account for possible issues

Report your findings in a comprehensive way, find PoC where possible

:: Positive Technologies

Thanks

for attention