3 Challenge security 2 - Venerba Mirco 872653 – Arbac parser

In this challenge we must develop a simple script, i wrote it in Python, to analyze some input arbac policies (.arbac files) and then verify if a role is reachable or not.

The script is divided into three part, the first is the initial part where the script takes in input the arbac file, the second one is the part where using backward slicing we reduce the number of possible states and the last one is the part where the script analyzes the input policy and it checks if the final state is reachable or not.

Then we must find the flag that is the sequence of the bites where each of them indicates that the i-th file has the final state reachable.

The final flag is 10110110.

Usefull definitions used for the development

Can assign rule:

(ra, Rp, Rn,rt) ∈ CA has the following meaning: users with administrative role ra can assign role rt to any user who has all the roles in Rp and none of the roles in Rn

Can revoke rule:

(ra,rt) ∈ CR has the following meaning: users with administrative role ra can revoke role rt from any user.

Role assignment / revocation performed by user ua with role ra on user ut

$$\frac{(u_a, r_a) \in UR \qquad (r_a, R_p, R_n, r_t) \in CA \qquad R_p \subseteq UR(u_t) \qquad R_n \cap UR(u_t) = \emptyset \qquad r_t \notin UR(u_t)}{UR \rightarrow_{\mathcal{P}} UR \cup \{(u_t, r_t)\}}$$

$$\frac{(u_a, r_a) \in UR \qquad (r_a, r_t) \in CR \qquad r_t \in UR(u_t)}{UR \rightarrow_{\mathcal{P}} UR \setminus \{(u_t, r_t)\}}$$

Backward slicing pseudocode:

S0 = {rg}

Si = Si−1 ∪ {Rp ∪ Rn ∪ {ra} | (ra, Rp, Rn,rt) ∈ CA ∧ rt ∈ Si−1}

Let S∗ be the fixed point to this set of equations, then:

1 remove from CA all the rules that assign a role in R \ S∗

2 remove from CR all the rules that revoke a role in R \ S∗

3 delete the roles in R \ S∗

Verify reachability pseudocode:

for each arbac file

    state to visit = [user roles]

    state already visited = [ ]

set actual flag equal to false

apply backward slicing algorithm

while actual flag is false

for each set of states to visit

create a list of user-role from the list of all users that can get a particular role using the list of can assign rules

update the two lists “state to visit” and “state already visited” appending a new element composed by the actual set of states plus the list calculated in the previous points

create a list of user-role from the list of all users that have a particular role using the list of all can revoke rules

update the two lists “state to visit” and “state already visited” appending a new element composed by the actual set of states to visit minus the list calculated in the previous point

if at least one user has the goal role

append one bit equal to one to the string flag

set actual flag to true

break the execution and start to analyze next file

if the timer is finished

append one bit equal to zero to the string flag

set actual flag to true

break the execution and start to analyze next file

print the flag string to obtain the result

Implementation thing:

I used a time of 5 seconds to execute each arbac file because it is sufficient for the file with goal state reachability and the cases where the state it isn’t reachable, after 5 seconds the program terminates and adds zero character to the final string.

Important things:

The computational complexity of the role reachability problem is PSPACE-complete and for this reason we can implement different techniques as: use restricted fragments of the ARBAC model, rely on approximated analysis techniques (false positives), perform an aggressive pruning of the ARBAC policy and develop backward and forward slicing algorithms (in this script i implement only the backward slicing because to solve the task is sufficient to do only this and also the development of all two slicing algorithms is in suggestions section and so it isn’t mandatory to implement also the forward slicing).