

如何(不)讓你發射飛彈

Disrupting factories, missile bases and warships –
Exploration into DDS protocol implementations

Ta-Lun Yen
Federico Maggi
Erik Boasson
C. Toyama, P. Kuo, and M. Cheng
Víctor Mayoral Vilches

TXOne Networks
Trend Micro Research
ADLINK
TXOne Networks
Alias Robotics

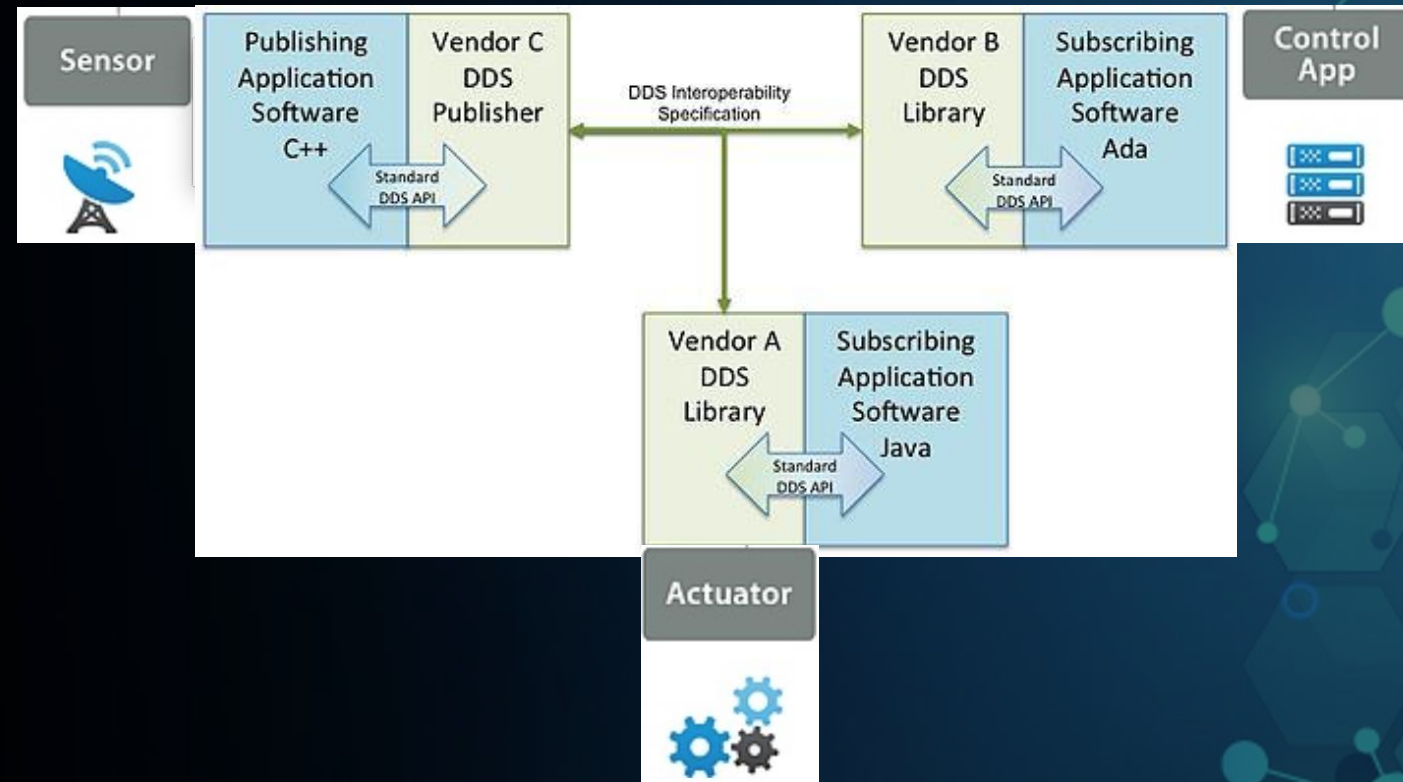
Ta-Lun Yen

- 漏洞研究員 - 睿控網安 (TXOne Networks)
- 逆向工程 / 模糊測試 / 嵌入式裝置 / 協定分析 / 無線電
- 於國內外各研討會發表過
 - Black Hat EU, CODE BLUE, HITCON, hardwear.io...

What is DDS

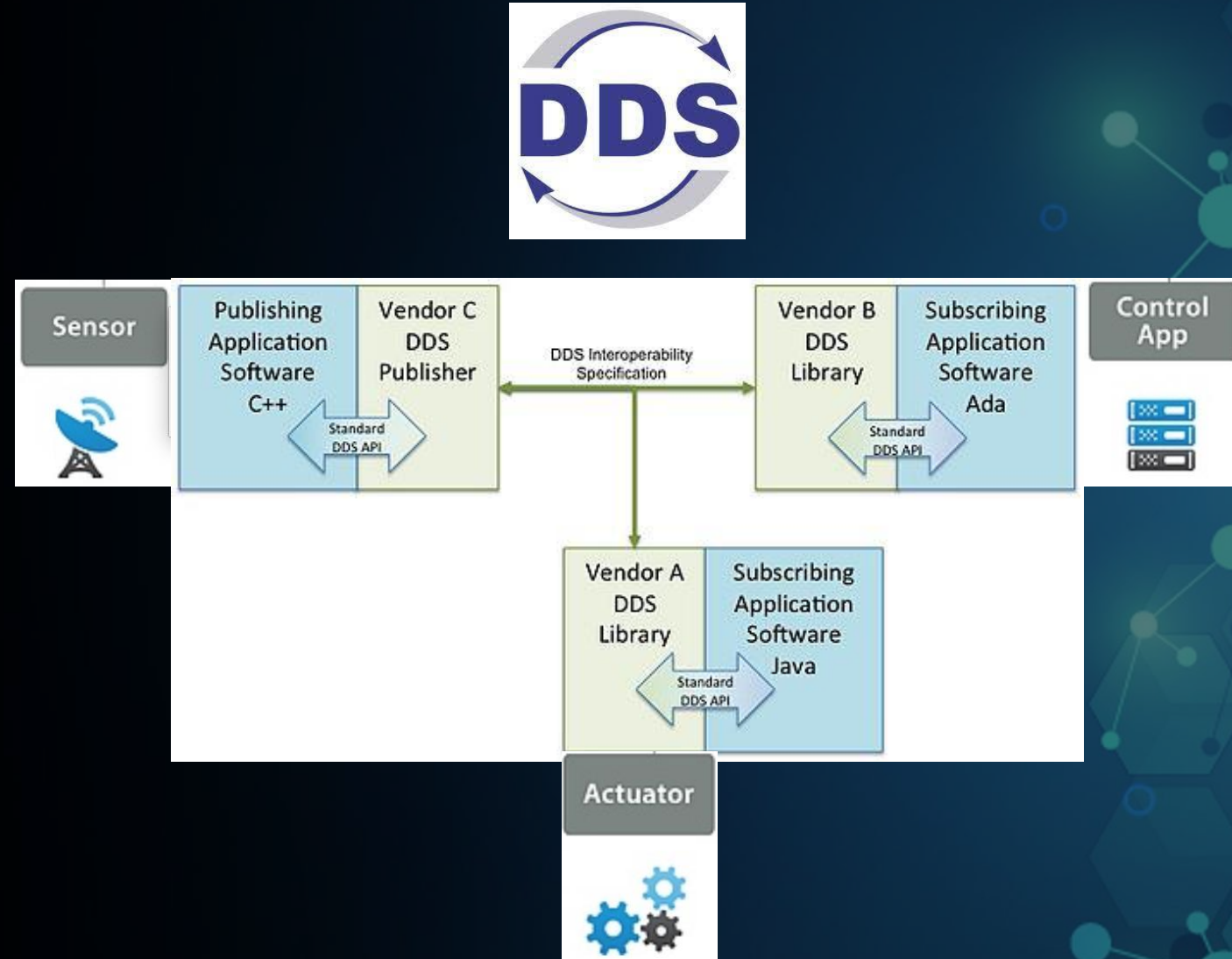
What is DDS

- 從前，地是空虛混沌，淵面黑暗
- 世上有許多的：
 - 感應器 (sensor)
 - 致動器 (actuator)
 - 控制它們的系統



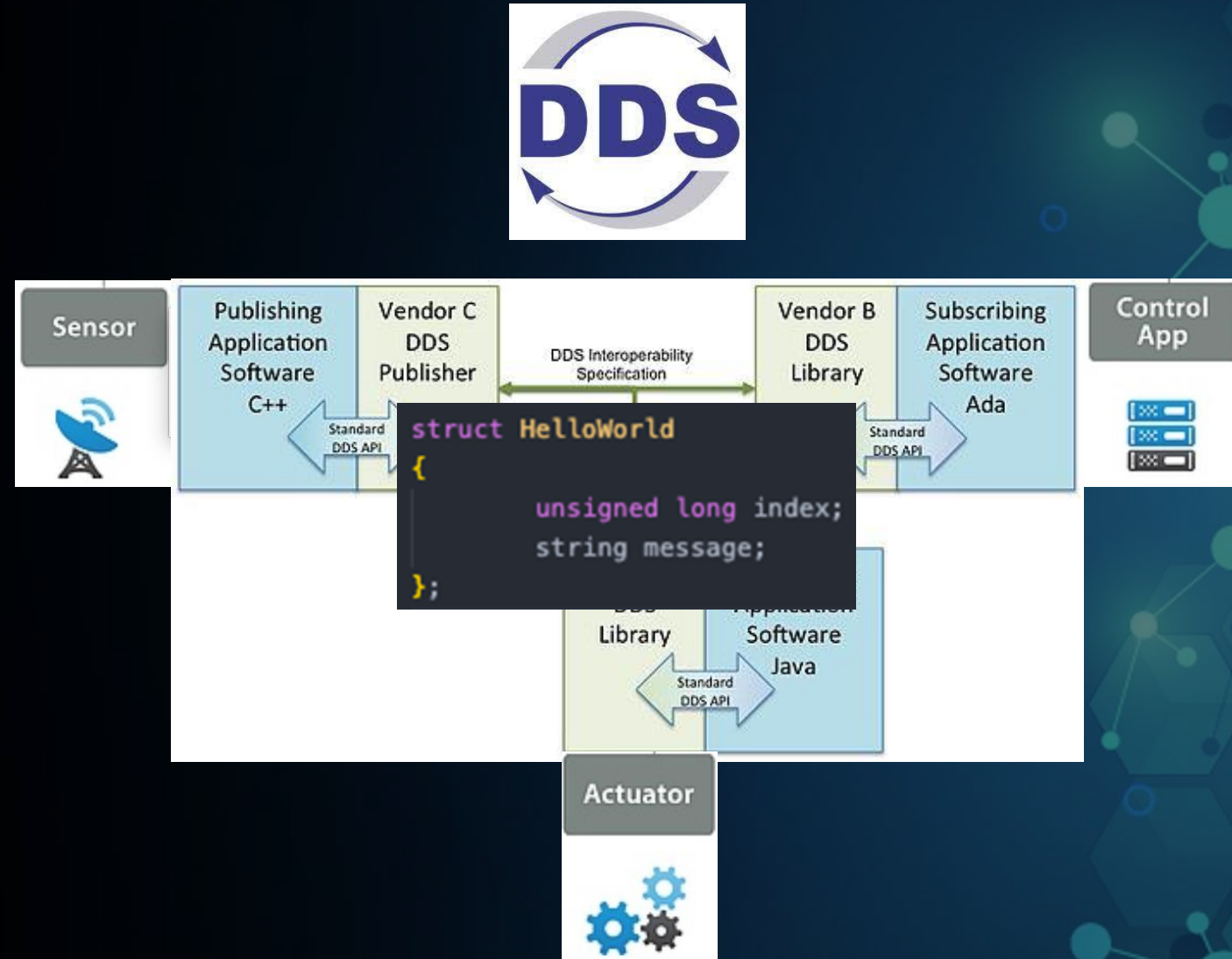
What is DDS

- 從前，地是空虛混沌，淵面黑暗
- 於是有了 DDS，且互通性良好



What is DDS

- 從前，地是空虛混沌，淵面黑暗
- 於是有了 DDS，且互通性良好
- 溝通需要訂定 Schema
 - IDL, 像 Protobuf
 - 協定在 1984 年起草



What is DDS

- 從前，地是空虛混沌，淵面黑暗
- 於是有了 DDS，且互通性良好
- 溝通需要訂定 Schema
 - IDL, 像 Protobuf
 - 但協定在 1984 年起草
 - 支援各式資料型別

```
enum MyEnum
{
    A,
    B,
    C
};

enum MyBadEnum
{
    A1,
    B1,
    C1
};

struct MyEnumStruct
{
    MyEnum my_enum;
};

struct MyBadEnumStruct
{
    MyBadEnum my_enum;
};

struct StringStruct
{
    string my_string;
};

struct LargeStringStruct
{
    string<41925> my_large_string;
};

struct WStringStruct
{
    wstring my_wstring;
};

struct LargeWStringStruct
{
    wstring<41925> my_large_wstring;
};

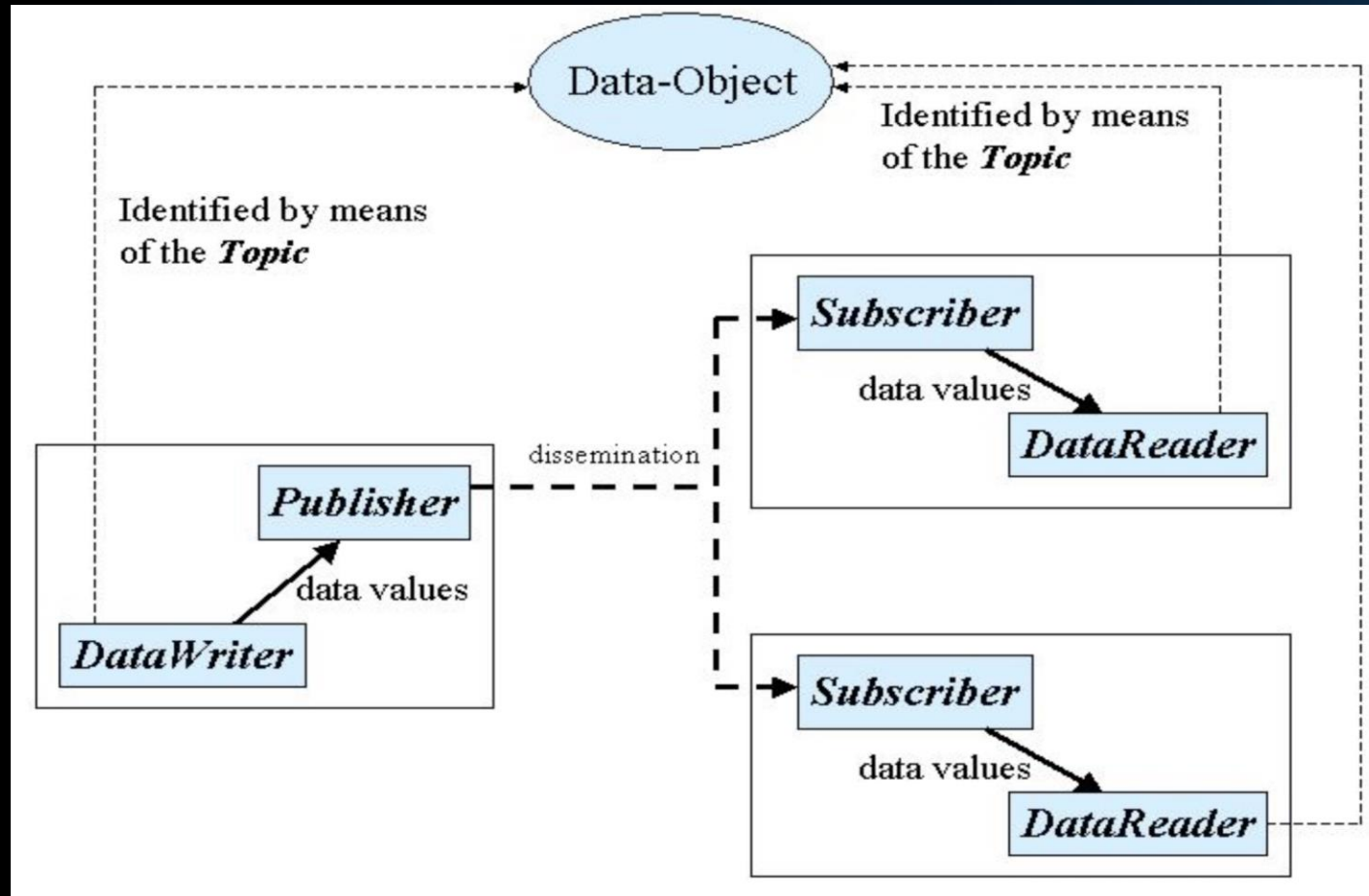
struct NewAliases
{
    int8 int8_;
    uint8 uint8_;
    int16 int16_;
    uint16 uint16_;
    int32 int32_;
    uint32 uint32_;
    int64 int64_;
    @default(555) uint64 uint64_;
    serialized string local_string;
};

switch (wchar)
{
    long case_zero;
    long case_one;
};

using
ication
tware
ava
```

What is DDS

- 它像 MQTT，但看起來又不像



從標準到實作

- DDS 是 OMG 組織維護的一套標準
- 廠家們亦多少參照標準推出產品



非常開源

ADLINK
Leading EDGE COMPUTING

ECLIPSE
FOUNDATION

EPROSIMA
The Middleware Experts

OCI | WE ARE SOFTWARE ENGINEERS.

或許開源

TWINOAKS
COMPUTING INC.
PRACTICAL MIDDLEWARE EXPERTISE

GURUM
Networks

SPARX
SYSTEMS

沒開源

rti Your systems.
Working as one.

說了這麼多， 哪裡在用它？

運輸行業

- 控制鐵路閉塞
- 航班管理系統
- 機場跑道

NAV CANADA improves air traffic management with RTI platform

8 September 2013 (Last Updated September 8th, 2013 18:30)

NAV CANADA is set to improve its NAVCANtrac air traffic management (ATM) system with Real-Time Innovations (RTI) Connex Data Distribution Service (DDS) middleware.



NAV CANADA is set to improve its NAVCANtrac air traffic management (ATM) system with Real-Time Innovations (RTI) Connex Data Distribution Service (DDS) middleware.

NAV Canada said that it selected the RTI middleware platform after a successful implementation of the evaluation stage in order to replace the first-generation distribution architecture of its NAVCANtrac ATM system.

Top 5 M

Analysis
How 3D v
could hel
challenge

News 4 da
EASA: Bo
unground

PRORAIL

ProRail

Large Scale Rail Network Management System

Coflight

Advanced Flight Data Processors Used in European Air Traffic Management System


NAV CANADA

NAV CANADA uses DDS to run real-time traffic management in the world's second busiest air space

 txOne
networks


航太



 [Topics](#) | [Missions](#) | [Galleries](#) | [NASA TV](#) | [Follow NASA](#) | [Downloads](#) | [About](#) | [NASA Audiences](#)

Orion Spacecraft

[Orion Spacecraft](#) | [Overview](#) | [Images](#) | [Videos](#) | [Media Resources](#)



自駕車、自動控制等

PRODUCT BRIEF

QNX Platform for ADAS 2.0



NEWS COMPANIES MARKETPLACE TECHNOLOGY GOT QUESTIONS? EVENTS COM

Industry news & trends

Adlink, Foxconn team up for automated driving system

by Moderation Team · 5 months ago

598 Views

17
SHARES

Share

Tweet



醫療產業

GE Healthcare

[Log in](#) or [register](#) to post comments [Last post](#)

Sat, 05/08/2021 - 02:05

anotherChris
Offline
Last seen: 1 month 1 week ago
Joined: 05/08/2021
Posts: 2

What means this error message: ModuleID=6 Errcode=59

Hello,

This is with **DDS Micro 2.4.11**

I have this following error message:

```
[15.700000000]ERROR: ModuleID=6 Errcode=59 X=1 E=0 T=1 undefined/RTPSInterface.c:6927/RTPS_Interface_receive
```

I am not able to interpret what it means, can anybody explain the root cause and how to solve it ?

Thank you,

Chris

Organization:
GE Healthcare

軍火

- 飛彈系統、C4ISR...
- 戰鬥系統
- 資料傳輸



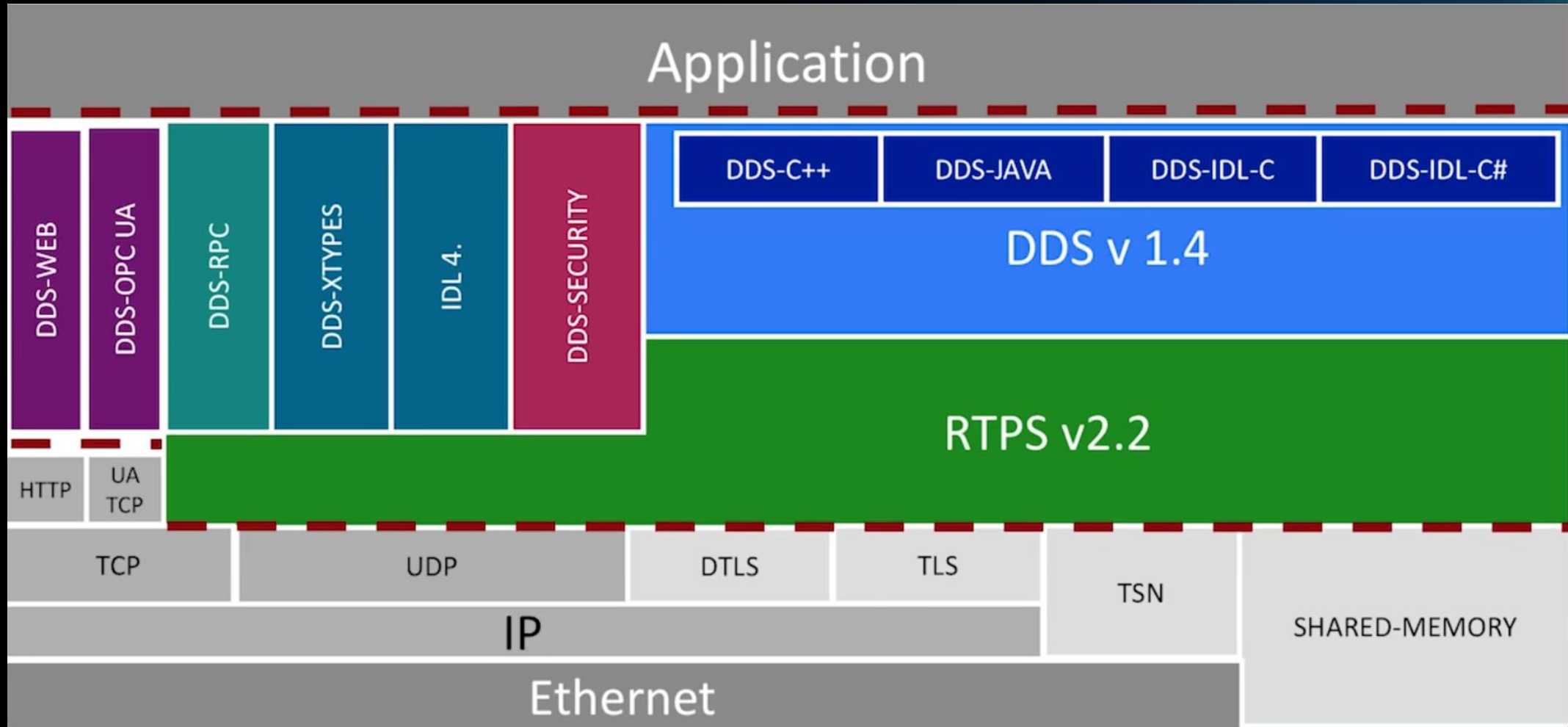
研究目標

研究目標

- 若 DDS 的使用這麼廣泛，我們是否可以找到問題？
- 我們針對前六使用率的實作做「測試」

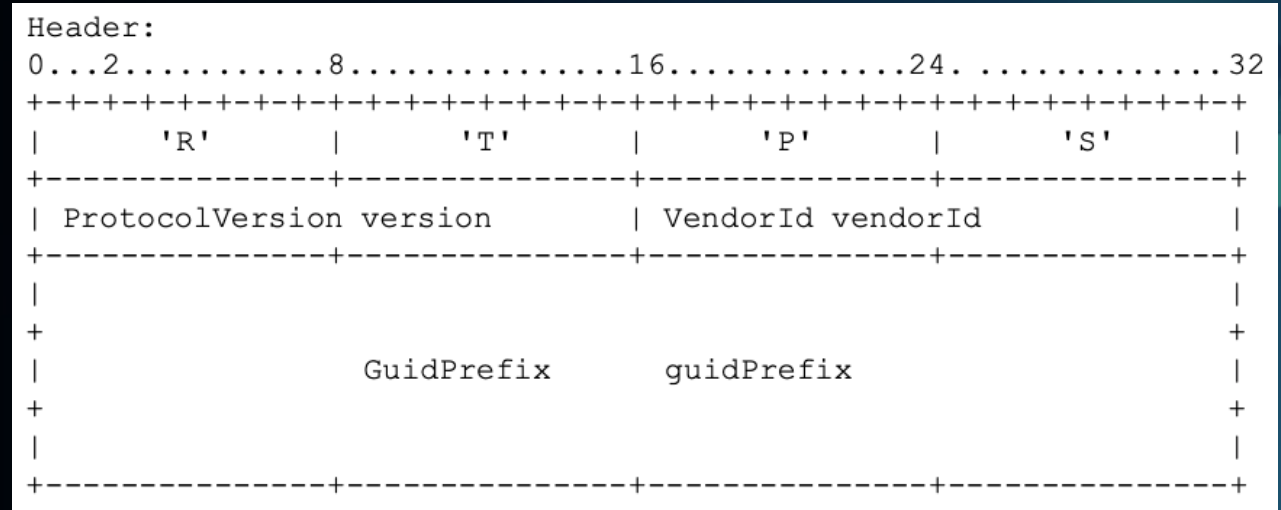


DDSI-RTPS



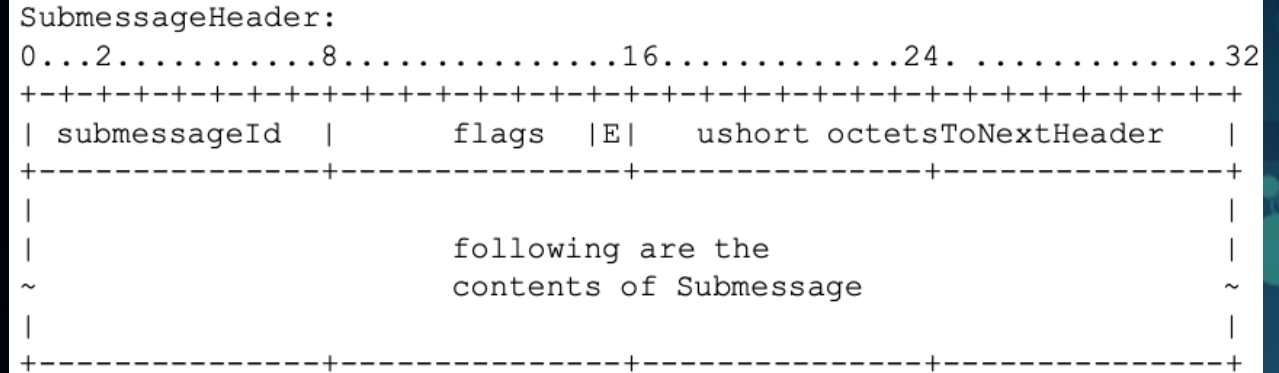
RTPS

- Layer-7
- 固定的 Preamble
- 每個端點皆有 GUID
- 以 Topic 作為不同「頻道」的分界
 - 頻道中，可以有多個 Participant (端點)



```
Real-Time Publish-Subscribe Wire Protocol
  Magic: RTPS
  Protocol version: 2.2
    major: 2
    minor: 2
  vendorId: 01.15 (eProsima - Fast-RTPS)
  guidPrefix: 010f7f014c0d00000a000000
    hostId: 0x010f7f01
    appId: 0x4c0d0000
    instanceId: 0x0a000000
  Default port mapping: domainId=2, participantIdx=8, nature=UNICAST_METATRAFFIC
    [domain_id: 2]
    [participant_idx: 8]
    [traffic_nature: UNICAST_METATRAFFIC (0)]
```

RTPS



```
enum SubmessageKind {
    @value(0x00) RTPS_HE,           /* HeaderExtension */
    @value(0x01) PAD,              /* Pad */
    @value(0x06) ACKNACK           /* AckNack */
    @value(0x07) HEARTBEAT        /* Heartbeat */
    @value(8x08) GAP              /* Gap */
    @value(0x09) INFO_TS          /* InfoTimestamp */
    @value(0x0c) INFO_SRC         /* InfoSource */
    @value(0x0d) INFO_REPLY_IP4   /* InfoReplyIp4 */
    @value(0x0e) INFO_DST         /* InfoDestination */
    @value(0x0f) INFO_REPLY       /* InfoReply */
    @value(0x12) NACK_FRAG        /* NackFrag */
    @value(0x13) HEARTBEAT_FRAG   /* HeartbeatFrag */
    @value(0x15) DATA            /* Data */
    @value(0x16) DATA_FRAG       /* DataFrag */
};
```

沒有現成的 RTPS 解析器...

- Wireshark 有，但並不能用在 Scapy 上
- 想要自動做網路層模糊測試
- 自己寫一個

```
###[ PID_DEFAULT_UNICAST_LOCATOR ]###
parameterId= 0x31
parameterLength= 24
\locator \
###[ RTPS Locator ]###
locatorKind= 0x1000000
port = 60349
address = 172.17.0.2
###[ PID_DEFAULT_MULTICAST_LOCATOR ]###
parameterId= 0x48
parameterLength= 24
\locator \
###[ RTPS Locator ]###
locatorKind= 0x1000000
port = 7401
address = 239.255.0.1
```

```
###[ RTPS Header ]###
magic = 'RTPS'
\protocolVersion\
###[ RTPS Protocol Version ]###
major = 2
minor = 1
\vendorId \
###[ RTPS Vendor ID ]###
vendor_id = b'\x01\x10'
\guidPrefix\
###[ RTPS GUID Prefix ]###
hostId = 0x57631001
appId = 0xd6ab407f
instanceId= 0x5bd9bb1c
###[ RTPS Message ]###
\submessages\
###[ RTPS INFO_DTS (0x0e) ]###
submessageId= 0xe
submessageFlags= 0x1
octetsToNextHeader= 12
\guidPrefix\
###[ RTPS GUID Prefix ]###
hostId = 0x882a1001
appId = 0x5d8c9740
instanceId= 0x78b62dc2
###[ RTPS INFO_TS (0x09) ]###
submessageId= 0x9
submessageFlags= E
octetsToNextHeader= 8
ts_seconds= 1619087604
ts_fraction= 475848017
```

“For every lock, there is someone out there trying to pick it or break in.”

-- David Bernstein

模糊測試 (fuzzing)

- 我們要測試六個實作
- 自己看自己半年前寫的 code 都有困難
 - 看別人的更難，但做模糊測試還是得對目標有了解
- Custom network-based fuzzer + aflpp + (unicorn+aflpp)
- But...

最強的模糊測試



最強的模糊測試

- 看到 length, port, IP 就照直覺亂改
- 結果：



1. EXECUTIVE SUMMARY

- **CVSS v3 8.6**
- **ATTENTION:** Exploitable remotely/low attack complexity
- **Vendors:** Eclipse, eProsima, GurusNetworks, Object Computing, Inc. (OCI), Real-Time Innovations (RTI), TwinOaks Computing
- **Equipment:** CycloneDDS, FastDDS, GurusDDS, OpenDDS, Connex DDS Professional, Connex DDS Secure, Connex DDS Micro, CoreDX DDS
- **Vulnerabilities:** Write-what-where Condition, Improper Handling of Syntactically Invalid Structure, Network Amplification, Incorrect Calculation of Buffer Size, Heap-based Buffer Overflow, Improper Handling of Length Parameter Inconsistency, Amplification, Stack-based Buffer Overflow

PID_METATRAFFIC_UNICAST_LOCATOR

- 「和 Participant 說，你目前有哪些人可以連上去」
- 無須認證

attribute	type	meaning
defaultUnicastLocator List	Locator_t[*]	Default list of unicast locators (transport, address, port combinations) that can be used to send messages to the Endpoints contained in the Participant. These are the unicast locators that will be used in case the Endpoint does not specify its own set of Locators.

172.17.0.4	172.17.0.3	RTPS	350 INFO_TS, DATA(p)
172.17.0.3	8.8.8.8	RTPS	366 INFO_DST, INFO_TS, DATA(p)
172.17.0.3	8.8.8.8	RTPS	174 INFO_DST, HEARTBEAT, HEARTBEAT, HEARTBEAT
172.17.0.3	8.8.8.8	RTPS	162 INFO_DST, ACKNACK, ACKNACK, ACKNACK
172.17.0.3	8.8.8.8	RTPS	174 INFO_DST, HEARTBEAT, HEARTBEAT, HEARTBEAT
172.17.0.3	8.8.8.8	RTPS	174 INFO_DST, HEARTBEAT, HEARTBEAT, HEARTBEAT
172.17.0.3	8.8.8.8	RTPS	174 INFO_DST, HEARTBEAT, HEARTBEAT, HEARTBEAT
172.17.0.3	8.8.8.8	RTPS	174 INFO_DST, HEARTBEAT, HEARTBEAT, HEARTBEAT
172.17.0.1	172.17.255.255		36 57621 → 57621 Len=44
172.17.0.3	8.8.8.8		74 INFO_DST, HEARTBEAT, HEARTBEAT, HEARTBEAT
172.17.0.3	8.8.8.8		74 INFO_DST, HEARTBEAT, HEARTBEAT, HEARTBEAT
172.17.0.3	8.8.8.8		74 INFO_DST, HEARTBEAT, HEARTBEAT, HEARTBEAT
172.17.0.3	8.8.8.8		36 INFO_DST, INFO_TS, DATA(p)
172.17.0.3	8.8.8.8		74 INFO_DST, HEARTBEAT, HEARTBEAT, HEARTBEAT
172.17.0.3	8.8.8.8		32 INFO_DST, ACKNACK, ACKNACK, ACKNACK



流量放大(反射)攻擊

- 從 wireshark 取封包，隨意修改後送出
- 放大倍率約 8~36
- 全實作皆受影響
 - 此問題為協定等級
 - 協定允許寫入任意 IP，無須認證
- Demo: <https://github.com/the-dds/pocs>



有效的模糊測試：擒 bug 先擒序列化

- Preamble 可以用來找到正確的「路」

```
if ((buff_.size() >= 4) && ACE_OS::memcmp(buff_.rd_ptr(), "RTPS", 4) == 0) {
    RTPS::Message message;

    DCPS::Serializer ser(&buff_, encoding_plain_native);
    Header header;
    if (!(ser >> header)) {
        ACE_ERROR((LM_ERROR,
                   ACE_TEXT("(%P|%t) ERROR: Spdp::SpdpTransport::handle_input() - ")
                   ACE_TEXT("failed to deserialize RTPS header for SPDP\n")));
        return 0;
    }
}
```

```
while (buff_.length() > 3) {
    const char subm = buff_.rd_ptr()[0], flags = buff_.rd_ptr()[1];
    ser.swap_bytes((flags & FLAG_E) != ACE_CDR_BYTE_ORDER);
    const size_t start = buff_.length();
    CORBA::UShort submessageLength = 0;
    switch (subm) {
        case DATA: {
```

有效的模糊測試：擒 bug 先擒序列化

- 或者也可以用 gdb...

```
(gdb) bt
#0  OpenDDS::DCPS::Serializer::smemcpy (this=0x7ffc615e18a0, to=0x7ffc614dddc0 "w", from=0x612000000078 "\017", n=2) at DCPS/Serializer.cpp:374
#1  0x00000000004d3d1b in OpenDDS::DCPS::Serializer::doread (this=0x7ffc615e18a0, dest=0x7ffc614dddc0 "w", size=2, swap=<optimized out>, offset=0)
#2  OpenDDS::DCPS::Serializer::buffer_read (this=<optimized out>, dest=<optimized out>, size=<optimized out>, swap=<optimized out>) at /usr/local/
#3  0x00007fdfa4d1439b in OpenDDS::DCPS::operator>> (s=..., x=@0x7ffc614dddc0: 119) at DCPS/Serializer.inl:1173
#4  0x00007fdfa16e0d2f in OpenDDS::DCPS::operator>> (outer_strm=..., uni=...) at RtpsCoreTypeSupportImpl.cpp:11440
#5  0x00007fdfa16dff18 in OpenDDS::DCPS::operator>> (strm=..., seq=...) at RtpsCoreTypeSupportImpl.cpp:8985
#6  0x00000000004d051f in main (argc=<optimized out>, argv=<optimized out>) at test.cpp:106
```

同工，不同師傅

- 不同實作下，剛好都長得一樣！

```
case SMID_DATA:
    state = "parse:data";
    {
        struct nn_rsample_info sampleinfo;
        unsigned char *datap;
        uint32_t datasz = 0;
        size_t submsg_len = submsg_size;
        /* valid_Data does not validate the payload */
        if (!valid_Data (rst, &sm->data, submsg_size, byteswap, &sampleinfo, &datap, &datasz))
            goto malformed;
        /* This only decodes the payload when needed (possibly reducing the submsg size). */
        if (!decode_Data (rst->gv, &sampleinfo, datap, datasz, &submsg_len))
            goto malformed;
        /* Set the sample bswap according to the payload info. */
        if (!set_sampleinfo_bswap(&sampleinfo, (struct CDRHeader *)datap))
            goto malformed;
        sampleinfo.timestamp = timestamp;
        sampleinfo.reception_timestamp = tnowWC;
        handle_Data (rst, tnowE, rmsg, &sm->data, submsg_len, &sampleinfo, datap, &deferred_wakeup, prev_smid);
        rst_live = 1;
        ts_for_latmeas = 0;
    }
    break;
```

同工，不同師傅

- 不同實作下，剛好都長得一樣！

```
nn_rtps_msg_state_t res = decode_rtps_message (ts1, gv, &rmsg, &hdr, &buff, &sz,
if (res != NN_RTPS_MSG_STATE_ERROR)
{
    handle_submsg_sequence (ts1, gv, conn, &srcloc, ddsrt_time_wallclock (), ddsrt.
```

```
state_smkind = sm->smhdr.submessageId;
switch (sm->smhdr.submessageId)
{
    case SMID_PAD:
        GVTRACE ("PAD");
        break;
    case SMID_ACKNACK:
        state = "parse:acknack";
        if (!valid_AckNack (rst, &sm->acknack))
            goto malformed;
        handle_AckNack (rst, tnowE, &sm->acknack);
        ts_for_latmeas = 0;
        break;
    case SMID_HEARTBEAT:
        state = "parse:heartbeat";
        if (!valid_Heartbeat (&sm->heartbeat))
            goto malformed;
        handle_Heartbeat (rst, tnowE, rmsg, &sm->heartbeat);
        ts_for_latmeas = 0;
        break;
    case SMID_GAP:
        state = "parse:gap";
        /* Gap is handled synchronously in
           sometimes have to record a gap :
           first case by definition doesn't
           the second one avoids that because
           rst after inserting the gap in t
        */
        if (!valid_Gap (&sm->gap, submsg_seq))
            goto malformed;
        handle_Gap (rst, tnowE, rmsg, &sm->gap);
        ts_for_latmeas = 0;
        break;
    case SMID_INFO_TS:
```



```
unsigned char *s = __AFL_FUZZ_TESTCASE_BUF;
size_t sz = (size_t) __AFL_FUZZ_TESTCASE_LEN;
while (__AFL_LOOP(10000))
{
    ACE_Message_Block *mb = new ACE_Message_Block (sz);

    // do our own write to mb
    ACE_OS::memcpy(mb->wr_ptr(), s, sz);
    mb->wr_ptr(sz);

    // most code below stolen from Spdp.cpp

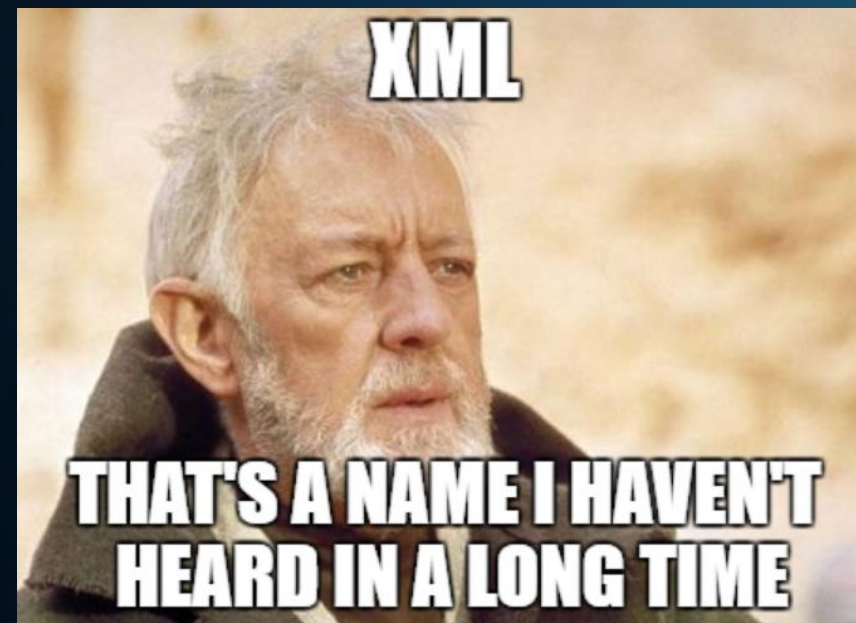
    OpenDDS::DCPS::Serializer ser (mb, encoding_plain_native);
    OpenDDS::RTPS::Header header;
    if (!(ser >> header)) {
        return 0; // this might be mutated by afl
        // ACE_OS::printf("%s\n", "fail deserialize");
    }
}
```

如果目標是 binary ?

- 針對二進制 (binary) 目標做模糊測試
 - aflpp + unicorn engine
 - 能夠對 Binary 任一地方、任一狀態做模糊測試

XML 和外部相依套件

- 最好：用別人的、實戰認證過、有測試過的、有更新
- 還行：自己重寫一遍，請很多人來測試
- 頗糟：用別人的，但用十年前的版本



@xml

```
<domain_participant name="MyPubParticipant"  
domain_ref="aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

@xml

```
<domain_participant name="CircleSubParticipant"  
domain_ref="MyDomaiaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

'aaaaaaaaabbbbbbbbbcddddddddd' part could overwrite memory address

```
Program received signal SIGSEGV, Segmentation fault.  
0x00007ffff60991aa in RTIXMLObject_lookupRef () from /home/trend/rticonnextdds-connector-py/rticonnextdds_conne  
(gdb) x/s *((char **)environ)  
0x6161616161616161: <error: Cannot access memory at address 0x6161616161616161>  
(gdb) x/s *((char **)environ+1)  
0x6262626262626262: <error: Cannot access memory at address 0x6262626262626262>  
(gdb) x/s *((char **)environ+2)  
0x6363636363636363: <error: Cannot access memory at address 0x6363636363636363>  
(gdb) x/s *((char **)environ+3)  
0x6464646464646464: <error: Cannot access memory at address 0x6464646464646464>  
(gdb) x/s *((char **)environ+4)  
0x7ffffffffffe00: ""  
(gdb) x/s *((char **)environ+5)  
0x7ffffffffffe0d: "LC_MONETARY=lzh_TW"  
(gdb) x/s *((char **)environ+6)  
0x7ffffffffffe0f: "XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/trend"  
(gdb) x/s *((char **)environ+7)  
0x7ffffffffffe121: "SESSION=ubuntu"  
(gdb) x/s *((char **)environ+8)  
0x7ffffffffffe130: "GPG_AGENT_INFO=/home/trend/.gnupg/S.gpg-agent:0:1"  
(gdb) x/s *((char **)environ+9)  
0x7ffffffffffe162: "SHELL=/bin/bash"  
(gdb) x/s *((char **)environ+10)  
0x7ffffffffffe172: "XDG_MENU_PREFIX=gnome-"  
(gdb) █
```

'bbbbbb' part could overwrite RIP

```
Program received signal SIGSEGV, Segmentation fault.  
0x0000062626262626 in ?? ()  
(gdb) bt  
#0 0x0000062626262626 in ?? ()  
#1 0x0000000000000000 in ?? ()  
(gdb) i r  
rax          0x0          0  
rbx          0x6161616161616161      7016996765293437281  
rcx          0x0          0  
rdx          0x61        97  
rsi          0x7fffffff9e10      140737488330256  
rdi          0x0          0  
rbp          0x6161616161616161      0x6161616161616161  
rsp          0x7fffffffaf190     0x7fffffffaf190  
r8           0x0          0  
r9           0x0          0  
r10          0x0          0  
r11          0x0          0  
r12          0x6161616161616161      7016996765293437281  
r13          0x6161616161616161      7016996765293437281  
r14          0x6161616161616161      7016996765293437281  
r15          0x6161616161616161      7016996765293437281  
rip          0x6262626262626262     0x6262626262626262  
eflags      0x10202     [ IF RF ]  
cs           0x33        51  
ss           0x2b        43  
ds           0x0          0  
es           0x0          0  
fs           0x0          0  
gs           0x0          0  
(gdb) █
```

漏洞一覽

	CVE	CWE	Notes	Status
All	-	CWE-406	Network reflection	
OMG (specs)	-	Extend specs to allow white-listing		Will fix in DDSI-RTPS 2.5
RTI ConnexDDS	CVE-2021-38487	Patched in $\geq 6.1.0$		Mitigated with patch
OCI OpenDDS	CVE-2021-38429	Patched in $\geq 3.18.1$		Mitigated with patch
ADLINK CycloneDDS	-	Had already an exp. back-off mechanism		Already mitigated
GurumDDS	-	Had already an exp. back-off mechanism		Already mitigated (No reply, 5 times)
eProsima Fast-DDS	CVE-2021-38425	Patched in master branch		Mitigated with patch
Twin Oaks CoreDX		Patched in $> 5.9.1$		Mitigated with patch

漏洞一覽

OCI OpenDDS	CVE-2021-38445	CWE-130	Failed assertion check
OCI OpenDDS	CVE-2021-38447	CWE-405	Resource exhaustion (slowloris)
RTI ConnexDDDS	CVE-2021-38435	CWE-131	Segmentation fault via network
GurumDDS	CVE-2021-38423	CWE-131	Segmentation fault via network
GurumDDS	CVE-2021-38439	CWE-122	Heap-overflow via network
GurumDDS	CVE-2021-38437	CWE-1104	Unmaintained, vulnerable XML lib.
CycloneDDS	CVE-2021-38441	CWE-123	Heap-write primitive in XML parser
CycloneDDS	CVE-2021-38443	CWE-228	8-bytes heap-write primitive
RTI ConnexDDDS	CVE-2021-38427	CWE-121	Stack-based overflow in XML parser
RTI ConnexDDDS	CVE-2021-38433	CWE-121	Stack-based overflow in XML parser

漏洞總結與分析

- Resource exhaustion in OpenDDS

```
c0: 5500 0000 4055 5503 e855 0040 5555 3755 U...@UU..U.@UU7U
d0: 5501 ffff ffff e81d 2948 0703 cc00 0003 U.....)H.....
e0: e055 5562 5555 ffff ffe8 1d29 4807 03cc .UUbUU.....)H...
f0: 5555 0101 0101 0101 0101 0101 0101 1501 UU.....
00: 0101 0101 0101 0101 0101 0101 0101 0101 .....
10: 0101 0101 0101 0101 0e01 0101 0101 0101 .....
20: 0101 0101 0101 0101 0101 0101 0101 0101 .....
30: 0101 0101 0101 0101 0101 0101 0101 0101 .....
```

```
at /usr/local/src/opensdds/ACE_wrappers/TA0/tao/Generic_Sequence_T.h:239
#5 0x00007fc3d7dee3e7 in TA0::unbounded_value_sequence<int>::length (this=0x7fff13410178, length=143164
8085) at /usr/local/src/opensdds/ACE_wrappers/TA0/tao/Unbounded_Value_Sequence_T.h:62
#6 0x00007fc3d64261c0 in OpenDDS::DCPS::operator>> (strm=..., seq=...) at RtpsCoreTypeSupportImpl.cpp:1
977
#7 0x00007fc3d643630a in OpenDDS::DCPS::operator>> (strm=..., stru=...) at RtpsCoreTypeSupportImpl.cpp:
2335
#8 0x00007fc3d64f7750 in OpenDDS::DCPS::operator>> (outer_strm=..., uni=...) at RtpsCoreTypeSupportImpl
.cpp:11836
#9 0x00007fc3d64e2f18 in OpenDDS::DCPS::operator>> (strm=..., seq=...) at RtpsCoreTypeSupportImpl.cpp:8
985
#10 0x00007fc3d66129bb in OpenDDS::DCPS::operator>> (strm=..., stru=...) at RtpsCoreTypeSupportImpl.cpp:
18452
#11 0x00000000004ce1c7 in main (argc=<optimized out>, argv=<optimized out>) at test.cpp:79
```

漏洞總結與分析

- Buffer overflow in XML parser in RTI Connexx DDS

```
loc_95A0FF:  
mov     r15, [rsp+48h+var_40]  
lea     rsi, [rsp+48h+var_40]  
mov     rdi, [rbx]  
call   __INTERNAL_trim_to_complete_utf8_characters  
mov     r14, [rsp+48h+var_40]  
mov     rsi, [rbx] ; src  
mov     r12, r14  
sub     r12, rsi  
mov     rdi, [rbp+0] ; dest  
mov     rdx, r12 ; n  
call   _memcpy ; bof here  
add     [rbx], r12  
add     [rbp+0], r12  
mov     eax, 2  
test   r13b, r13b  
jnz    short loc_95A145
```

@xml

```
<domain_participant name="CircleSubParticipant"  
domain_ref="MyDomaiaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
from /src/rticonnext/rticonnextdds-connector-py/rt  
b/x64Linux2.6gcc4.4.5/librtiddsconnector.so  
#1 0x6262626262626263 in ?? ()  
#2 0x0000626262626262 in ?? ()  
#3 0x0000000001521cc0 in ?? ()  
#4 0x0000000001525440 in ?? ()  
#5 0x000000000138d650 in ?? ()  
#6 0x00000000012ae210 in ?? ()  
#7 0x00007fb25c63f12b in RTI_utf8_toUtf8 ()  
from /src/rticonnext/rticonnextdds-connector-py/rt
```


漏洞總結與分析

- 族系不及備載
- 六大實作，無一倖免
- ICSA-21-315-02 Multiple Data Distribution Service (DDS) Implementations
- 至今尚未完全修補 (流量放大、某些廠商不讀不回)

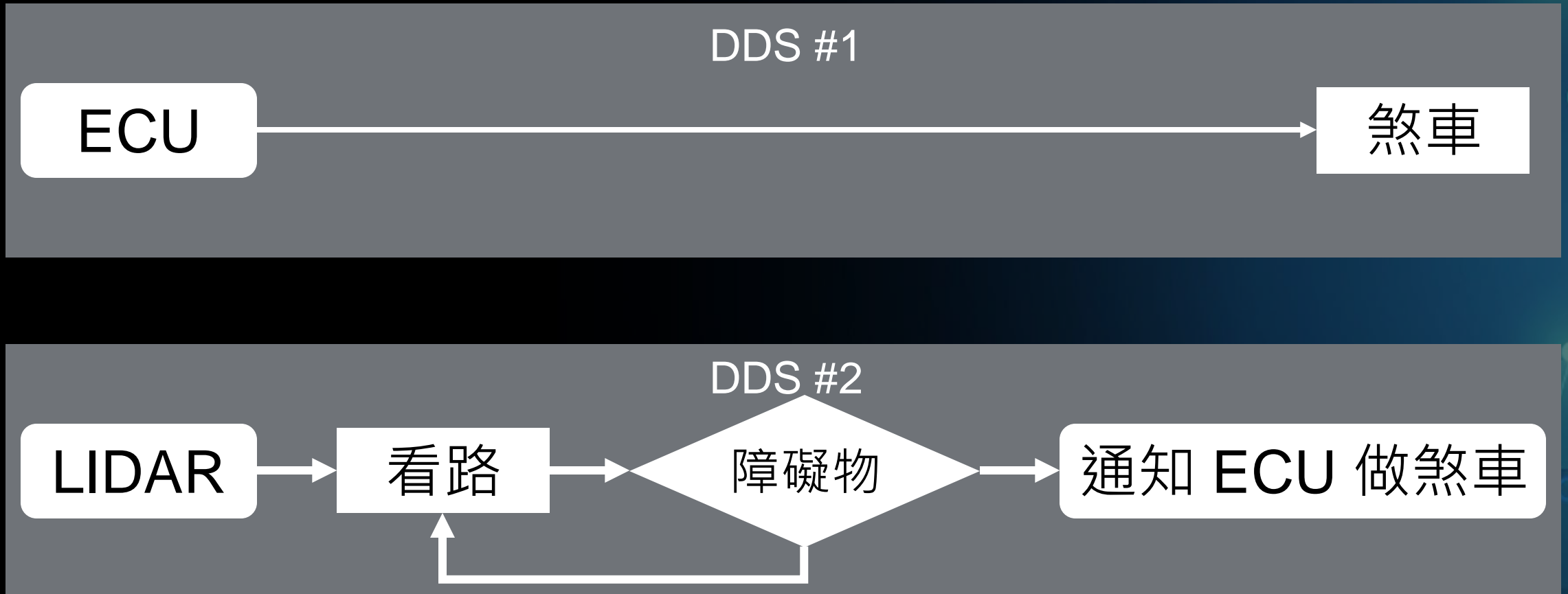
漏洞總結與分析

- 現在是 {year} 年，但還是有不少：
 - Buffer Overflow (heap/stack)
 - Failed assertion
 - Outdated dependencies
- 我們弄壞了 OMG 的規範：
 - 在未來的 DDSI-RTPS 中會修正
 - 目前大部分廠家已經先「違反」DDSI-RTPS 來修正此問題

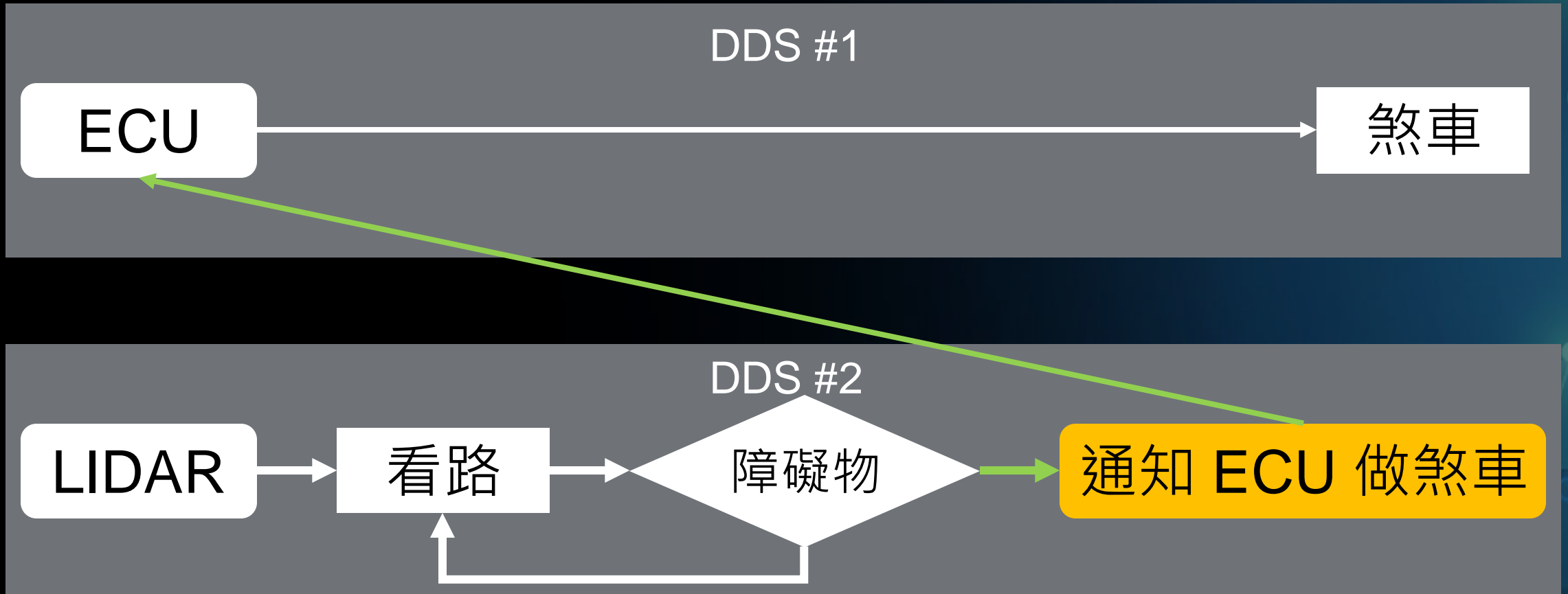
漏洞總結與分析

- 即使不能 exploit，亦能造成使用環境上的衝擊

正常運作情形



攻擊情境 #1 (DoS 導致執行中斷)



攻擊情境 #1 (DoS 導致執行中斷)



DDS #1

~~煞車~~

DDS #2

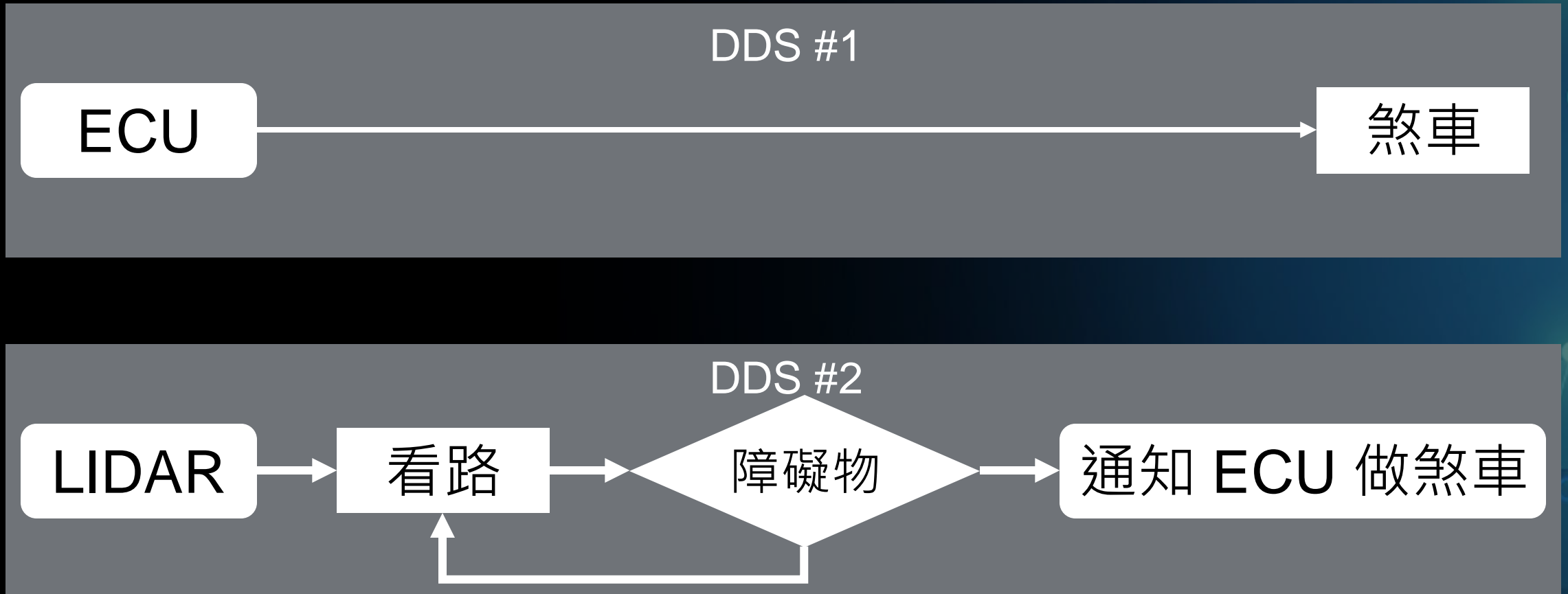
LIDAR

看路

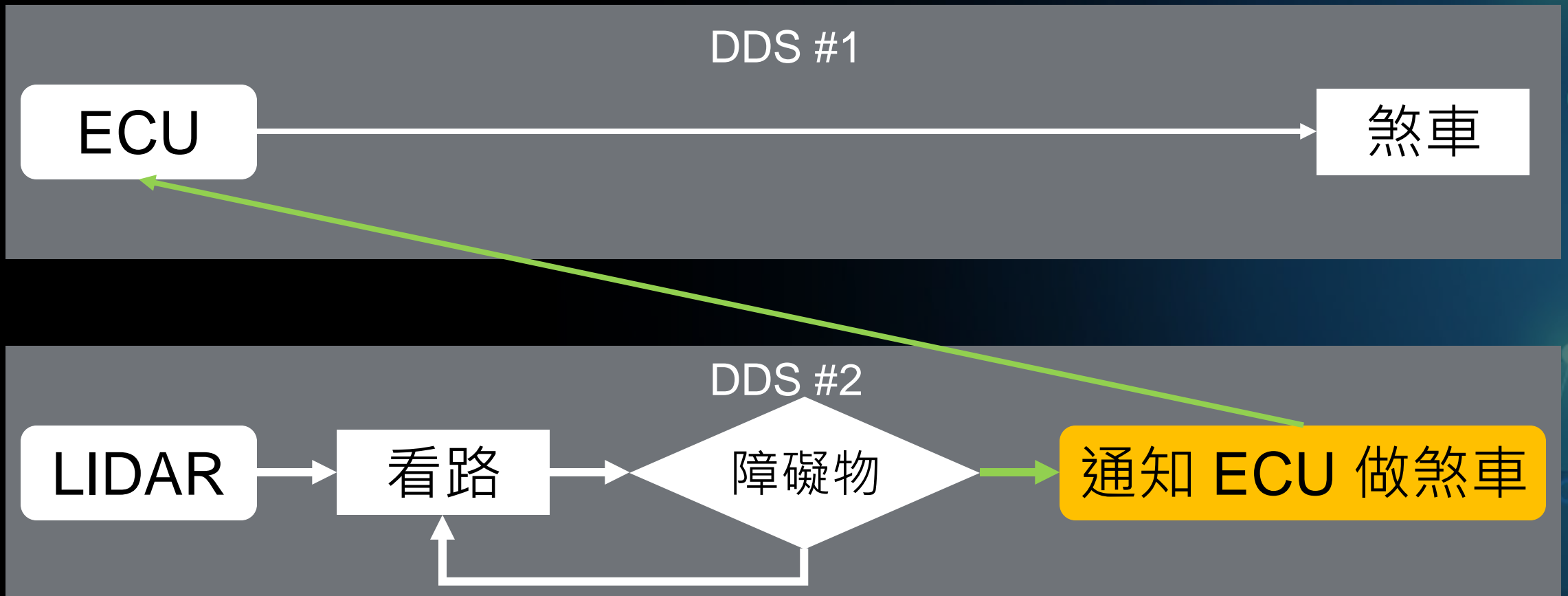
障礙物

通知 ECU 做煞車

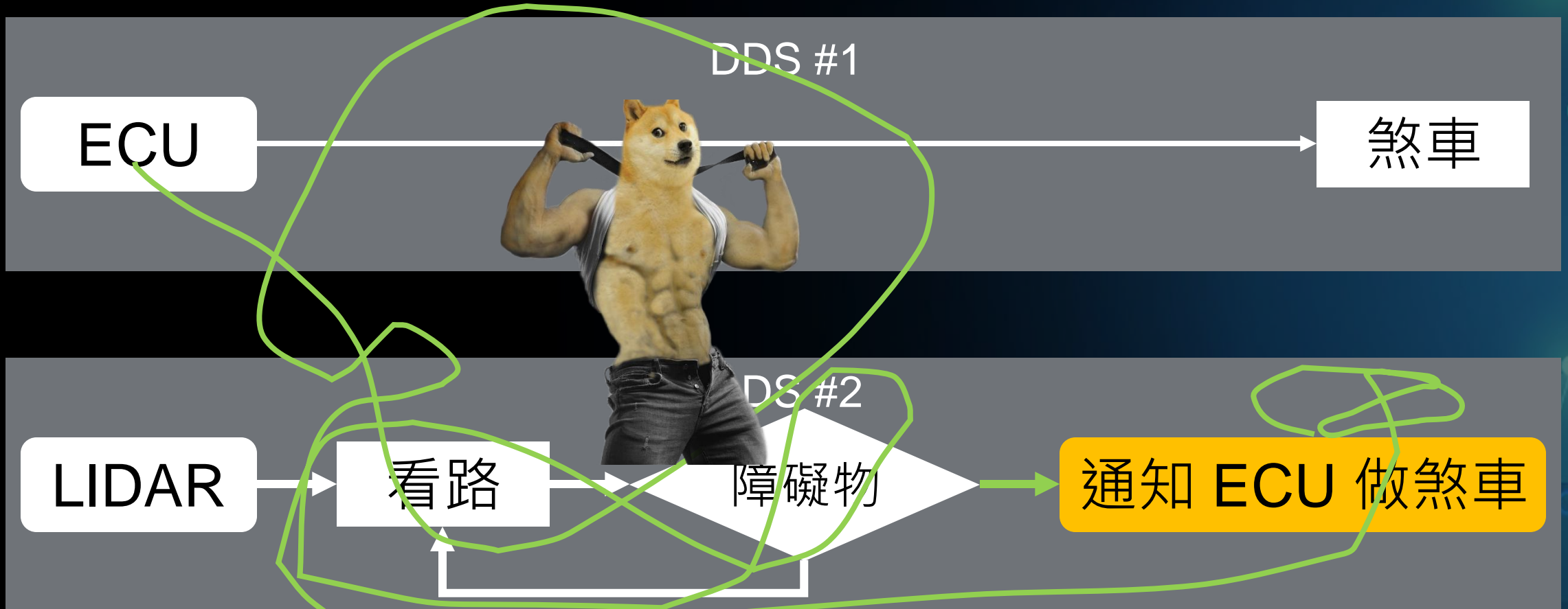
攻擊情境 #2 (流量放大導致傳輸阻塞)

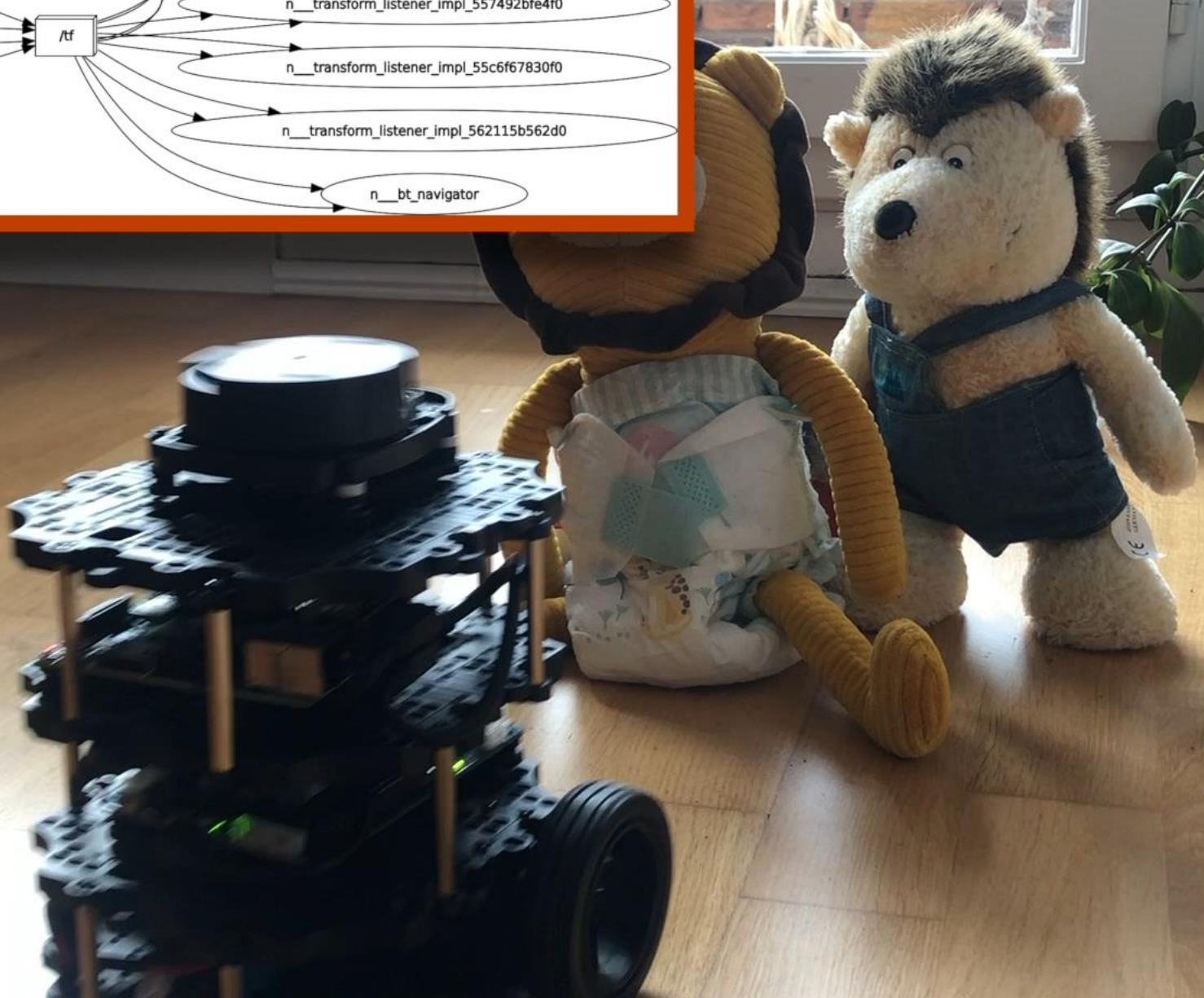
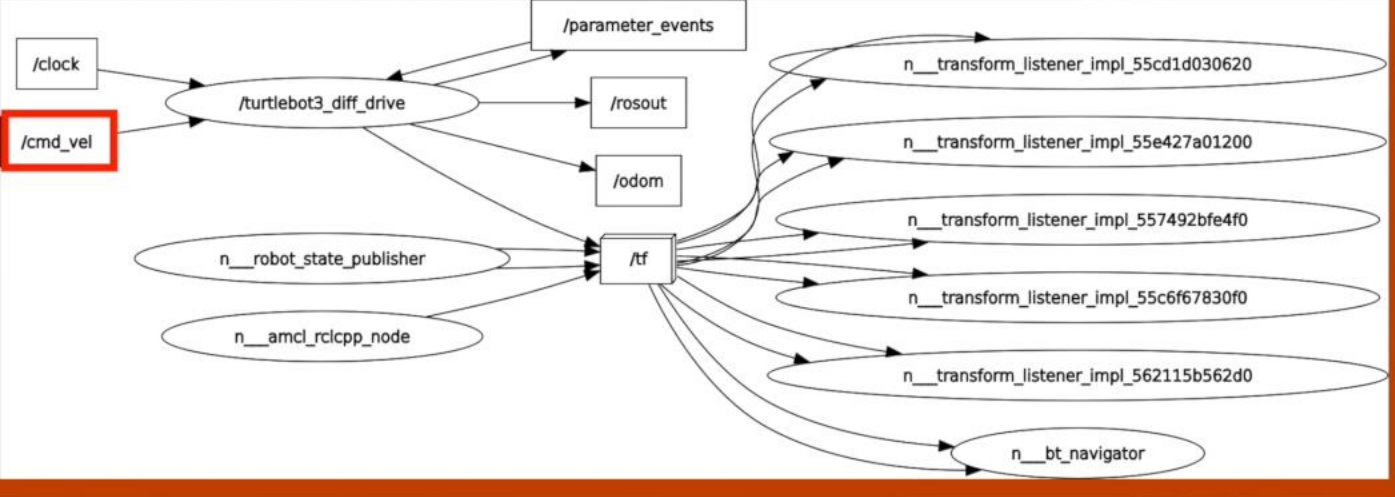


攻擊情境 #2 (流量放大導致傳輸阻塞)



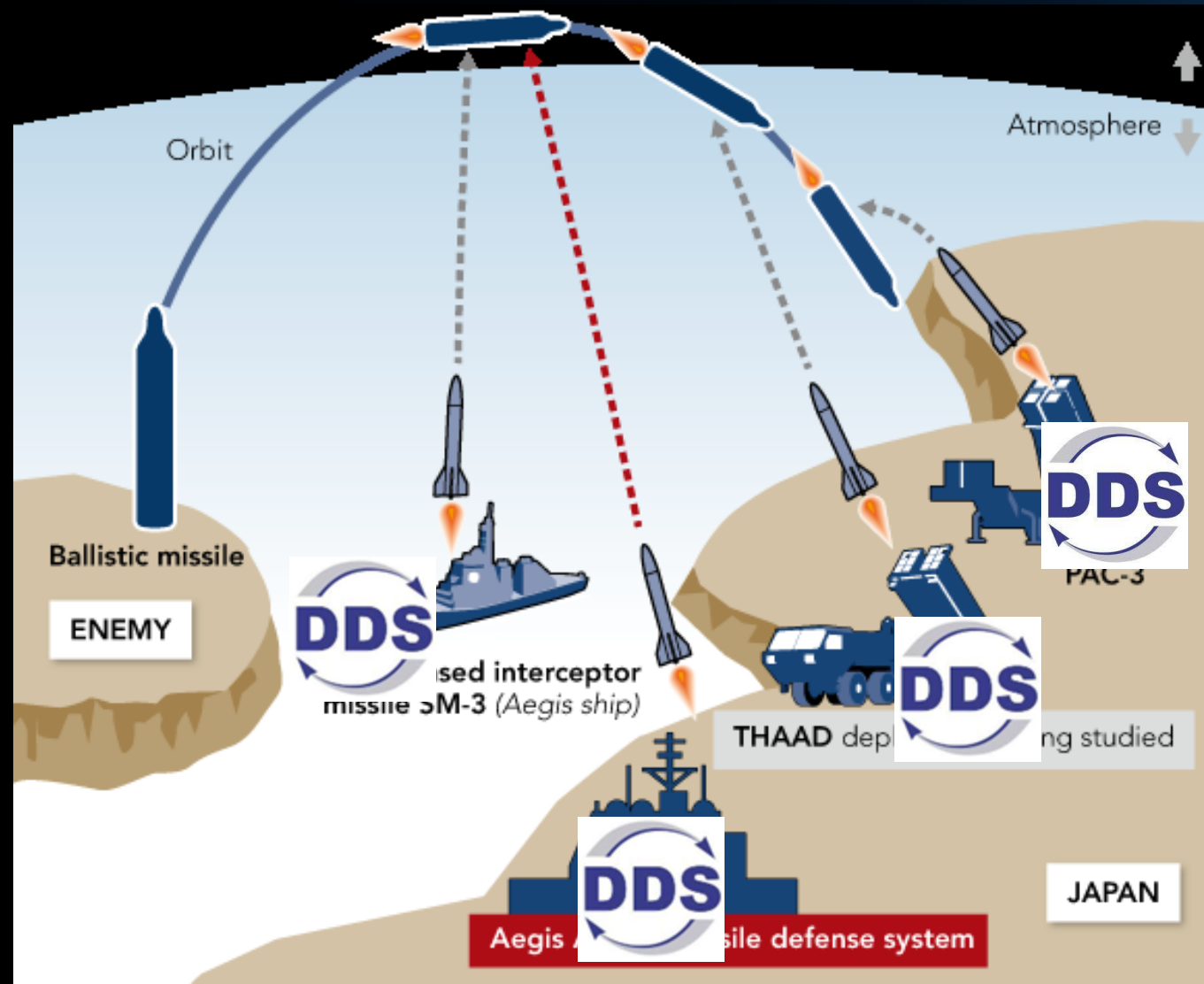
攻擊情境 #2 (流量放大導致傳輸阻塞)



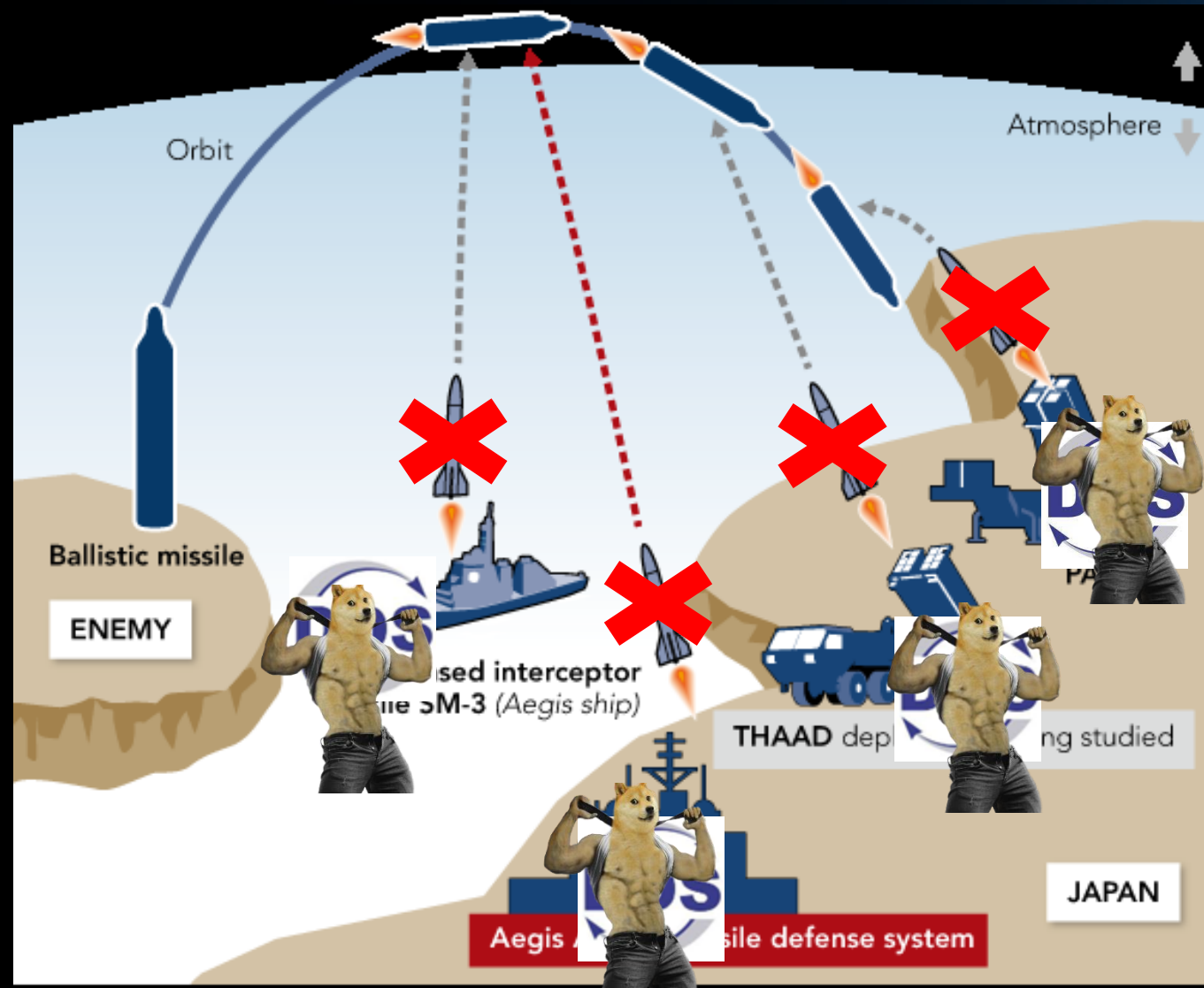




攻擊情境 #3



攻擊情境 #3



威脅模型分析

- DDS 位置應該在 Control Plane 中，但找到不少暴露在外的觀點
- 在任一使用 DDS 的網路中，攻下任一端點即代表全境淪陷
- DDS 的複雜性使其能夠用來做資料竊取、橫向移動、探索網路環境

DDS 於網路上的能見度

安全的透過 Internet 使用 DDS ?

- DDS 是 Layer 7 協定
- 沒有一致對於
透過 Internet 使用 DDS 的好建議
- OCI 有 Example Code , 但 :

```
111 msock.on('message', (msg, rinfo) => {
112     if (rinfo.port === args.upt) {
113         return
114     }
115     if (args.verbose) {
116         console.log(`mc from ${rinfo.address}:${rinfo.port} ${rinfo.size} B`)
117     }
118     for (let sendAddr of Object.keys(sendTo)) {
119         msock.send(msg, sendTo[sendAddr], sendAddr)
120     }
121 })
```

◦ start the multicast repeater

```
--rm --net=host objectcomputing/opendds:repeater /opt/OpenDDS/tools/repeater/repeater.js --gro
```

◦ start the subscriber

```
docker run --name=subscriber --rm --net=host -w /opt/OpenDDS/tests/DCPS/Messenger
```

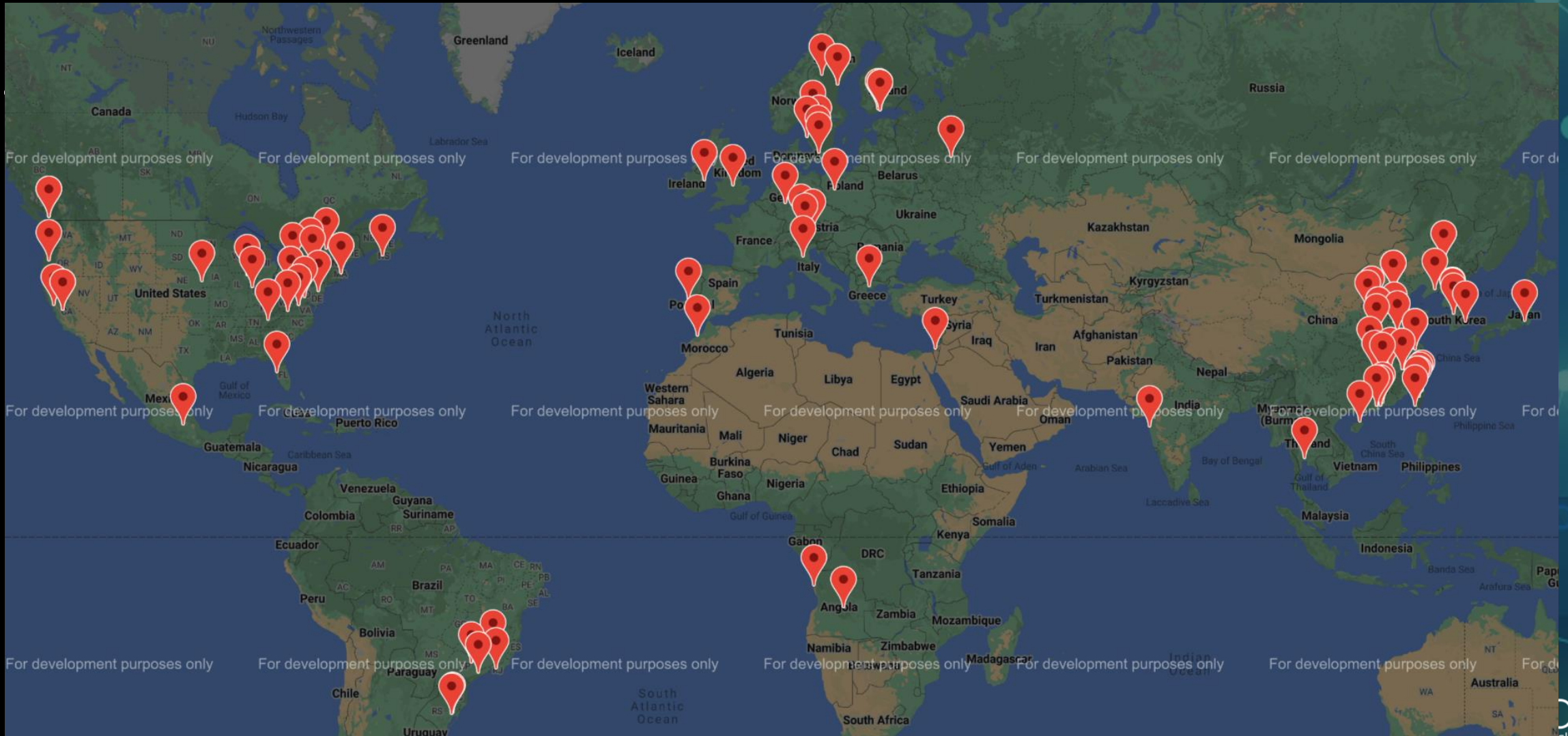
```
x High severity vulnerability found in node
Description: Insufficient Hostname Verification
Info: https://snyk.io/vuln/SNYK-UPSTREAM-NODE-570869
Introduced through: node@10.15.0
From: node@10.15.0
Fixed in: 10.21.0
```

```
x High severity vulnerability found in node
Description: Memory Corruption
Info: https://snyk.io/vuln/SNYK-UPSTREAM-NODE-570870
Introduced through: node@10.15.0
From: node@10.15.0
Fixed in: 10.21.0
```

```
Package manager: deb
Project name: docker-image|objectcomputing/opendds
Docker image: objectcomputing/opendds:repeater
Platform: linux/amd64
```

```
Tested 218 dependencies for known vulnerabilities, found 427 vulnerabilities.
```


The DDS Internet Scanner Project

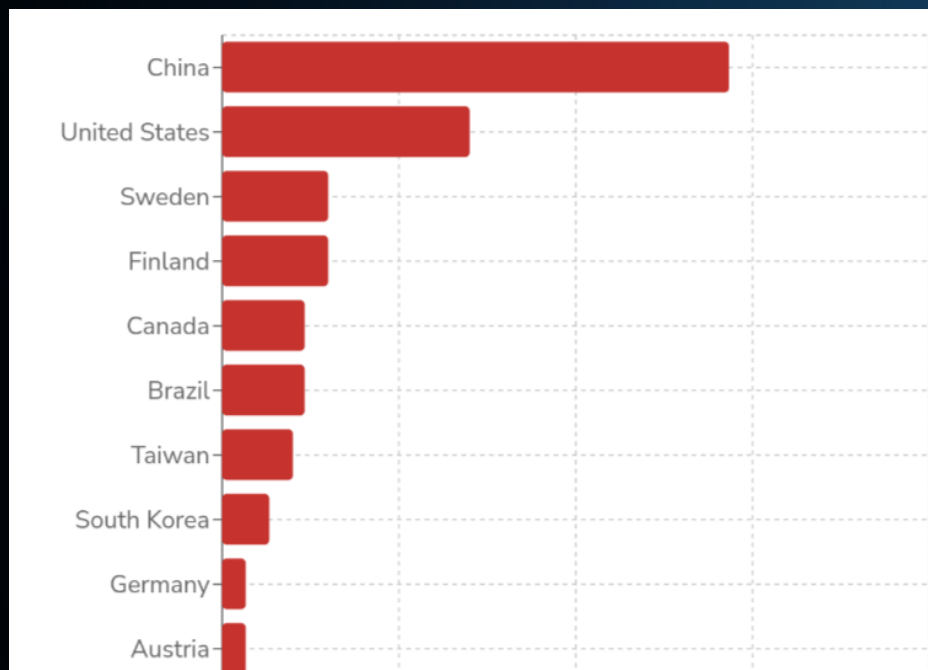
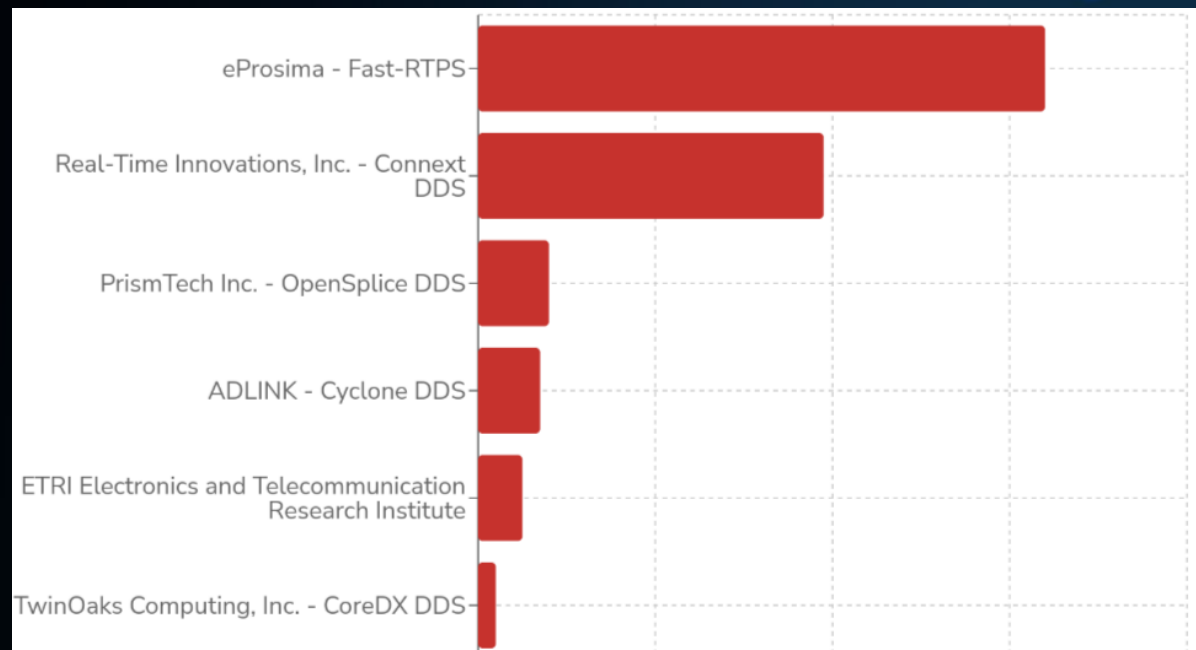


掃描 DDS 的難點

- 規格上沒有指定 Listen Port
 - Multicast Port = $7400 + 250 * d$, $d = 0..250$
 - Port 可以被映射，亦需要判斷返回內容 (e.g. 有可能 HTTPS 在 7400)
- 某些實作只會處理自家 Vendor ID 傳來的封包

觀察結果

- 暴露在外的 DDS 有獲減少
 - 2021/12: ~350, 2022/8: ~170
- FastRTPS (ROS2) 依舊最多
- 某些國家的暴露數減少許多 (>50%)



結論

- 在結論之前...

「良好的軟體漏洞修補流程，
建立在完善的漏洞回報流程之上。」

--Federico Maggi

良好漏洞修補 = 良好漏洞回報

- 我們試著與各廠商以透明的方式合作
 - 有用的部分：廠商願意協助我們「研究」他們的產品，進而使其更安全
 - 不太有用的部分：
即使以不同的管道連絡，有些廠商從來沒有回覆過
- 吾人應試著將廠商納入漏洞研究與挖掘的流程
 - 我們讚揚 ADLINK (凌華科技) 的勇敢和真誠

結論

- 今日的 DDS 應用發展已十分磅礴，但其安全性仍需加強
- 規格並不代表一切 – 有時完全按照規格實作是有危險的
- 我們釋出工具：
 - <https://github.com/the-dds>
 - 其餘的已 Upstream 到 Scapy 等專案
 - 我們亦貢獻 oss-fuzz 的數個整合

Questions?

- talun_yen@txone.com
- [@evanslify](#)