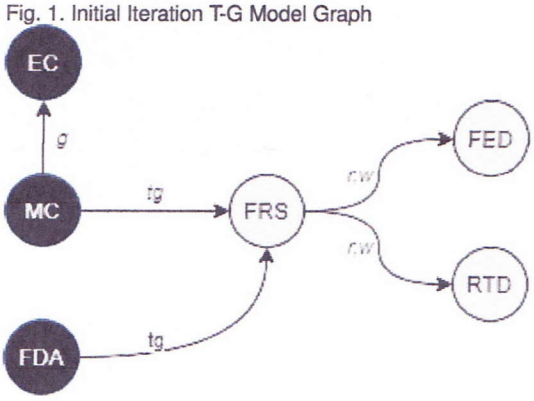
Fig. 1. Initial Iteration T-G Model Graph



2 Survey

2.1 Take-Grant

Next, we take a look at the Take-Grant (TG) model interpretation of our rights leakages. This model represents our system as a directed graph, with our subject and objects represented as vertices (black for subjects, white for objects). Edges represent the rights on vertex has over another, where special take (*t*) and grant (*g*) representing the ability of a subject or object to give or obtain rights from another subject or object. In our case, subjects can execute a grant HRU command representing the grant right, and take can be achieved by rights automatically allocated by granting access to specific objects (i.e., the FRS). This models our HRU example in terms of a TG model where the *Flight Record* created object has *read, write* rights over the *Flight Record System* that our **External Contractor (EC)** can utilize to gain *read, write* access to the *Flight Engine Data*.

Our bad actor could achieve this leaked through the following TG commands.

1) **MC** creates object **FR**.
2) **MC** grants (*t* to **FR**) to **EC**.
3) **EC** takes (*r,w* to **FRS**) from **FR**.
4) **EC** takes (*r,w* to **FED**) from **FRS**.

Figure 1 shows the initial state of rights among actors and objects in the **FRS** system.

Figure 2 shows the state of rights after the **MC** runs the HRU command *create_flight_record(MC)* which a new **FR** object.

Figure 3 shows the state after the **EC** leads the **MC** into giving them *t* rights over **FR** they can exploit the system to take our leaked rights to **FED** through the **FRS**. This would be established after the command *grant_flight_record_access(MC, EC, FR)*.

Therefore allowing EC to execute *update_flight_record_system(EC, FR)* which leaks access to the FED as shown by the state in Figure 4.

2.2 Bell-LaPadula

The Bell-LaPadula Model (BLP) focuses on establishing access control confidentiality through security levels and categories. BLP focuses on two simple rules in a security policy within a given ACM. The simple security property states a subject can not read an object at a higher security level. The star property states that a subject may not write to any object at a lower security level.

Fig. 2. Iteration 2 T-G Model Graph

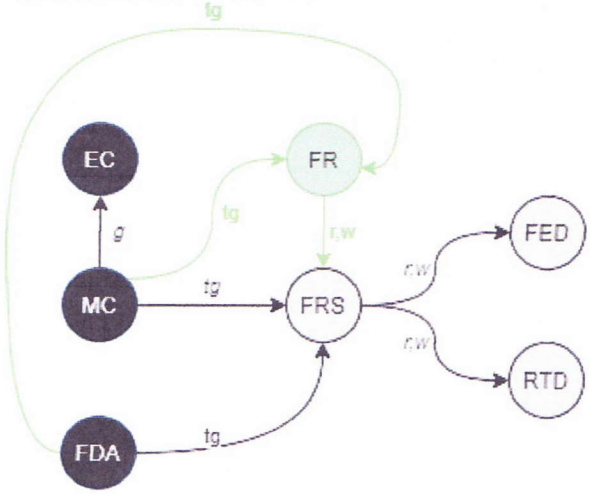
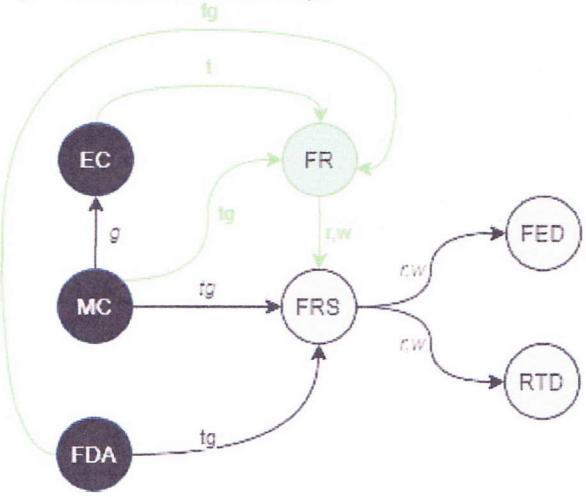


Fig. 3. Iteration 3 T-G Model Graph



The primary limitation we see when implementing a BLP model into our existing policy is our external contractor and maintenance controllers. We need access to the flight records, so they require at least *confidential* clearance levels. These restrictions make it hard to read the flight record system at a *secrete* level and our FRS to write to the flight records.

For our BLP implementation, we first define the security classifications for our subjects and objects. Given we are dealing with military records, it makes sense to use the standard government security levels. In addition to the security level, subjects and objects will fall into categories based upon the appropriate work unit. Table 5 shows the security clearance levels and the subjects/objects at the given levels. Table 6 shows the work unit categories of each subject/object.

Now that we've established security classifications for our subjects and objects, we can see how the system and security policy could still allow leakage and further theft of flight record data.

Fig. 4. Final Iteration T-G Model Graph

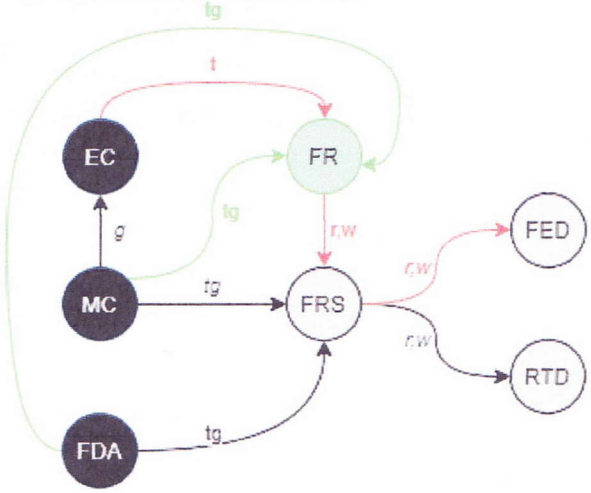


TABLE 5
Security Classifications for the UAS

Clearance Level	Subjects	Objects
Top Secret (TS)	PC, FDA	ANC, RO
Secret (S)	IP	FRS, RTD
Confidential (C)	MC, EC	FED, FR
Unclassified (UC)		

```
create_flight_record(Maintenance Crew
    (MC));
grant_flight_record_access(Maintenance
    Crew, External Contractor (EC),
    Flight Record (FR));
upload_flight_record(External Contractor
    (EC), Flight Record (FR));
make_owner(MC, EC, FR);
update_flight_record_system(EC, FR);
```

1) **MC** executes *create flight record* creating flight record object **FR**.
   - This write is allowed as the object **FR** created has clearance level of *Confidential* which is greater than or equal to **MC's** level of *Confidential* and **FR** dominates **MC** by categories.
2) **MC** executes *upload flight record* granting the object **FRS** access to the flight record object **FR**.
   - This write is allowed as the object **FRS** created has clearance level of *Secret* which is greater than **FR's** level of *Confidential* and **FRS** dominates **MC** by categories.
3) **MC** executes *grant flight record access* granting **EC** access flight record object **FR**.
   - This read is allowed as the subject **EC** created has clearance level of *Confidential* which is greater than or equal to **FR's** level of *Confidential* and **MC** dominates **EC** by categories.
4) **MC** executes *make owner* making **EC** have owner rights over flight record object **FR**.
   - This write is allowed as the subject **EC** created has clearance level of *Confidential* which is greater than or equal to **FR's** level of *Confidential* and **FR** dominates **EC** by categories.
5) **EC** executes *update flight record system* allowing **EC** write to the flight record system object **FRS**.
   - This write is allowed as the subject **EC** created has clearance level of *Confidential* which is less than or equal to **FRS's** level of *Secret* and **FRS** dominates **EC** by categories.

As you can see, while this system does keep our bad actor **EC** from reading the confidential flight records from the flight record system, it does not prevent an integrity attack of writing up to the **FRS** and **FR**.

2.3 Biba

The Biba model focuses on data integrity as opposed to the BLP model, which enforces confidentiality by restricting access. The Biba model ensures the data objects once created remain unmodified by untrusted sources by establishing integrity levels. The model achieves this by ensuring ACM transitions do not allow subjects to write/modify data above their integrity level. Subjects do not read data below their level to avoid being influenced by data below their trusted level.

These notions apply to our infrastructure well as the bad actor (External Contractor), in theory, could either read data they're not supposed to (violating confidentiality). Or the EC could modify existing flight record data (violating integrity). If the confidentiality policy is perfectly implemented, no data is accessed that is outside their security level. However, there's still an amount of "trust" we have in the EC and system preventing the integrity of the data from being violated. The Biba model attempts to quantify this level of trust and maintain the flow of information, inhibiting its modification by lower trusted subjects/objects.

2.4 BLP-Biba Lipner Like

Lipner proposed an integrity model that combined aspects of the BLP and Biba model to fit a more real-world commercial software development environment. Lipner created security levels and categories similar to BLP to prevent various software development lifecycle subjects from accessing different systems. Lipner then added to the security classifications with integrity classifications for system programs and integrity categories to differentiate development and production environments.

TABLE 6
Work Unit Categories for the UAS

Category	Subjects	Objects
Flight Ops (FO)	PC,IP	ANC,RO
Maintenance Ops (MO)	PC,IP,MC,FDA	FED,RTD,FR,FRS
Flight Data Ops (FDO)	PC,IP,FDA,EC	FED,FR,FRS

TABLE 7
Security Categories for the UAS

Security Category	Subjects	Objects
Archived Data (AD)	PC,MC,FDA	FRS
Operational Data (OD)	PC,IP,MC	ANC,RO,FED,RTD
Ready To Load (RTL)	FDA,MC,EC	FR,FRS

TABLE 8
Integrity Categories for the UAS

Integrity Category	Subjects	Objects
On Aircraft (IOA)	PC,IP,MC	ANC,RO,FED,RTD
Flight Record System (IFRS)	FDA,MC,EV	FRS

In our BLP model we established various security classifications (Table 5) and work unit categories (Table 6) so we need to establish security categories to define the area of impact in the given category (Table 7).

We can now incorporate integrity categories to allow distinction between on aircraft systems and the flight record keeping system (Table 8) and integrity classifications to determine the trust of the data combing from that source (Table 9 highest to lowest level).

This gives us an overall security and integrity clearance levels for subject categories (Table 10) and objects (Table 11).

REFERENCES

[1] A. Erwin, and J. Gibson, Navy, Boeing Make Aviation History with MQ-25 Becoming the First Unmanned Aircraft to Refuel Another Aircraft, Accessed on: Sept. 1, 2021. [Online]. Available: https://www.boeing.com/defense/mq25/

Matthew Whitesides Master's Student at Missouri University of Science and Technology.

TABLE 9
Integrity Classifications for the UAS

Integrity Classifications	Subjects	Objects
System Programs (ISP)	∅	ANC,RO,FRS
Operational (IO)	PC,IP,FDA	∅
Maintenance (IM)	MC,EC	FR,FED,RTD,FR

TABLE 10
Security and Integrity Levels for Work Units

Work Unit Category	Security	Integrity
Flight Ops	({TS,{OD})	(ISP,{IOA})
Maintenance Ops	({S,{AD,RTL})	(IO,{IOA,IFRS})
Flight Data Ops	({S,{OD,RTL})	(IM,{IFRS})

TABLE 11
Security and Integrity Levels for Objects

Object	Security	Integrity
ANC	({TS,{OD})	(ISP,{IOA})
RO	({TS,{OD})	(ISP,{IOA})
FED	({S,{AD,OD,RTL})	(IO,{IOA,IFRS})
RTD	({S,{OD})	(IO,{IOA,IFRS})
FRS	({S,{AD,OD,RTL})	(IM,{IFRS})

Does this block any rights leakages?