VULNERABILITY OF RADAR PROTOCOL AND PROPOSED MITIGATION

Eduardo Esteban Casanovas[1]*, Tomas Exequiel Buchaillot*[2]*, Facundo Baigorria*[3]

1 Instituto Universitario Aeronáutico, ecasanovas@iua.edu.ar
Av. General Paz 142, 1° "B", CP:5000 Córdoba, Argentina. 54-351-95-426291

2 Instituto Universitario Aeronáutico, tombuchaillot89@gmail.com
Complejo Palmas del Claret, casa 165, CP:5000 Córdoba, Argentina. 54-351-6874923

3 Instituto Universitario Aeronáutico, facubaigorria89@gmail.com
Perez del Viso 4428, CP:5009 Córdoba, Argentina. 54-351-6821829

ABSTRACT

The radar system is extremely important. Each government must ensure the safety of passengers and the efficiency of the system. This is why it has to be considered by suitable and high-performance professionals. In this paper, we have focused on the analysis of a protocol used to carry the information of the different flight parameters of an aircraft from the radar sensor to the operation center. This protocol has not developed any security mechanism which, itself, constitutes a major vulnerability. Every country in the world is going down this road, relying just on the security provided by other layer connections that could mean a step forward but definitely still not enough. Here we describe different parts of the protocol and the mitigation politics suggested to improve the security level for such an important system.

Keywords— Air traffic control, Radar, Transport protocol, Vulnerability, Mitigation

1. INTRODUCTION

ASTERIX is a standard protocol designed to exchange data between radar sensors and the control centers (ATC Systems) through means of a message structure. The protocol was designed by Eurocontrol and its acronym stands for "**A**ll Purpose **ST**ructured **E**urocontrol Su**R**veillance **I**nformation E**X**change".

ASTERIX has been developed bit by bit to provide and optimize surveillance information exchange inside and between countries (among other purposes) which makes the aerial traffic control centers (ATC) ASTERIX's main users.

Nowadays, almost every state of the ECAC (European Civil Aviation Conference) – are using it at their ATCs.

This protocol defines a standard information structure which is to be exchanged in a communication network, going from the codification of every single bit of data to the organization of the data in a data block.

These transactions can use any means of communication available like LAN networks, Internet Protocols (IP), and WAN. For this kind, data elements group up into Asterix categories. At present there exist 256 different types of categories.

The ASTERIX structure for the surveillance information exchange can be defined like this:

Data Categories

The data to be exchanged by a means of communication among different users must be standardized and classified into categories. These categories define the information that can be transmitted and encoded; in addition, its data will be standard for all of Asterix users. The purpose of this classification is to make easier the identification and the consignment of the data and also to establish a hierarchy based on their priority.

Data Item

It is the smallest unit of information in every category. For each one of them a Data Item group is determined, which constitutes the index of Data Items. Every Data Item has a unique reference that identifies it in an unmistakable way.

The symbolic reference is made up of eight characters and it is to be written in the following way: **Innn / AAA**

I stands for data item, **nnn** is a three-digit decimal number which indicates the data category it belongs to and **AAA** is a three-digit decimal number which indicates the data item number.

Data Block

It is a unit of information that contains one or more registers, each of which has the information about the same category. It is made up of:

A data octet called Category (CAT) which indicates what category the transmitted data belong to.

A 2 octet data field indicates the size of the data block (LEN).

One or more register which contain the data of the same category. Each register has a variable length but with a defined octet limit. The length will always be multiple of an octet.

The maximum size of a data block will be a mutual agreement between the data source and the users.

Field Specification (FSPEC)

The FSPEC is a content table in a bits sequence where each bit indicates the presence or absence of a determined Data Field.

There exists the possibility of using a non-standard Data Field. In order to do that, a bit is enabled and it indicates the presence of a special purpose (SP). On the other side and in this very field, another bit indicates the usage of a random organization (RFS).

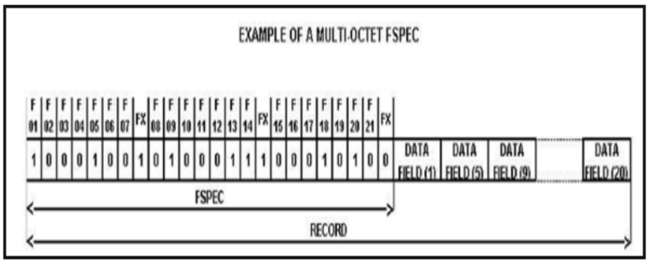
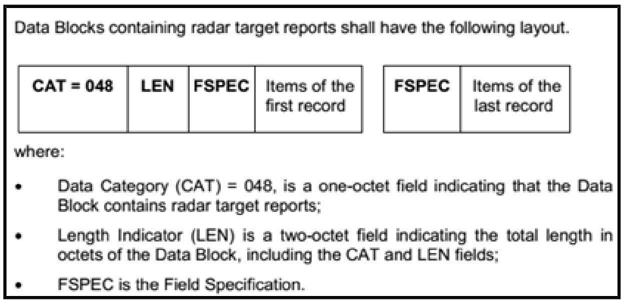


Figure 1: Example of a multi-octet FSPEC

We particularly focused on the Cat 48, a new version of the Cat 01 and Cat 16 SSR Mode-S since now is the most used for the civil aviation in our country.

Asterix CAT 48 is a category where the information about target radars goes from a header to a radar data-process system. In this category plot of tracks, messages can be transmitted by a combination of the two.

In the following table, you can see standard data items of the category.



Data Blocks containing radar target reports shall have the following layout.

CAT = 048	LEN	FSPEC	Items of the first record	FSPEC	Items of the last record

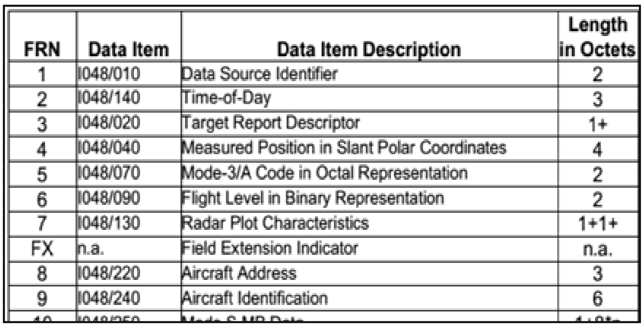
where:

- Data Category (CAT) = 048, is a one-octet field indicating that the Data Block contains radar target reports;
- Length Indicator (LEN) is a two-octet field indicating the total length in octets of the Data Block, including the CAT and LEN fields;
- FSPEC is the Field Specification.

Figure 2: Category 48 Data Block

As an example, in the following table it is shown the standard UAP (User Application Profile) for the information about tracks of category 48.

Table 1: example of standard UAP for the track information



FRN	Data Item	Data Item Description	Length in Octets
1	I048/010	Data Source Identifier	2
2	I048/140	Time-of-Day	3
3	I048/020	Target Report Descriptor	1+
4	I048/040	Measured Position in Slant Polar Coordinates	4
5	I048/070	Mode-3/A Code in Octal Representation	2
6	I048/090	Flight Level in Binary Representation	2
7	I048/130	Radar Plot Characteristics	1+1+
FX	n.a.	Field Extension Indicator	n.a.
8	I048/220	Aircraft Address	3
9	I048/240	Aircraft Identification	6

As shown in the table, each field has an integer amount of octets where the information is represented.

In order to conclude the ASTERIX matter, we will highlight one of the main protocol's issues: The lack of security.

The protocol does not include any security system in it, meaning that it does not have a corroboration of the information that is transmitted in the communication.

Since it is not able to assure the integrity or the authenticity of the information, ASTERIX turns out to be a protocol vulnerable to many different types of malicious attacks, as for instance, a Man in the Middle attack.

New Technologies

In these days, several countries are trying out a new technology surveillance in commercial aviation, know as ``ADS-B´´ (Automatic dependent surveillance-broadcast)

It is included in the US Next Generation Air Transportation System (NextGen) and the Single European Sky ATM Research (SESAR).

This cooperative technology is used in the aircraft setting its position through satellite navigation and transmitting it regularly so that it is possible to be tracked down. This information can be read not only by air-traffic control stations but also other aircrafts implementing such technology. The objective of ADS-B is to replace the secondary radars.

Even though this technology replaces the current system of transposition, its low level of communication remains in ASTERIX.

Using other ASTERIX categories like the Cat 21 will be the only difference. Therefore, the objective of this work will still be valid since all needed onwards is to encode / decode a new ASTERIX category for every program to work perfectly well.

3. MITM ATTACK

Man in the middle is a type of attack in which the attacker has the ability to read, insert and modify the messages that are being sent between the two hosts without either of them knowing that the link has been violated. Once the data link has been compromised, the attacker has the capacity of sniffing and intercepting the messages that are exchanged between the victims.

Below are some of the most common techniques to commit an MITM attack.

ARP Poisoning o ARP Spoofing: Is an MITM type of attack for Ethernet networks which allows the attacker to capture the network traffic exchanged over the LAN network, stopping it and also being able to deny it.

DNS Spoofing: This type, uses fake responses to the DNS resolution requests sent by a victim. There are two methods that the attacker may use: "ID Spoofing" and "Cache poisoning".

Port Stealing: here, the attacker sends a large amount of Ethernet frames (OSI Model Layer 2 packets), with the MAC address of the victim as source and with the attacker's own MAC address as the target. This switch makes the victim believe to be connected at the attacker's port.

DHCP Spoofing: The DCHP requirements are made up of broadcast frames due to the fact that these must be heard by all devices within the local network. If an attacker answers the request before the server does, the former may send the wrong information to the victim.

In this case, we use the **ARP Poisoning** method. This technique is used in local networks aimed to acquire network traffic destined for another host. Using this method allows us to redirect the data intended for the original host to our own network card and by doing this we are able to block, modify or even add new data.

This technique is not based on a particular vulnerability that may disappear over time, but on a TPC network design bug. That is why this kind of attack shall remain in force unless new specific security measures are taken.

4. SIMULATION

In order to do the testing in a controlled environment, there have been conducted simulations of a communication between a plane and an airport's control turret. To recreate each of the elements, custom software was coded. This software was located in different virtual machines which, as a unit, simulate the airport communication infrastructure. The communication method we have chosen to use was UDP sockets.

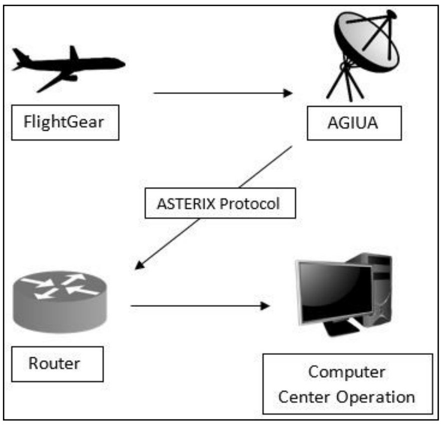


Figure 3: Network Simulation

The simulations are described in the following items:

Airplane Transponder

It is responsible for generating and sending the flight data used by the radar tower. Through Asterix packets, this information is transmitted to the operation center. In order to get more accurate data, we used the simulator called "FlightGear".

FlightGear

It is a multiplatform open-sourced flight simulator. We used this software getting after the objective of obtaining real-time plane data, generated virtually using the software's GUI (graphical user interface).

Radar

It is the responsible of receiving the raw data from the planes, using it for the creation of ASTERIX packets and sending them through a UDP socket to the network. So as to do this, we created a software called AGIUA (Asterix Generator IUA), fully developed in C++.

AGIUA

It takes the raw data from a predefined port, analyzes and makes the calculations to transform the information from the plane, into ``Data Items´´ of a category ASTERIX to be sent. After the length of the resulting fields is defined, they can be inserted in the corresponding ASTERIX´s headboard FSPEC in order to have the complete package that is going to be transmitted by another UPD socket to the next node. As for now, only AGIUA creates category 48 and category 34 ASTERIX packets most of which are the being used in commercial aircraft.

Router

This router/firewall is responsible for redirecting the ASTERIX´s packets to the operation center node, and drop another packet. We do this through a script made with iptables which have the following attributes:

DROP by default all the packets.

Redirect all the packets that are sent from the radar node IP, from a specified UDP port, and which protocol is ASTERIX to a specific port from the operation center node. In order to do that, we created NAT, PRE-FORWARDING and POST-FORWARDING rules.

We also took the security measures needed to achieve this simulation as close to the reality of the airport routers as possible, for instance: port blocks or an update of different services to avoid known vulnerabilities.

Operation Center

The operation center is the responsible of receiving the ASTERIX packets sent from the radar and at the same time, of decoding their data and distributing the packets to the different stakeholders. For example, send the location data to the control tower so the ATC can manage the air traffic. To simulate this system we made a C++ software, which is in charge of making those decoding. It also has a GUI (graphical user interface) in which is represented the location of the aircraft in order to visualize all the different tests that we made for the project.

To achieve this, it puts all the data in a queue where it will consider whether the ASTERIX and FSPEC headboard matches the rest of the saved package. If it is positive, it will take FSPEC byte by byte and shall be taking elements from the queue (which would come to form the ASTERIX DATA ITEMS package) and analyzing information sent for. At the same time, in another thread, the program will correctly be formed by plotting the packages taking its Aircraft Address and coordinates.

5. MITM APPLIED TO ASTERIX

In this section, we will explain how we applied this type of attack to manipulate the ASTERIX protocol according to our aims.

Basically, all the packets that are going to travel on this network have the same structure: header – packet body. In the header, we can find a different type of elements such as the source IP, target IP, packet length, checksum, etc. The packet body contains ASTERIX blocks (each one with its specific category) and own registers of each block specifying the flight data.

Having understood these concepts, we can approach the custom software coded for this section: MITMAST (Man in The Middle ASTerix). The main objective of this software is to capture all the packets between two nodes (in our case, the ASTERIX packet generator and the operation center) and manipulate them. It is a simple software, developed in C, that launches an MITM attack using the ARP Poison technique between two hosts. To do this, the software uses osdep, a tunnel creation library which is part of the air crack project. With this, we can create an interface (mitm0) in which the response packets will be written in order to be easier to sniff.

MITMAST will receive the following parameters:

mitmast -i interface **-t** ip1 ip2 **-o** option

- **-i**: It specifies the network interface to be used which will get in the promiscuous mode to sniff the network.
- **-t**: It specifies the victims' host IP network.
- **-o**: Using this option, we specify one of three options to determine the attack to make: BLOCK, MOD or ADD.
  - **BLOCK** – Delete an aircraft: Once the ASTERIX packets have been obtained, the software will recognize the packets that belong to a particular aircraft and will not forward them to the operation center, by doing so, the aircraft is deleted from the operation center data.
  - **MOD** – Modify the track of aircraft: Once the ASTERIX packets have been obtained, the software will recognize the packets belonging to a particular aircraft and by using an algorithm it will pretend a detour to make it look like the aircraft has changed its original route.
  - **ADD** – Insert a ghost aircraft: The software will generate new ASTERIX packets with reliable data and will send them to the operation center. This will make it look like the radar is receiving an aircraft that does not actually exist. This packet injection can be made with many aircrafts at the same time.

Once all the parameters are determined, it will request some information depending on the options we specified before:

- **BLOCK**: It will request the Aircraft Address. This is an element contained in each CAT 48 ASTERIX packet and it identifies unequivocally the aircraft.
- **MOD**: It will request the Aircraft Address and the modification TYPE. This last option can be: MANUAL or SIMULATED. If it is manual, it will ask the aircraft's final coordinates and the simulator will automatically make a parable until it arrives at the specified point and then, the aircraft will disappear from the radar, becoming blocked. If we choose the simulated option, within five seconds another instance of FlightGear will take over the flight coordinates, using them on a specified port.
- **ADD**: It will request the Aircraft Address, CANT and DIST. The aircraft address is asked in order to identify the aircraft, the CANT option is requested to specify the number of ghost planes to be added and the DIST option is requested to specify the distance between the aircrafts.

Having finished this stage, the software will make an ARP Poison attack to the specified hosts, misdirecting the packets to the attacker host. If this attack is successful we should be able to see in our screen confirmation message and the software will begin to transform the packets. It is important to point out that before sending them, their header is modified, changing the source/target of the packet and making a new checksum leave no trace of our intrusion.

6. EXPAMPLES

In the following images, we will demonstrate how the attack works.

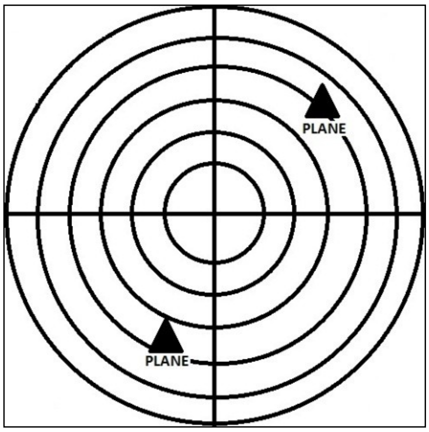


Figure 4: Normal Radar

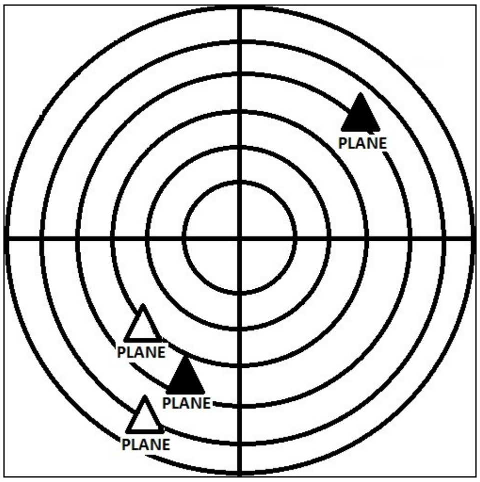


Figure 5: Radar in ADD Attack

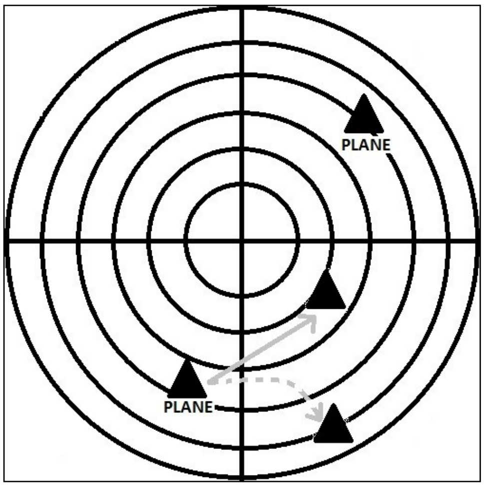


Figure 6: Radar in MOD Attack

7. MITIGATION

An action that can make the attacker to the network is to perform a listening on traffic established between radar and the operations center and save it. This will allow the attacker to subsequently perform a valid format inkjet packages and features but will not be valid in time. This indicates that within the mechanism proposed, we must remember that the attacker may be interested in making injection valid packets (Replay attack). Additionally, the attacker can select the stored traffic and perform a selective injection.

The main point is that Asterix packages have not implemented any security mechanism. This means that security mechanisms should be done outside Asterix. The problem is that if the attacker can pass through these security barriers, he will find all packages in plain text and can perform attacks Block, Mod Add.

Many electronic countermeasures are used in radar. They all seek to mitigate attacks that are made on the sensor, but once the signal is validated, this signal goes into a network protocol that has no additional security beyond the one that can be provided at network level.

As we all know, the security at the network level is continuously broken, you just see what happened with SSL-TLS during the last years, therefore, put our security in this protocol it is not enough.

Our first problem is to ensure the integrity of Asterix package. Although, in reality, we shall see that for the moment we will only guarantee the integrity of information of certain flight parameters. Listed in Table 1: standard UAP for the track information, we are going to focus on the FRN:

2	Day time,
8	Aircraft`s Address,
9	Aircraft`s identification,
11	Track number,
12	Position velocity calculate,
13	Track calculate,
14	Track`s status,

We will focus on these fields because it's on them that we have raised our attack. However, this does not mean that we cannot guarantee the integrity of any other field.

The mechanism to ensure the integrity of these fields is the use as a hash function. Because of the replay attack, we will add a field "time stamp". This field is added to the aforementioned and all of them will do the hash calculation. Due to our network characteristics, we can say that among of different components of the network we can have a pre-shared secret, this is going to allow the use of other cryptographic functions such as the HMAC. The extra advantage in the use of such functions is that we can authenticate the sensor that is receiving the information.

Processing Time

A very important point in our analysis is if the processing time in the incorporation of these security measures compromises the normal flow of packet reception.

To verify this evidence, we apply this security method on flows with different types of frequency. Even in the worst situation, that is a scenario of maximum traffic, there can be processed more than 30 packets per second per sensor, and no bad effect appears, so no degradation in the flow of the packets was performed.

These measurements make us think about how to develop an additional security features.

Package encryption

While most importantly for this scheme is to ensure the integrity of the packages, an additional feature is to have confidentiality on the information we send. That is why we also propose additional security features as it is to perform encryption on the same fields on which it will ensure integrity.

To verify that it is possible to perform, we applied on the aforementioned fields, an encryption algorithm. Here we use AES-CBC. Other encryption mechanisms can be used, such as AEAD (Authenticated Encryption with Associated Data). This mechanism is very attractive because it provides confidentiality, integrity, and authenticity.

Once we complete the encryption process we replaced in the selected field, the plain text information with the cipher text.

And finally making a combination of both mechanisms, the HMAC function or just the typical hash function is applied. This allowed us to have guaranteed the integrity of the previously encrypted fields.

Final tests

With the two mechanisms (integrity and confidentiality) in place, we perform several tests in order to analyze the impact of the application of the two security features.

Here also taken into account the characteristics of the type of traffic we have between the sensor and the operation center.

Two different situations were studied. Low traffic operation can have 3 packets flow per second per sensor and in a high traffic operation we have approximately 10 packets per second per sensor.

During the complete operation, we can process 18 packets per second per sensor without any kind of delay in the normal flow. Therefore, the incorporation of the encryption method in the required fields can do without compromising the normal flow of traffic.

8. CONCLUSION

As a part of the critical infrastructure of a country, the radar system is fundamental in the air transport system. That is why we must make every effort to ensure the maximum availability and security. Asterix protocol designed by Eurocontrol is very efficient but lack of an adequate safety mechanism itself. That is why you should have to move to another link layer to obtain a security status, which seems insufficient, considering the criticality of the information handled.

The proposed mitigation presented in this paper covers possibilities described attacks but also provides an additional level of security thinking of an attack from inside of the organization.

Our proposed mitigation against vulnerability raised sharply covers actions that can be performed by an attacker who has enough information to be able to listen the communication channel between the radar sensor and the operation center, because it will not be able to manipulate any packages. Also, during a situation while implementing encryption of packages, the attacker cannot display the flight parameters that are being transmitted.

Last but not least, we highlight at this conclusion the importance of processing times involved in the cryptographic mechanism, to ensure the protocol`s integrity and confidentiality, saying we have been highly satisfied because there are no limits with the data flow required to be transmitted at all times.

REFERENCES

[1] Adolf Mathias, Matthias Heß (2012). “Machine-Readable Encoding Standard Specifications in ATC”.IEEE - Digital Communications - Enhanced Surveillance of Aircraft and Vehicles (TIWDC/ESAV), 2011 Tyrrhenian International Workshop.

[2] Bruce Schneier. Schneier on Security. https://www.schneier.com.

[3] Craig Hunt (1997). “TCP/IP Network Administration”. O'Reilly & Associates.

[4] D. Brent Chapman & Elizabeth D. Zwicky (1995). “Building Internet Firewalls”. O'Reilly & Associates.

[5] DEFCON. DEFCON Conferences. https://www.youtube.com/channel/UC6Om9kAkl32dWlDSNlDS9Iw.

[6] EUROCONTROL-European Organization for the Safety of Air Navigation. Asterix protocol. https://www.eurocontrol.int/asterix.

[7] Ministerio de Defensa de España. Ciberseguridad: Retos y amenazas a la seguridad Nacional en el Ciberespacio. http://bibliotecavirtualdefensa.es/BVMDefensa/i18n/catalogo_imagenes/grupo.cmd?path=17029.

[8] Milw0rm Hacker Group. W4rri0r.http://www.w4rri0r.com/.

[9] Naga RohitSamineni, Ferdous A, Barbhuiya and Sukumar Nandi (2012).“Stealth and Semi-Stealth MITM Attacks, Detection and Defense in IPv4 Networks”. IEEE - Parallel Distributed and Grid Computing (PDGC), 2012 2nd IEEE International Conference.

[10] RenderMan. RenderLab.http://renderlab.net/.

[11] Zhe Chen, ShizeGuo, KangfengZheng and Yixian Yang (2007). “Modeling of Man-in-the-Middle Attack in theWireless Networks”.IEEE -Wireless Communications, Networking and Mobile Computing, 2007.WiCom2007. International Conference.

[12] ADS-B Technologies Website. http://www.ads-b.com/

[13] ADS-B and Asterix application for Eda. http://era.aero/technology/ads-b-2/