Arm IDA and Cross Check: Reversing the 787's Core Network

Ruben Santamarta
Principal Security Consultant

IOActive.


BACK
TO THE FUTURE
IOActive

2008

WIRED SUBSCRIBE

KIM ZETTER BUSINESS 01.04.08 12:00 PM

FAA: BOEING'S NEW 787 MAY BE VULNERABLE TO HACKER ATTACK

https://www.wired.com/2008/01/dreamliner-security/

"The proposed architecture of the 787 is different from that of existing production (and retrofitted) airplanes. ***It allows new kinds of passenger connectivity to previously isolated data networks connected to systems that perform functions required for the safe operation of the airplane****. Because of this new passenger connectivity, the proposed data network design and integration* ***may result in security vulnerabilities from intentional or unintentional corruption of data and systems critical to the safety and maintenance of the airplane****"*

https://www.federalregister.gov/documents/2008/01/02/E7-25467/special-conditions-boeing-model-787-8-airplane-systems-and-data-networks-security-isolation-or

IOActive.

2015

Is It Possible for Passengers to Hack Commercial Aircraft?

> Lemme says there may be some aircraft that now use ethernet connections in place of ARINC 429 buses to transmit data from the avionics to the entertainment system. But in a design like that, he says, there would be a box sitting between the avionics system and the in-flight system to securely convey information to the latter without allowing a connection back to the avionics from the IFE.

https://www.wired.com/2015/05/possible-passengers-hack-commercial-aircraft/

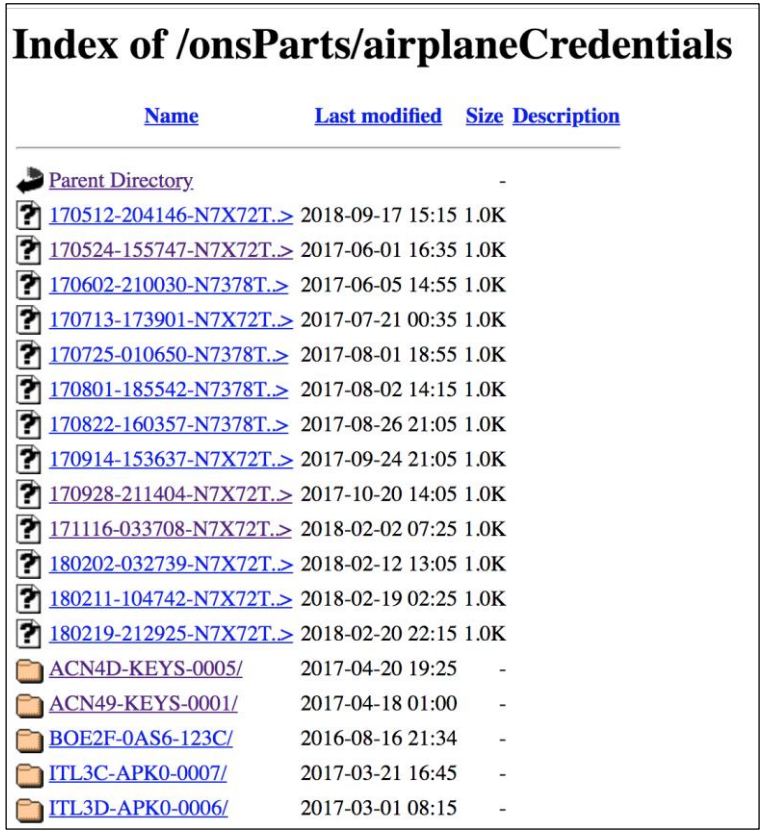
IOActive.

787's Core Network Cabinet

https://www.redimec.com.ar/producto-isc2200-information-services-cabinet-601

IOActive.

Old tricks never die…



Index of /onsParts/airplaneCredentials

Name	Last modified	Size	Description
Parent Directory		-	
170512-204146-N7X72T..>	2018-09-17 15:15	1.0K	
170524-155747-N7X72T..>	2017-06-01 16:35	1.0K	
170602-210030-N7378T..>	2017-06-05 14:55	1.0K	
170713-173901-N7X72T..>	2017-07-21 00:35	1.0K	
170725-010650-N7378T..>	2017-08-01 18:55	1.0K	
170801-185542-N7378T..>	2017-08-02 14:15	1.0K	
170822-160357-N7378T..>	2017-08-26 21:05	1.0K	
170914-153637-N7X72T..>	2017-09-24 21:05	1.0K	
170928-211404-N7X72T..>	2017-10-20 14:05	1.0K	
171116-033708-N7X72T..>	2018-02-02 07:25	1.0K	
180202-032739-N7X72T..>	2018-02-12 13:05	1.0K	
180211-104742-N7X72T..>	2018-02-19 02:25	1.0K	
180219-212925-N7X72T..>	2018-02-20 22:15	1.0K	
ACN4D-KEYS-0005/	2017-04-20 19:25	-	
ACN49-KEYS-0001/	2017-04-18 01:00	-	
BOE2F-0AS6-123C/	2016-08-16 21:34	-	
ITL3C-APK0-0007/	2017-03-21 16:45	-	
ITL3D-APK0-0006/	2017-03-01 08:15	-	

- **September 2018**
- **Publicly available Boeing server**
- **Google query**

Files

- **787's Core Network Cabinet Fw**
- **737's Onboard Network System Fw**
- **VM to VPN into a Boeing network**

IOActive.

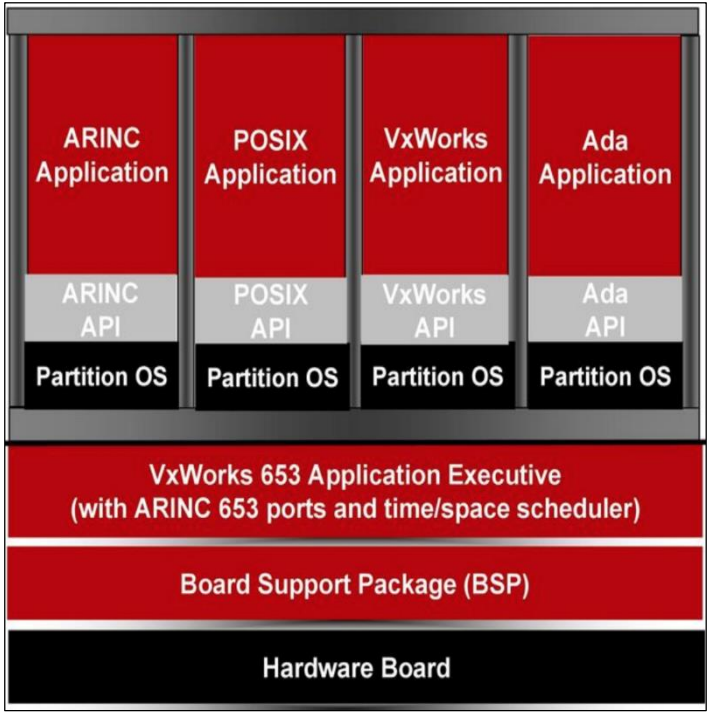
Methodology

- Information Gathering
  - Documents, multimedia material, presentations, papers, press releases, patents, books, etc.
- Reverse Engineering
  - Identify the elements, components, and functionalities described in the patents
  - Identify attack vectors
  - Prioritize attack areas
  - Find a minimum set of vulnerabilities required to demonstrate each of the attack scenarios described in 2.2
  - Assess the exploitability and post-exploitation scenarios, which included reviewing the machine code for the presence of compiler-level mitigations.
  - Evaluate the overall security posture of the in-scope elements taking information and knowledge gained in the previous phases into account

Boeing 787 Overview

Common Core System

- General Processing Modules
- Remote Data Concentrators
- A664-P7 Network



http://www.artist-embedded.org/docs/Events/2007/IMA/Slides/ARTIST2_IMA_WindRiver_Wilson.pdf

IOActive.

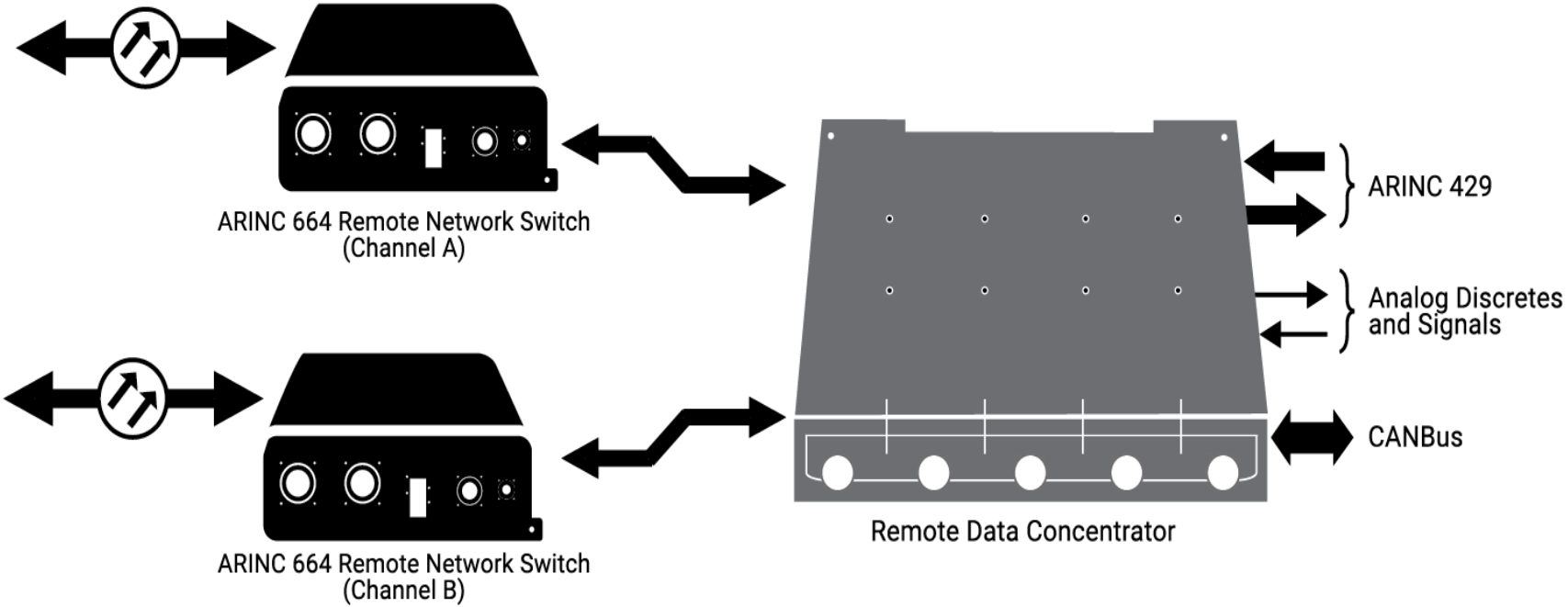
Common Computing Resource Cabinets

- Two (2) Power Conditioning Modules (PCMs)
- Eight (8) General Processing Modules (GPMs)
- Two (2) ARINC 664-P7 network Cabinet Switches (ACSs)
- Two (2) Fiber Optic Translator Modules (FOXs)
- Two (2) Graphic Generators (part of the Display and Alert Crew System)

GPM Hosted Functions

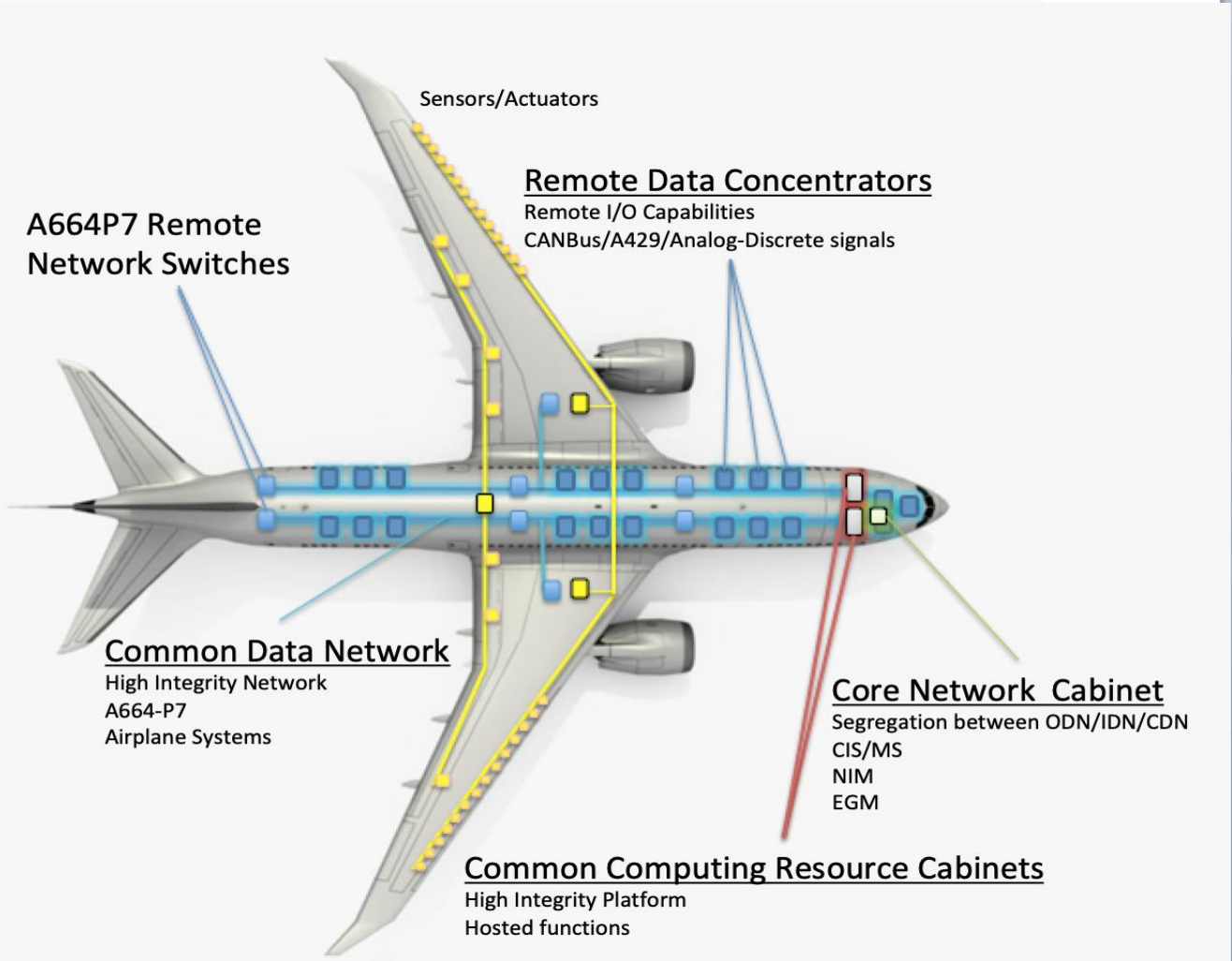
- Cabin Air Temperature Control System
- Remote Power Distribution System (RPDS)
- Power Distribution Panels (PDPs)
- Generator/Bus Power Control Units (GCU/BPCU)
- Low Pressure System
- Fuel Quantity System
- Hydraulic System Control
- Power Electronics Cooling System
- Communication Management Function
- Landing Gear Indication and Control
- Flight Management Function
- Circuit Breaker Indication and Control
- Electrical Power Distribution and Control
- …

Remote Data Concentrators



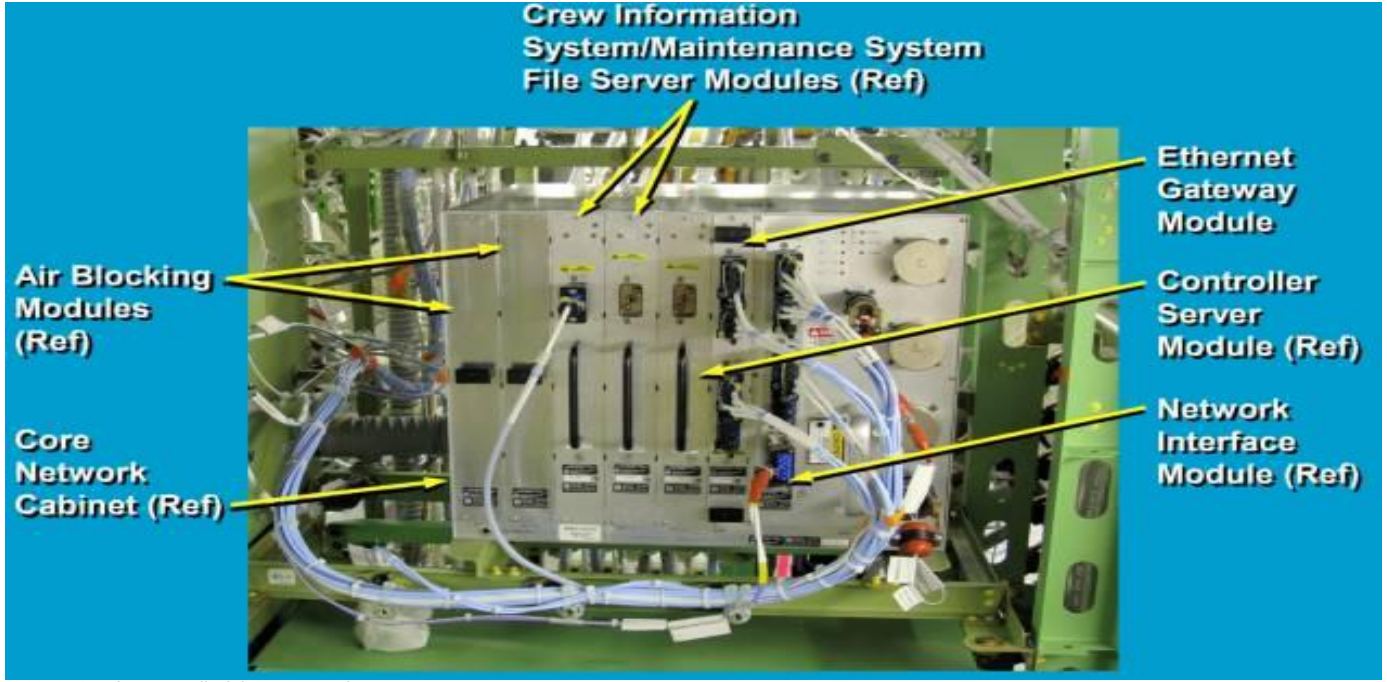
IOActive.

Common Data Network



Crew Information System

Interface between the ‘outside world’/non-critical domains and the CCS



IOActive.

CIS Networks

- ODN (Open Data Network)
  - Non-critical aircraft domains
  - External Networks
  - Potentially hostile devices: IFE, SATCOM, TWLU/CWLU…
- IDN
  - Secure, but non-safety, devices.
    - Voice and Data Flight Recorders (Black Box)
    - Electronic Flight Bag
- CDN
  - High Integrity Network (Avionics)
  - Airplane systems

CIS Modules

- CIS/MS FSM
  - VxWorks 6.2 (x86) COTS Board (CPB4612)
  - RTPs
- EGM
  - Linux ZNYX ZX4500
- NIM
  - End System is a GE's ASIC

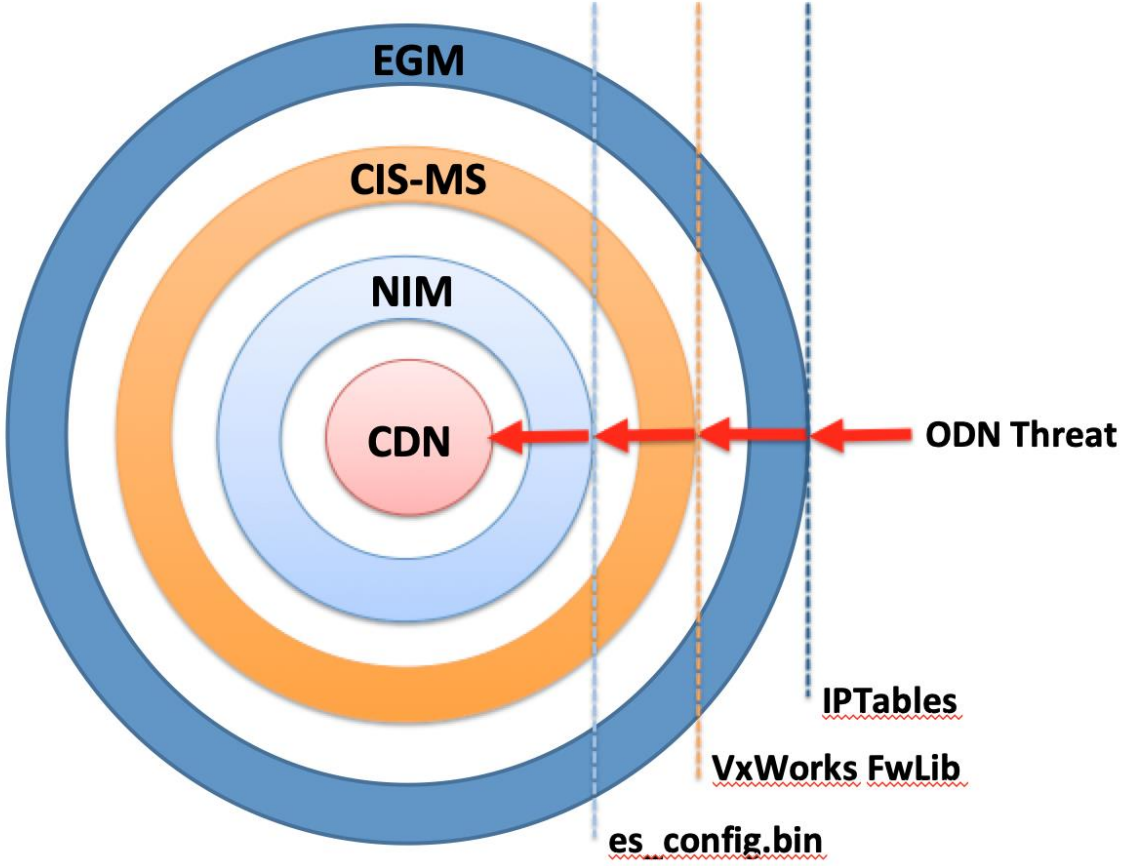
Attack Surface of the Core Network Cabinet

- *“The architecture provides segregation between network devices in the IDN and CDN related to operation and navigation of the vehicle, and network devices in the ODN”*

US Patent 7756145 B2

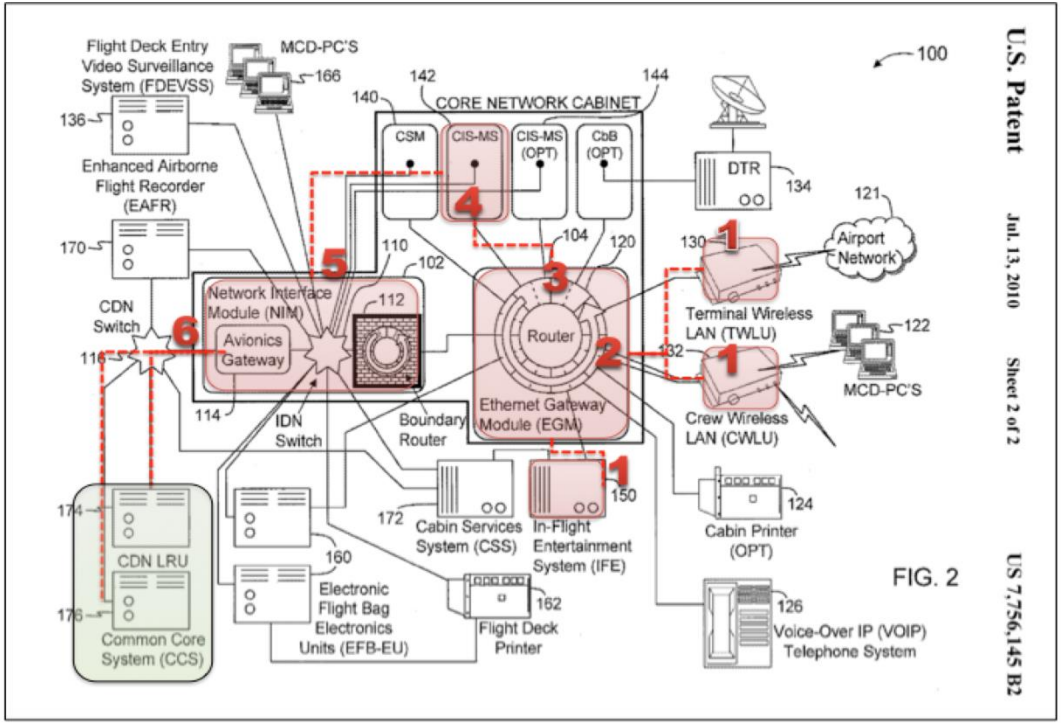
https://patents.google.com/patent/WO2007117285A2/en

Security Boundaries



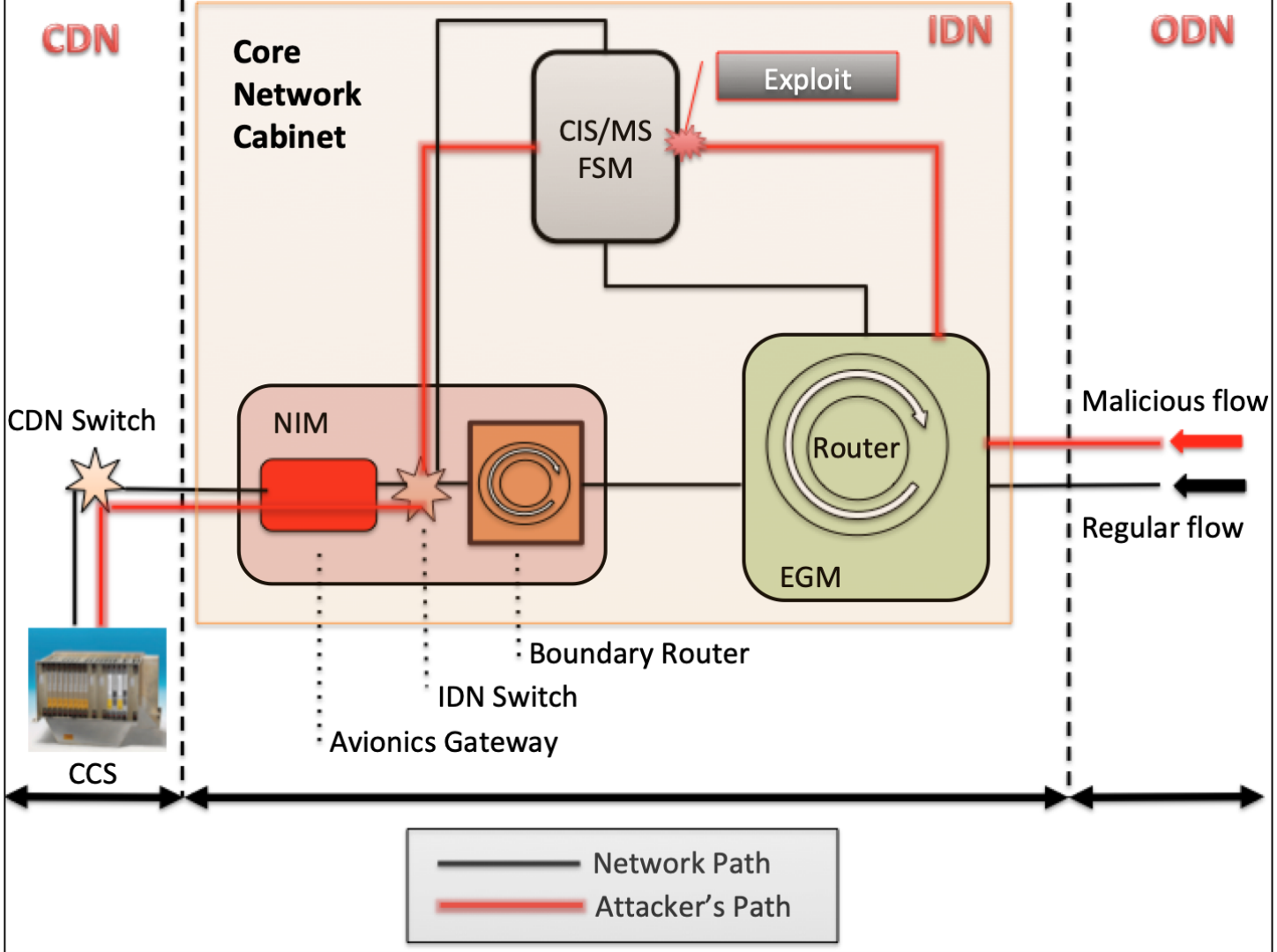
From ODN to CDN

1. ODN attack vectors
2. Iptables Rules (EGM)
3. EGM to CIS/MS rule
4. RCE on CIS/MS
5. Unblock Access to CDN
6. Jump to CDN through NIM



IOActive.

Attack Scenario



CIS/MS FSM

VxWorks Kernel

User-Mode

Real-Time Processes

1. FBM.vxe
2. FTS_Manager.vxe
3. MSPE.vxe
4. OBEDS.vxe
5. ODLF.vxe
6. bmt.vxe
7. fsmTgtLdr.vxe
8. ftpd.vxe
9. mtf_main.vxe
10. mtf_rtp.vxe
11. omls.vxe
12. osm.vxe
13. rexec_server.vxe
14. wlanmf_rtp.vxe

Shared Libraries

1. ACP.so
2. AMI.so
3. DiskUtilities.so
4. DisplayUtilities.so
5. FCCS.so
6. FSMAAircraftVerification.so
7. JSON.so
8. LDI.so
9. Messaging.so
10. OBEDSInterface.so
11. OrderedList.so
12. SNMP.so
13. cisUtil.so
14. mtfIOUtilities.so
15. ossAccessors.so

CIS/MS Vulnerabilities

- Hundreds of references to insecure functions (‘sprint’,’strcpy’,’strcat’..)
- Integer overflows
- Buffer Overflows
- Denial of Service
- Memory Corruption
- Out-Bound-Read/Write
- …

Breaking into the CIS/MS

- Minimum set of vulnerabilities that allow the CIS/MS to be compromised from the ODN
  - Remote Code Execution
  - Privilege Escalation to Kernel

FTS_Manager.vxe - TFTP Opcode Stack Overflow

```
value = recvfrom(
          serversocket,
          &requestbuffer,
          0x200u,
          0,
          (struct sockaddr *)&clientaddr,
          &clientaddrlength);
if ( value == -1 )
{
  sprintf(&log_buffer, "TFTP --> could not read on TFTP port %d", fs_listen_port[server_instance]);
  rtpLog(3, 0, &log_buffer);
  goto LABEL_107;
}
opcode = ((unsigned __int8)requestbuffer << 8) | ((requestbuffer & 0xFF00) >> 8);
strncpy(&fileforoptneg, (const char *)&requestbuffer + 2, 0x80u);
v20 = 0;
sprintf(
    &log_buffer,
    "%s --> %s Request Received for file %s from ",
    &fs_tftp_task_name[20 * server_instance],
    &opcode_string[5 * opcode],
    &fileforoptneg);
```


IOActive.

Exploitability

- **Controlled parameters**
  - **Destination File**
  - **Opcode**
  - **Adjacent memory**
  - **0x4FFFB (.data+.bss)**
  - **~0x700 bytes**

```
&opcode_string[5 * opcode],
```

```
.data:080DF040                  public opcode_string
.data:080DF040 opcode_string    db 'INV',0
.data:080DF044                  db    0
.data:080DF045 aRrq             db 'RRQ',0
.data:080DF049                  db    0
.data:080DF04A aWrq             db 'WRQ',0
.data:080DF04E                  db    0
.data:080DF04F aData            db 'DATA',0
.data:080DF054 aAck             db 'ACK',0
.data:080DF058                  db    0
.data:080DF059 aErr_0           db 'ERR',0
.data:080DF05D                  db    0
.data:080DF05E aOack            db 'OACK',0
.data:080DF063                  db    0
```


IOActive.

Exploitability

- From ODN to CIS/MS' 'FTS_Manager.vxe' Service
- EGM Iptable Rules

File: 'S24egmcfg'

```
# VLAN 140; In-Flight Entertainment System
zconfig zhp22 : vlan140=zre20
….
iptables -A IFE -j ACCEPT -i zhp22 -s 172.27.40.2 -d 172.24.10.12 -p udp --sport 1024:65535 --dport 16005
```

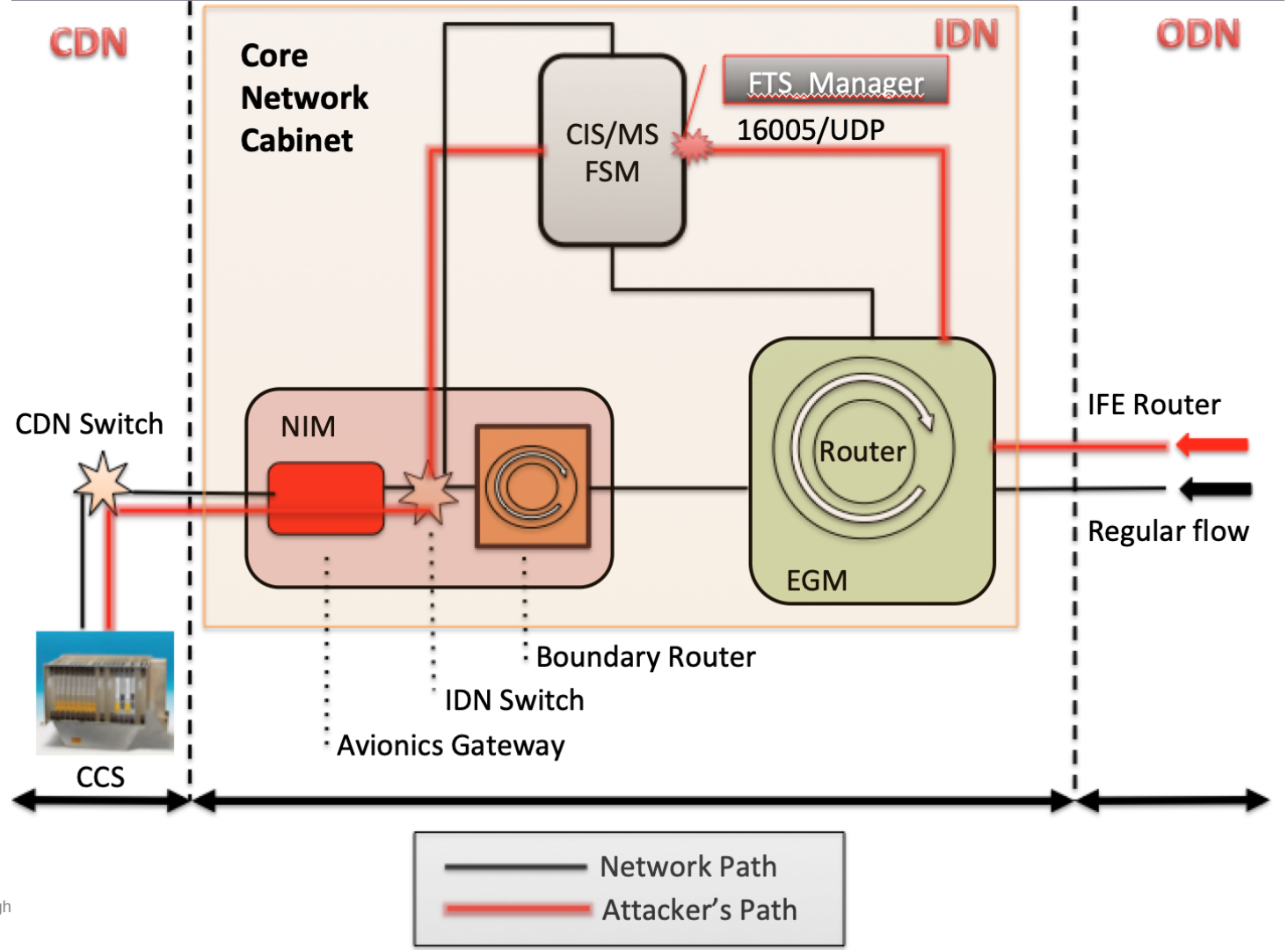
- 172.27.40.2 - ife-router.odn.pnet
- 172.24.10.12 - cis-ms-active.idn.pnet

Exploitability

- File ‘AimCfg.xml’
- NIM’s Boundary Router

```
<!-- *** IFE FTS *** -->
<!-- CABINET-CABIN_EQPMT_CENTER-IFE interfaces with APP-CIS MS ACTIVE   (FTS Service) -->
<Rule chain="FORWARD" target="ACCEPT">
    <Parameters>-i eth0 -p udp -s 172.27.40.2 --sport 1024:65535 -d 172.24.10.12 --dport 16005</Parameters>
    <Tag>Boundary router rule</Tag>
    <Sanction>Do-Nothing</Sanction>
</Rule>
```

- 172.27.40.2 - ife-router.odn.pnet
- 172.24.10.12 - cis-ms-active.idn.pnet



‘duParseLUSFile’ Memory Corruption

- ‘diskUtils.so’
  - Exercise vulnerable path from ‘ODLF.vxe’ (Onboard Data Loading Function)
  - Caller allocates a fixed fileData structure (stack)
  - buffer is pointing to the attacker controlled LUS file.

```
for ( idxf = bytesReadc; idxf < bytesReadc + 2; ++idxf )
  fileData->numberHeaders += buffer[idxf] << (8 - 8 * (idxf - bytesReadc));
bytesReadd = bytesReadc + 2;
for ( hdrIndex = 0; hdrIndex < fileData->numberHeaders; ++hdrIndex )
{
  fileData->files[hdrIndex].headerNameLength = 0;
  for ( idxg = bytesReadd; idxg < bytesReadd + 1; ++idxg )
    fileData->files[hdrIndex].headerNameLength += buffer[idxg] << -8 * (idxg - bytesReadd);
  v4 = bytesReadd + 1;
  strncpy(fileData->files[hdrIndex].headerName, (const char *)&buffer[v4], fileData->files[hdrIndex].headerNameLength);
  fileData->files[hdrIndex].headerName[fileData->files[hdrIndex].headerNameLength] = 0;
  bytesReade = fileData->files[hdrIndex].headerNameLength + v4;
  fileData->files[hdrIndex].partNumberLength = 0;
```

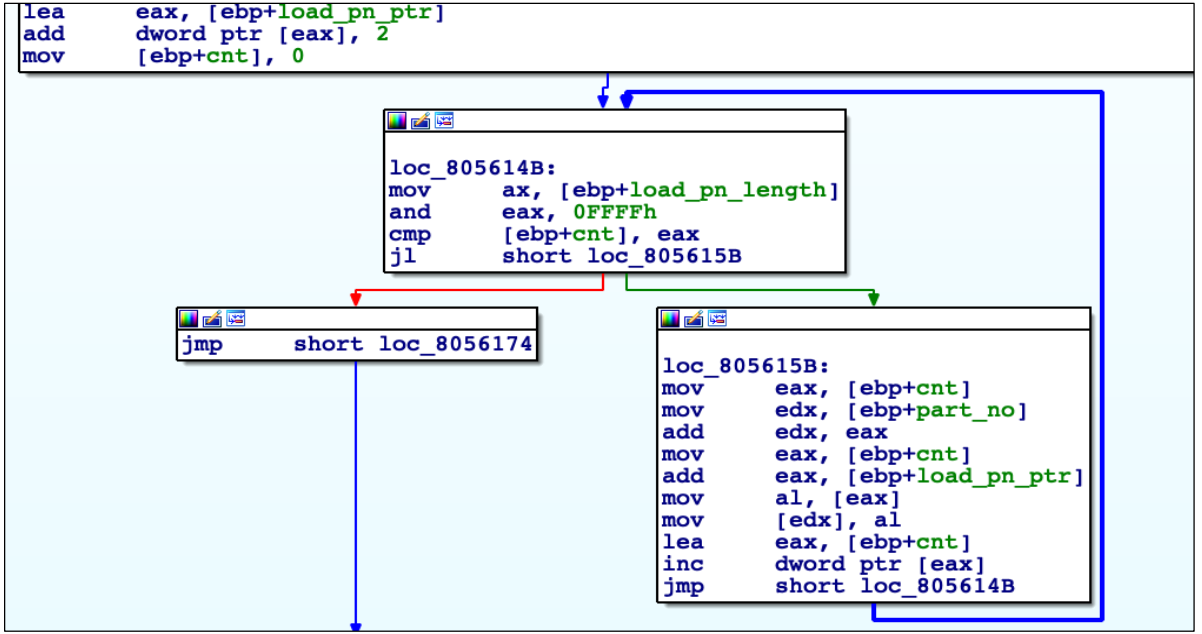
Exploitability

- The attacker can corrupt the stack buffer in a solid way, using controllable values from the LUS file, which allows the attacker to effectively control the EIP and those registers required to initiate a ROP chain (if it's ever required)*

- A remote unauthenticated attacker can exercise the vulnerable execution path. It is worth mentioning that any compromised LRU that is about to be updated may also trigger this vulnerability as the ODLF acts as a server/client.

FsmTgtLdr.vxe – LUH Part Number Stack Overflow

- When parsing .LUH files (ARINC Load Upload Headers), the part number length is not properly checked.



Exploitability

- The attacker is able to corrupt the stack buffer in a solid way, using controllable values from the LUH file, which allows the attacker to effectively control the EIP and those registers required to initiate a ROP chain.
- A remote unauthenticated attacker can exercise the vulnerable execution path.

VxWorks – Insecure Syscall Handlers Privilege Escalation

- RCE done, now let's jump to the Kernel.
- CIS/MS Custom SysCall Group 'FSMSYSTEM'
- Invoked from user mode by using a CallGate

```
.data:0081C020                  public syscallGroupTbl
.data:0081C020 syscallGroupTbl  dd 0                    ; DATA XREF: rtpSysctlSyscall+4D↑r
.data:0081C020                                          ; rtpSysctlSyscall+AE↑r ...
.data:0081C024 dword_81C024     dd 0                    ; DATA XREF: rtpSysctlSyscall+11E↑r
.data:0081C024                                          ; syscallGroupRegister+12E↑w ...
.data:0081C028                  align 10h
.data:0081C030                  dd offset FSMSYSTEMRtnTbl
.data:0081C034                  db  26h ; &
```


IOActive.

VxWorks – Insecure Syscall Handlers Privilege Escalation

- They are not validating any pointer received from user-mode, so it is possible to read/write arbitrary kernel memory
- They use insecure functions and other insecure patterns, which can be used to trigger different kinds of vulnerabilities.

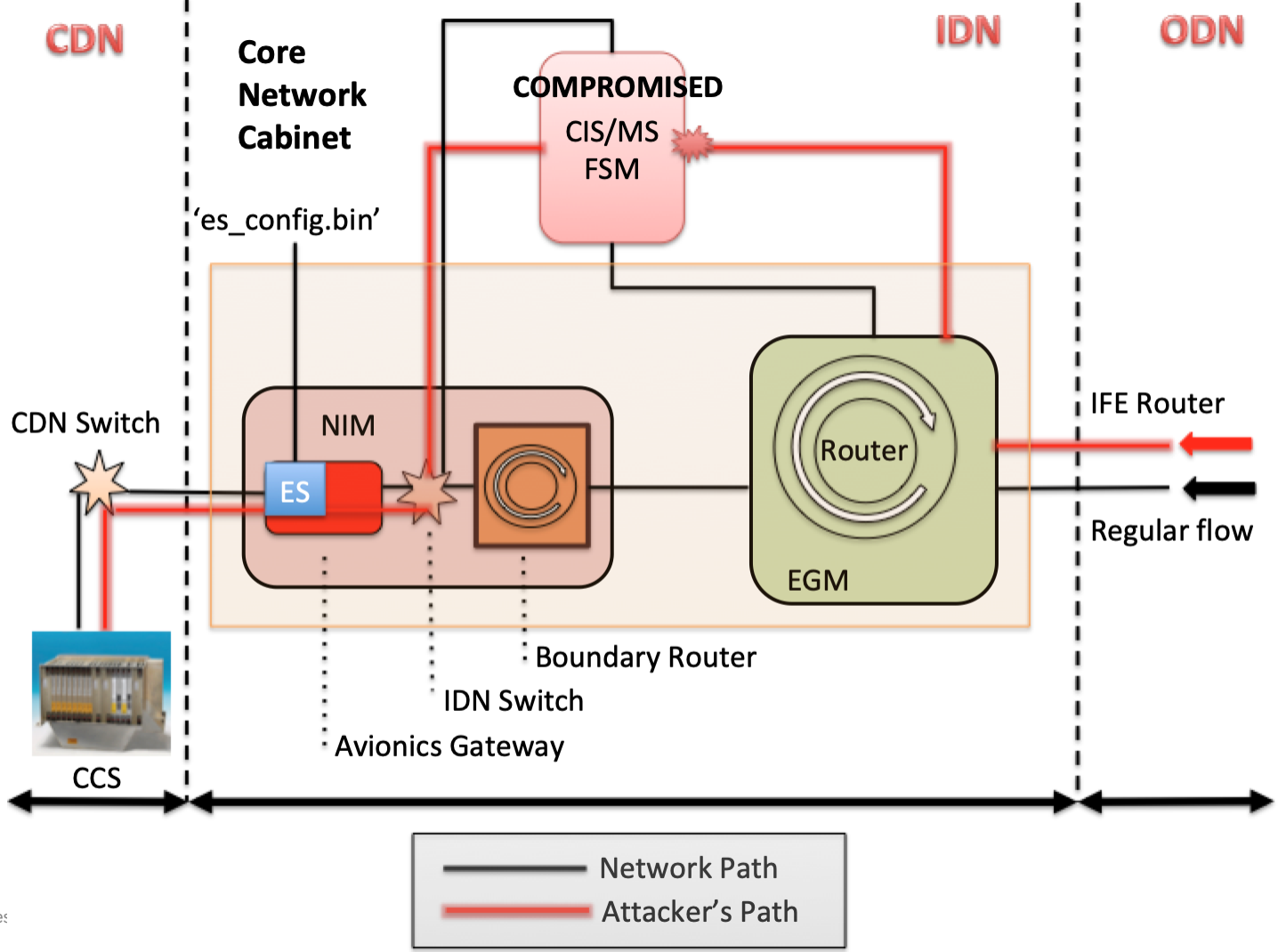
‘cissFwSetByDynFirewallRule’ SysCall (0x224)

- It enables RTPs to add arbitrary firewall rules to the CIS/MS packet filter
- Useful during exploitation to unblock CDN access.

Syscall 0x224

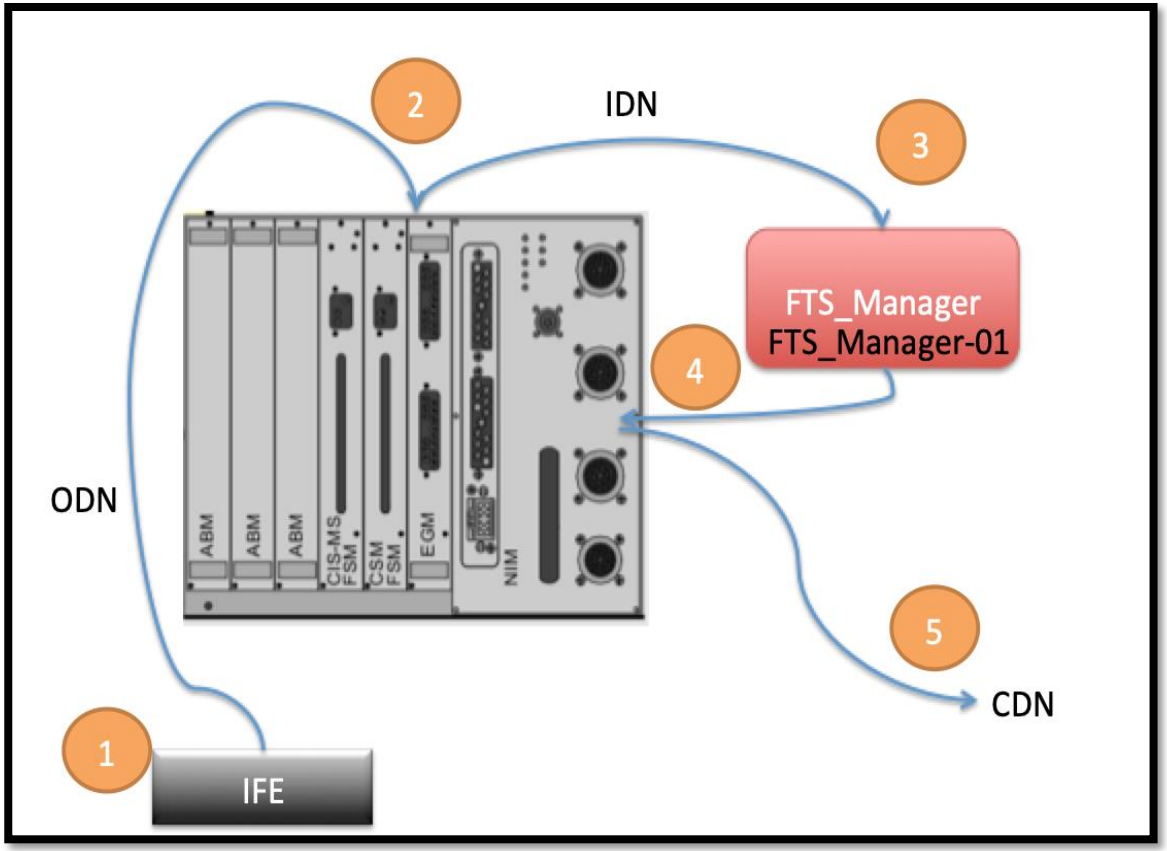
```
.text:003246E3 ; Attributes: bp-based frame
.text:003246E3
.text:003246E3                 public cissFwSetDynFirewallRuleSc
.text:003246E3 cissFwSetDynFirewallRuleSc proc near     ; DATA XREF: .data:0081BFF0↓o
.text:003246E3
.text:003246E3 arg_0           = dword ptr  8
.text:003246E3
.text:003246E3                 push    ebp
.text:003246E4                 mov     ebp, esp
.text:003246E6                 sub     esp, 18h
.text:003246E9                 mov     eax, [ebp+arg_0]
.text:003246EC                 mov     eax, [eax+0Ch]
.text:003246EF                 mov     [esp+0Ch], eax
.text:003246F3                 mov     eax, [ebp+arg_0]
.text:003246F6                 mov     eax, [eax+8]
.text:003246F9                 mov     [esp+8], eax
.text:003246FD                 mov     eax, [ebp+arg_0]
.text:00324700                 mov     eax, [eax+4]
.text:00324703                 mov     [esp+4], eax
.text:00324707                 mov     eax, [ebp+arg_0]
.text:0032470A                 mov     eax, [eax]
.text:0032470C                 mov     [esp], eax
.text:0032470F                 call    cissFwSetDynFirewallRule
.text:00324714                 leave
.text:00324715                 retn
.text:00324715 cissFwSetDynFirewallRuleSc endp
.text:00324715
.text:00324716
.text:00324716 ; =============== S U B R O U T I N E =======================================
.text:00324716
.text:00324716 ; Attributes: bp-based frame
.text:00324716
.text:00324716                 public cissFwShowRulesSc
.text:00324716 cissFwShowRulesSc proc near              ; DATA XREF: .data:0081C000↓o
.text:00324716                 push    ebp
.text:00324717                 mov     ebp, esp
.text:00324719                 sub     esp, 18h
.text:0032471C                 mov     dword ptr [esp+4], offset aCissfwshowrule ; "cissFwShowRulesSc: Showing Firewall Rul"...
.text:00324724                 mov     dword ptr [esp], offset cissFwlogTmp_0
.text:0032472B                 call    sprintf
.text:00324730                 mov     dword ptr [esp+8], offset cissFwlogTmp_0
.text:00324738                 mov     dword ptr [esp+4], offset aCissfw_0 ; "CISSFW"
.text:00324740                 mov     dword ptr [esp], 6
.text:00324747                 call    kernLog
.text:0032474C                 mov     dword ptr [esp], 0
.text:00324753                 call    fwRulesShow
.text:00324758                 mov     dword ptr [esp+4], offset aCissfwshowru_0 ; "cissFwShowRulesSc: Showing Firewall Rul"...
.text:00324760                 mov     dword ptr [esp], offset cissFwlogTmp_0
.text:00324767                 call    sprintf
.text:0032476C                 mov     dword ptr [esp+8], offset cissFwlogTmp_0
.text:00324774                 mov     dword ptr [esp+4], offset aCissfw_0 ; "CISSFW"
.text:0032477C                 mov     dword ptr [esp], 6
.text:00324783                 call    kernLog
.text:00324788                 mov     dword ptr [esp], 2
.text:0032478F                 call    fwRulesShow
.text:00324794                 mov     dword ptr [esp+4], offset aCissfwshowru_1 ; "cissFwShowRulesSc: Firewall Logging is "...
.text:0032479C                 mov     dword ptr [esp], offset cissFwlogTmp_0
.text:003247A3                 call    sprintf
.text:003247A8                 mov     dword ptr [esp+8], offset cissFwlogTmp_0
.text:003247B0                 mov     dword ptr [esp+4], offset aCissfw_0 ; "CISSFW"
.text:003247B8                 mov     dword ptr [esp], 6
.text:003247BF                 call    kernLog
.text:003247C4                 leave
.text:003247C5                 retn
.text:003247C5 cissFwShowRulesSc endp
```

ODN to CDN



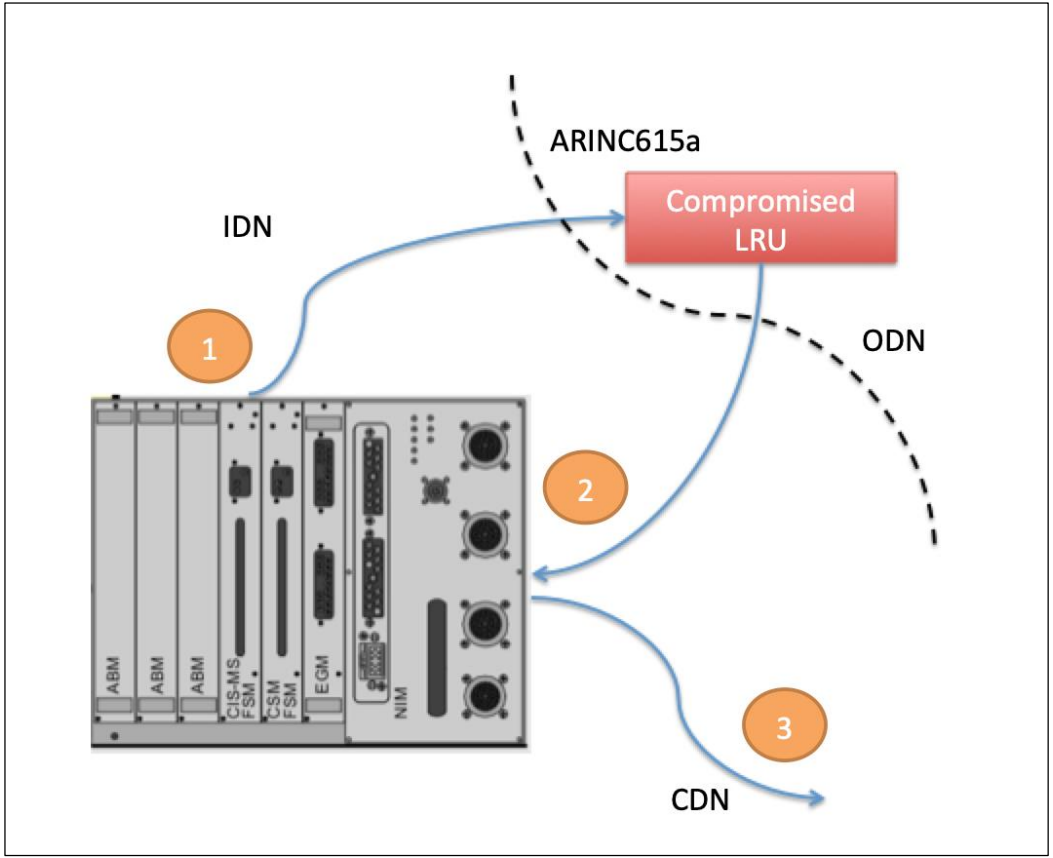
Attack Scenario #1 – IFE to CDN

- Compromise IFE
- EGM->CIS/MS
- TFTP Exploit
- SysCal 0x224
- CIS/MS to CDN



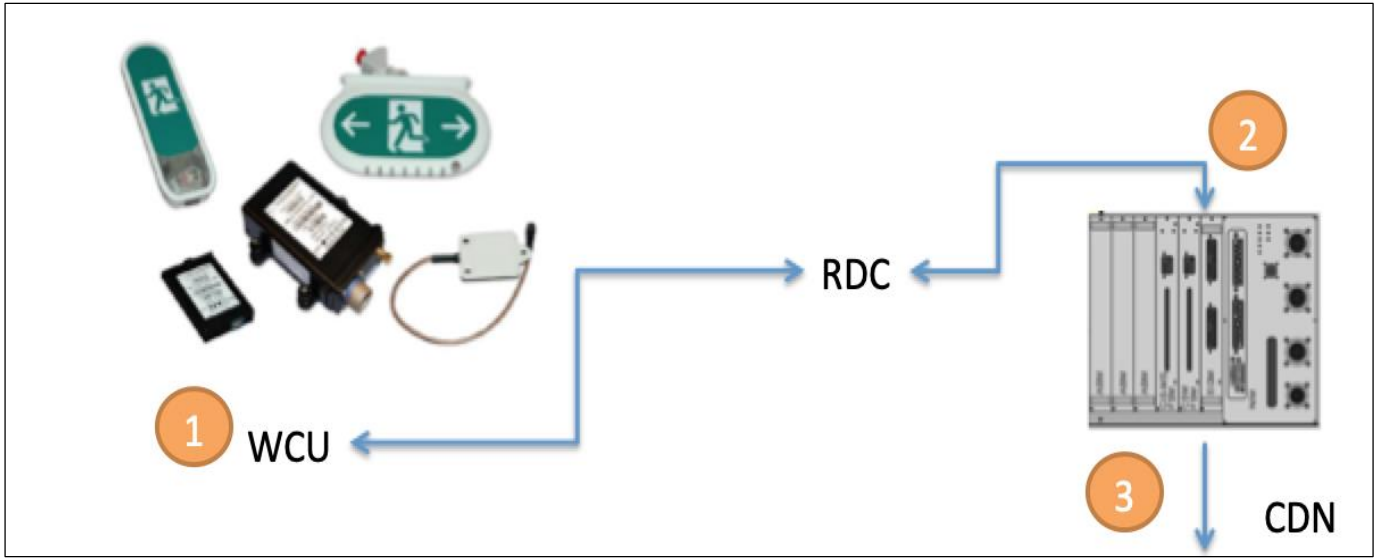
Scenario #2 – Arbitrary LRU to CDN

- CIS/MS Data Load
- Compromised LRU
- ODLF.vxe exploit
- LRU->CIS/MS->CDN



Scenario #2.1 – Wireless LRU to CDN

- Wireless Emergency Lighting System



https://www.securaplane.com/products/wireless/

IOActive.

Scenario #2.1 – Wireless LRU to CDN

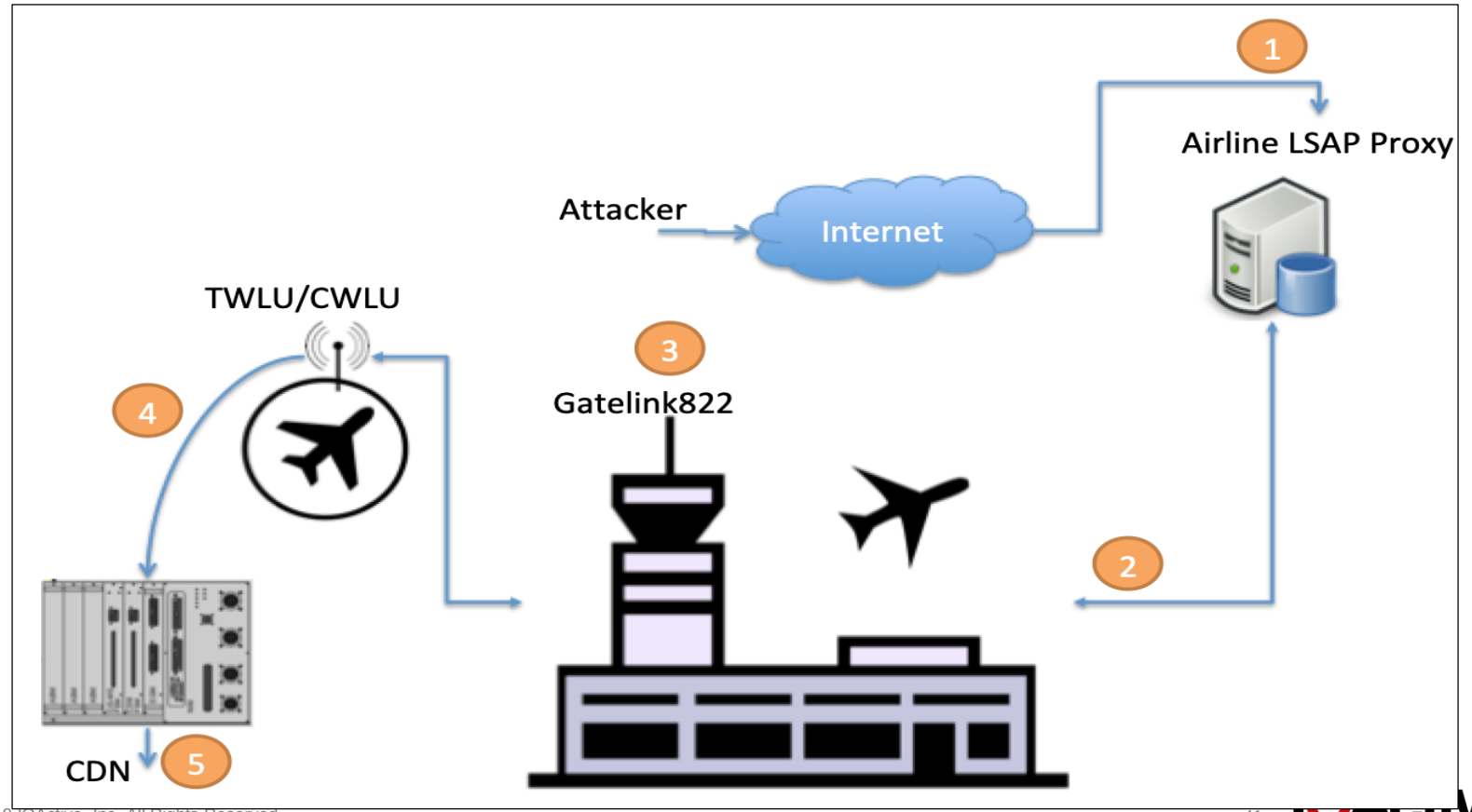
- Onboard attacker (or supply chain) compromises WCU
- WCU exploits ODLF.vxe vulnerabilities through RDC
- Attacker gains access to the CDN

File: ‘AimCfg.xml’

```
<!-- PDT = [CAN Data Load]   ODLF_200kb_CAN_S1_Tx1 |Usg to UNIT-WELS CONTROL-PRIME-DR 3L |Usg.R5_C2_ODLF_WKP_HFQI_10 |Usg -->
<!--*** APP-CIS MS ACTIVE-1 |Usg to UNIT-WELS CONTROL-PRIME-DR 3L |Usg ***-->
<Uni_Tx>
    <CDN_Dest_IP>10.42.165.10</CDN_Dest_IP>
    <CDN_Dest_UDP>59</CDN_Dest_UDP>
    <Es_Tx_Port_ID>66112</Es_Tx_Port_ID>
    <IDN_Source_IP>172.24.10.12</IDN_Source_IP>
    <IDN_Source_UDP>62911</IDN_Source_UDP>
</Uni_Tx
```


IOActive.

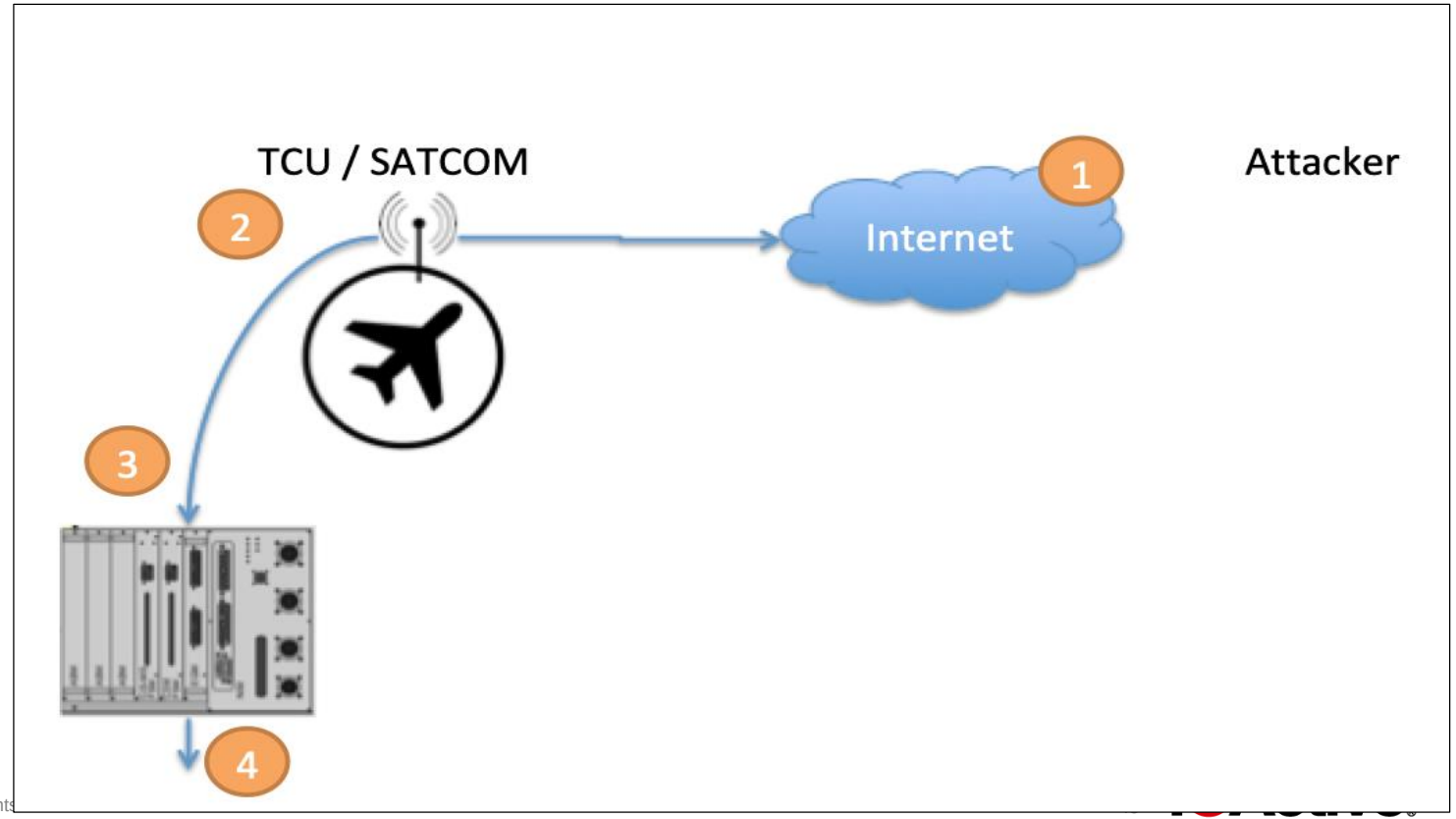
Scenario #3 – External Network to CDN



Scenario #3 – External Network to CDN

- An attacker compromises an **Internet-facing** vulnerable LSAP proxy server.
- The attacker controls LSAP repository/uplink-downlink requests (OBEDS.vxe/FTS_Manager.vxe)
- The Gatelink822 Airport's local infrastructure may also expose an attack vector.
- The attacker reaches the IDN through the TWLU/CWLU EGM rules.
- The attacker gains CDN access by exploiting any of the documented vulnerabilities.
- **IOActive discovered two vulnerable instances of Internet-accessible LSAP proxy servers belonging to airlines operating Boeing aircraft and shared the details with Boeing..**
- Gatelink822 infrastructure and reachability may vary between airports. As an example, in Terminal 4S of the Barajas Adolfo Suarez Airport in Madrid (Spain), the Gatelink822 SSID is publicly broadcast throughout the terminal.

Scenario #4 – Communication Link to CDN



Scenario #4 – Communication Link to CDN

- TCU/SATCOM providers may assign a public IP that is exposed to the Internet.
- An attacker gains access to the TCU/SATCOM device.
- The attacker reaches the CIS/MS through the EGM rules for TCU/SATCOM interfaces (if any, as SATCOM may be optional)
- The attacker gains CDN access by exploiting any of the vulnerabilities documented in the CIS/MS services.

Post Exploitation

- Initiate a **potentially** malicious firmware update in Safety Critical Units
  - Potential Mitigation: Integrity controls
- Abuse maintenance (Circuit breakers, tests..)
- Deceive maintenance engineers

We are just using the system in the way it was designed for.

From CDN to Safety Critical Systems

21 Remote Data Concentrators

- Main Engine Data Concentrators
- Brake System Control Cards
- Valve Control Circuit Cards
- Proximity Sensors Data Concentrators
- Electric Motor Pump Controller
- Electric Control Break Actuator
- Fuel Quantity
- Emergency Power Assist System
- Wireless Emergency Light System
- Ram Air Fan Controller
- Maintenance Display Unit
- Cabin Air Compressor
- Shutoff Fuel Module
- Refuel Control Panel
- Wing Ice Protection System
- Bus Power Control Unit
- Electronic Control Unit
- Secondary Power Distribution Unit
- Engine Monitor Unit
- Electronic Engine Control
- Remote Power Distribution Unit
- Graphics Generator Display
- Flight Recorder
- Audio Units

From CDN to Safety Critical Systems

Electronic Engine Controller – DataLoad A615A Rule

```
<!-- PDT = [ARINC 615A]   ODLF_10Mb_S2_Tx1 |Usg to CONTROLLER-EEC_CHANNEL==B-1 |Occ.EECB_Data_Load_Rx0_L |Occ -->
<!--*** APP-CIS MS ACTIVE-1 |Usg to CONTROLLER-EEC_CHANNEL==B-1 |Occ ***-->
<Uni_Tx>
    <CDN_Dest_IP>10.73.2.0</CDN_Dest_IP>
    <CDN_Dest_UDP>59</CDN_Dest_UDP>
    <Es_Tx_Port_ID>10009</Es_Tx_Port_ID>
    <IDN_Source_IP>172.24.10.12</IDN_Source_IP>
    <IDN_Source_UDP>62904</IDN_Source_UDP>
</Uni_Tx>
```

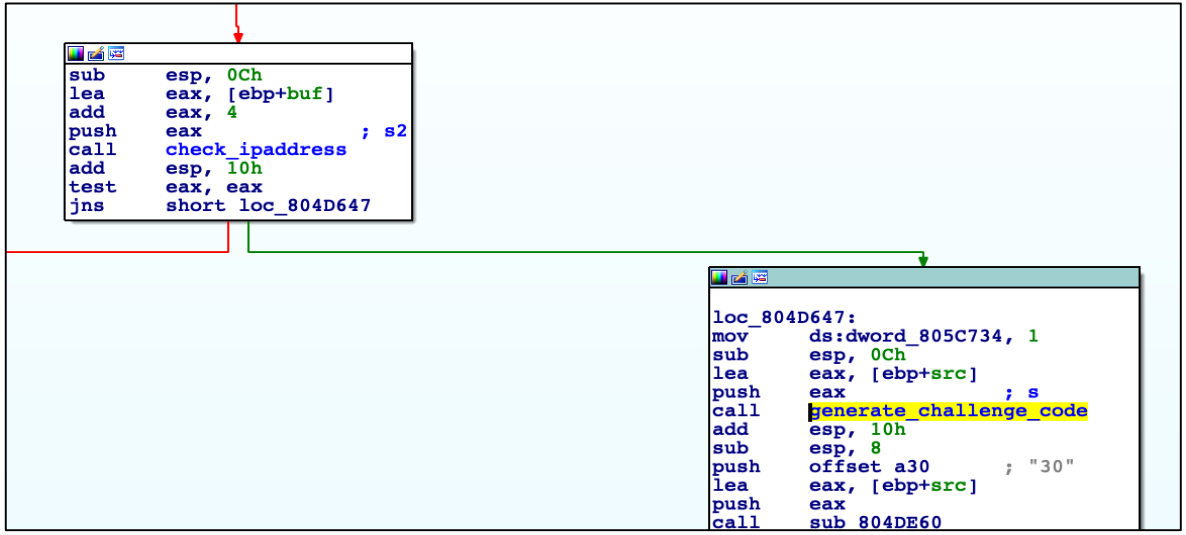
172.24.10.12 – CIS/MS FSM
10.73.2.0 – EEC_Controller
59 TFTP Port

Maintenance Operations

- **Wired:** When the engineer is connected through one of the three wired ports in the flight deck or equipment centers, it is possible to exercise any maintenance operation available on the system.
- **Wireless:** When the engineer is connected wirelessly through the CWLU/TWLU, only a limited set of maintenance operations are enabled.
- **Full Wireless:** This mode enables the engineer who is wirelessly connected through the CWLU/TWLU to ‘upgrade’ from a Wireless connection to a Full Wireless mode, which is equivalent to the Wired mode. In order to enable all of the operations, the engineer needs to enter a code that is generated in the CIS/MS through the cabin interphones. If the code entered matches the locally generated challenge code, the engineer is upgraded to Full Wireless mode, and the CIS/MS unblocks CDN access for the engineer’s Maintenance Terminal IP.

Maintenance Operations

- Challenge Code Generator – OMLS.vxe (Onboard Maintenance Laptop Function)



IOActive.

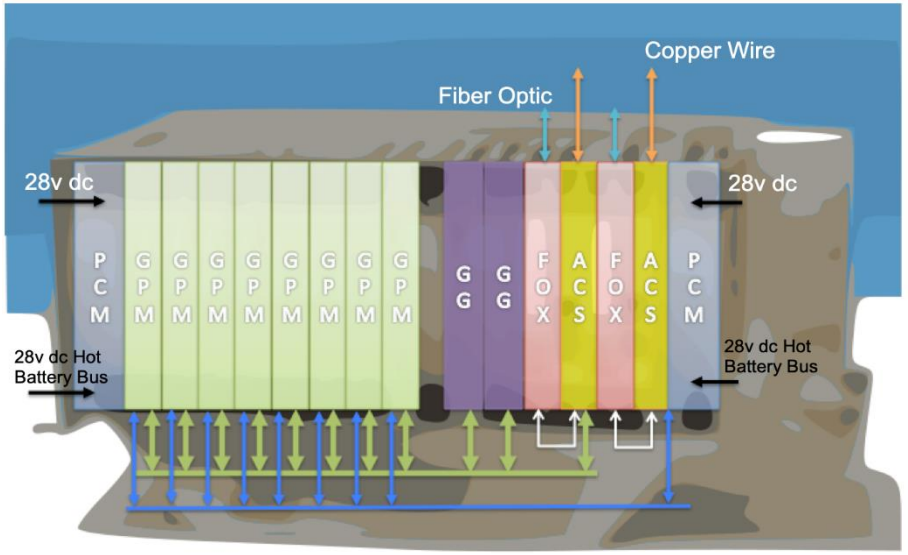
Maintenance Operators

- Challenge received from Cabin Interphone

```
text:0804DA78 loc_804DA78:                            ; CODE XREF: authentication_manager+B71↑j
text:0804DA78                 sub     esp, 8
text:0804DA7B                 lea     eax, [ebp+s1]
text:0804DA81                 push    eax             ; s
text:0804DA82                 push    dword ptr [ebp+addr.sa_data+2] ; char
text:0804DA88                 call    inet_ntoa_b
text:0804DA8D                 add     esp, 10h
text:0804DA90                 sub     esp, 8
text:0804DA93                 push    offset byte_805D14C ; s2
text:0804DA98                 lea     eax, [ebp+s1]
text:0804DA9E                 push    eax             ; s1
text:0804DA9F                 call    _strcmp
text:0804DAA4                 add     esp, 10h
text:0804DAA7                 test    eax, eax
text:0804DAA9                 jz      short loc_804DAE4
text:0804DAAB                 sub     esp, 4
text:0804DAAE                 lea     eax, [ebp+s1]
text:0804DAB4                 push    eax
text:0804DAB5                 push    offset aCabinInterphon ; "Cabin Interphone Message received from "...
text:0804DABA                 lea     eax, [ebp+var_6A8]
text:0804DAC0                 push    eax             ; s
text:0804DAC1                 call    _sprintf
text:0804DAC6                 add     esp, 10h
text:0804DAC9                 sub     esp, 4
text:0804DACC                 lea     eax, [ebp+var_6A8]
text:0804DAD2                 push    eax
text:0804DAD3                 push    0
```

Maintenance Operations

- Central Maintenance Computing Function (Hosted Function at CCS)
- Deceive Technicians
- Initiate tests
- Circuit Breakers



IOActive.

Responsible disclosure

- Boeing and Honeywell confirmed that these vulnerabilities are present in the current 787's Core Network codebase
- The official response IOActive received from Boeing was that they do not consider our reported findings exploitable vulnerabilities, as they could not reproduce these flaws.
- Boeing stated that they have 'compiler-level' mitigations in place that prevent the vulnerabilities from being exploited. No further details were shared.

Responsible disclosure

- Versions

Boeing did not share with IOActive the version of the CIS/MS firmware they were using in their testing, despite the fact that this information was requested several times. This is a crucial part in any responsible vulnerability disclosure, even more important when discrepancies in the results exist.

- Testing plan

During the vulnerability coordination process IOActive did not have any visibility over the tests, methodologies, proof-of-concept code, exploitation techniques, or any technical details in general terms, that Boeing and partners implemented during their internal evaluation of the vulnerabilities. To help address this situation, IOActive offered to assist Boeing in reproducing these vulnerabilities at their own controlled environment. Unfortunately, Boeing declined.

Responsible Disclosure

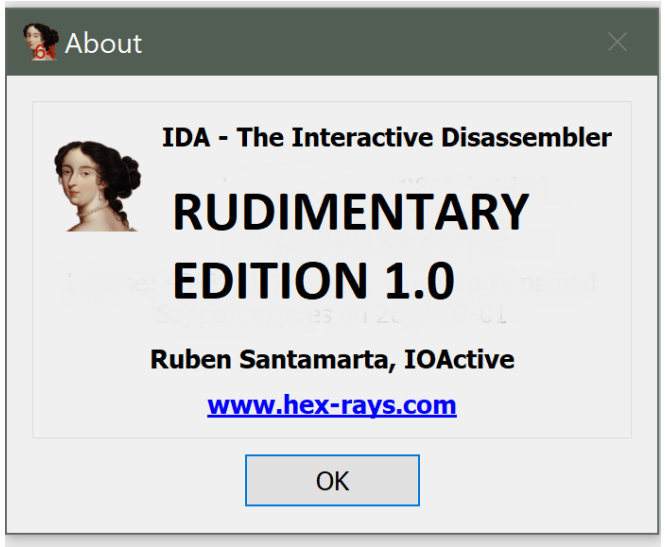
- Mitigations

Boeing communicated to IOActive that there are certain built-in compiler-level mitigations that, in their point of view, prevent these vulnerabilities from being successfully exploited. IOActive was unable to locate or validate the existence of those mitigations in the CIS/MS firmware version we analyzed. **When asked, Boeing declined to answer whether these mitigations might have been added on a later version.**
Honeywell is checking it *today*

Response from Boeing

- "*IOActive reviewed only one part of the 787 network using **rudimentary** tools*"

"IOActive chose to ignore our **verified** results"*



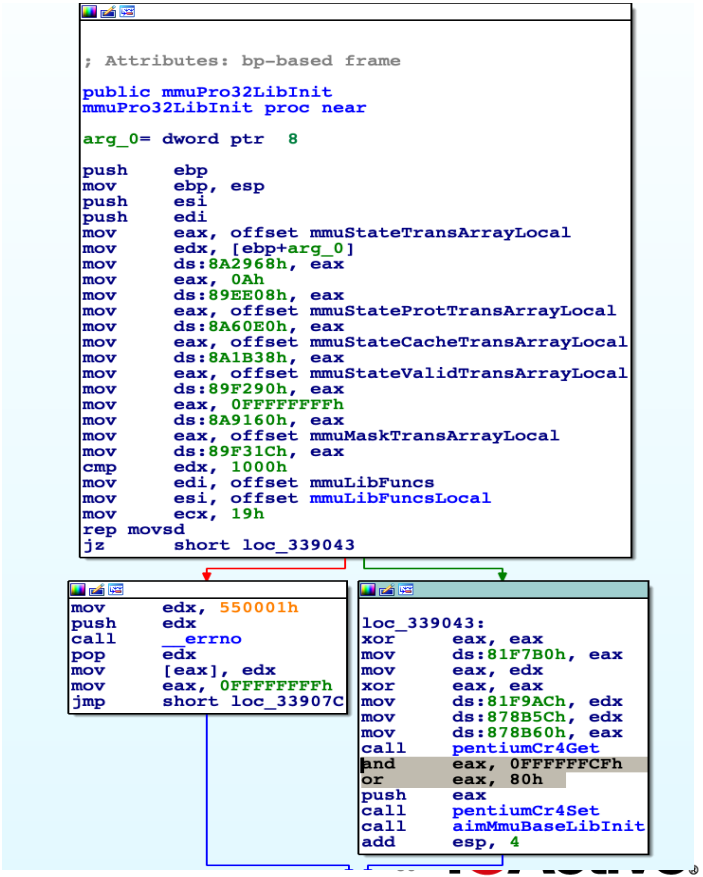
(*) Verified by those who consider that :

- IDA is a rudimentary tool
- 'sprintf' is an unexploitable function.
- A ROP chain in an RTOS is not possible because there are not enough functionalities
- Compiler-level mitigations can work even if they are not added to the resulting binary

IOActive.

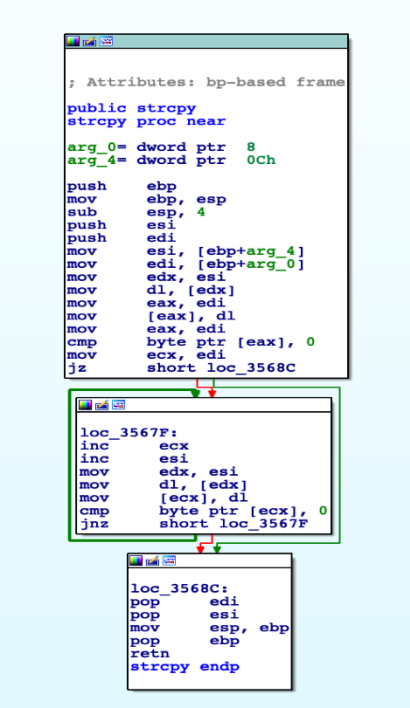
Mitigations (or the lack of them)

- NX/XD
  - 32-bit Pentium M
  - Unknown model
  - VxWorks MMU Initialization
    - PAE/PSE bits cleared



Mitigations (or the lack of them)

- GCC Stack Protection



```
; Attributes: bp-based frame

public strcpy
strcpy proc near

arg_0= dword ptr  8
arg_4= dword ptr  0Ch

push    ebp
mov     ebp, esp
sub     esp, 4
push    esi
push    edi
mov     esi, [ebp+arg_4]
mov     edi, [ebp+arg_0]
mov     edx, esi
mov     dl, [edx]
mov     eax, edi
mov     [eax], dl
mov     eax, edi
cmp     byte ptr [eax], 0
mov     ecx, edi
jz      short loc_3568C
```

```
loc_3567F:
inc     ecx
inc     esi
mov     edx, esi
mov     dl, [edx]
mov     [ecx], dl
cmp     byte ptr [ecx], 0
jnz     short loc_3567F
```

```
loc_3568C:
pop     edi
pop     esi
mov     esp, ebp
pop     ebp
retn
strcpy endp
```

```
.text:08076DC8
.text:08076DC8 ; int __cdecl server_task(int, char *src)
.text:08076DC8                 public server_task
.text:08076DC8 server_task     proc near               ; DATA XREF: sub_806852D+169↑o
.text:08076DC8
.text:08076DC8 var_E2C         = dword ptr -0E2Ch
.text:08076DC8 var_E28         = dword ptr -0E28h
.text:08076DC8 var_E24         = dword ptr -0E24h
.text:08076DC8 var_E20         = dword ptr -0E20h
.text:08076DC8 var_E1C         = dword ptr -0E1Ch
.text:08076DC8 var_E18         = dword ptr -0E18h
.text:08076DC8 var_E14         = dword ptr -0E14h
.text:08076DC8 var_E10         = dword ptr -0E10h
.text:08076DC8 var_E0C         = dword ptr -0E0Ch
.text:08076DC8 var_E08         = byte ptr -0E08h
.text:08076DC8 dest            = byte ptr -808h
.text:08076DC8 var_7C0         = dword ptr -7C0h
.text:08076DC8 var_7BC         = dword ptr -7BCh
.text:08076DC8 var_7B8         = byte ptr -7B8h
.text:08076DC8 var_799         = byte ptr -799h
.text:08076DC8 s1              = byte ptr -738h
.text:08076DC8 var_6B9         = byte ptr -6B9h
.text:08076DC8 var_6AC         = dword ptr -6ACh
.text:08076DC8 var_6A8         = dword ptr -6A8h
.text:08076DC8 var_2A8         = dword ptr -2A8h
.text:08076DC8 var_298         = dword ptr -298h
.text:08076DC8 optval          = dword ptr -290h
.text:08076DC8 var_290         = dword ptr -28Ch
.text:08076DC8 var_28C         = dword ptr -288h
.text:08076DC8 var_288         = dword ptr -284h
.text:08076DC8 var_284         = dword ptr -280h
.text:08076DC8 var_280         = dword ptr -27Ch
.text:08076DC8 var_27C         = dword ptr -278h
.text:08076DC8 var_278         = dword ptr -274h
.text:08076DC8 var_274         = dword ptr -270h
.text:08076DC8 var_270         = dword ptr -26Ch
.text:08076DC8 var_26C         = dword ptr -268h
.text:08076DC8 var_268         = dword ptr -264h
.text:08076DC8 n               = dword ptr -260h
.text:08076DC8 var_260         = dword ptr -25Ch
.text:08076DC8 fd              = dword ptr -258h
.text:08076DC8 var_258         = dword ptr -254h
.text:08076DC8 var_244         = dword ptr -244h
.text:08076DC8 var_240         = dword ptr -240h
.text:08076DC8 ptr             = dword ptr -23Ch
.text:08076DC8 buf             = dword ptr -238h
.text:08076DC8 s               = byte ptr -28h
.text:08076DC8 var_27          = byte ptr -27h
.text:08076DC8 var_26          = word ptr -26h
.text:08076DC8 var_24          = dword ptr -24h
.text:08076DC8 addr            = sockaddr ptr -18h
.text:08076DC8 var_8           = byte ptr -8
.text:08076DC8 var_4           = dword ptr -4
.text:08076DC8 arg_0           = dword ptr  8
.text:08076DC8 src             = dword ptr  0Ch
.text:08076DC8
.text:08076DC8                 push    ebp
.text:08076DC9                 mov     ebp, esp
.text:08076DCB                 push    edi
.text:08076DCC                 sub     esp, 0E34h
.text:08076DD2                 mov     [ebp+var_1], 0
```

```
text:0807846C                   add     esp, 10h
text:0807846F                   mov     eax, 0
text:08078474                   mov     edi, [ebp+var_4]
text:08078477                   leave
text:08078478                   retn
text:08078478 server_task       endp
```

Conclusions

- We hope that a determined, highly capable third party can safely confirm that these vulnerabilities are not exploitable due to the mitigation controls not visible to us during this analysis. We are confident owners and operators of these aircraft would welcome such independent validation and verification.
- We believe as strongly in safety as we do in security. We provide these detailed findings herein so that all stakeholders, security industry and affected entities can form their own judgment as to the exploitability and impact of these confirmed software vulnerabilities.

Thank you!