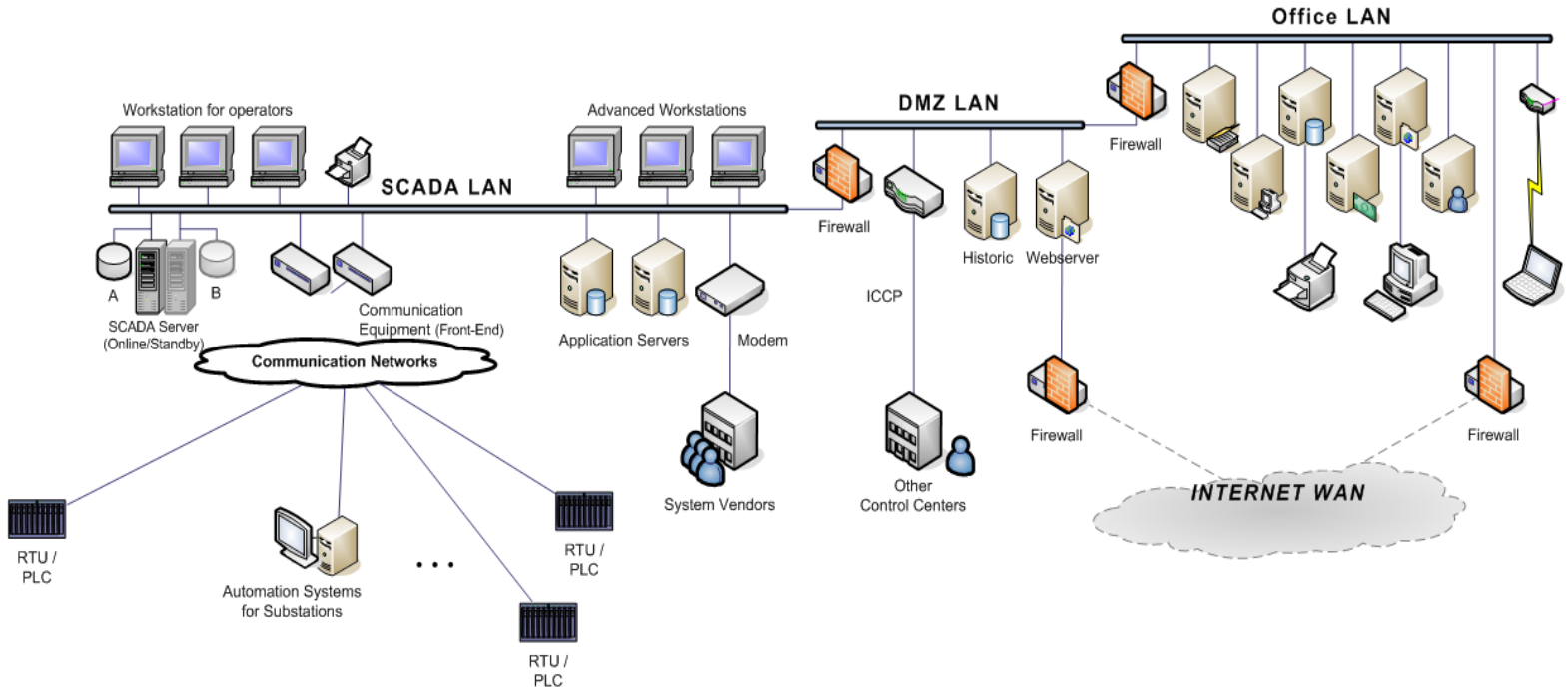
Modeling and Simulating with Adversary-driven System Architecture Dynamics

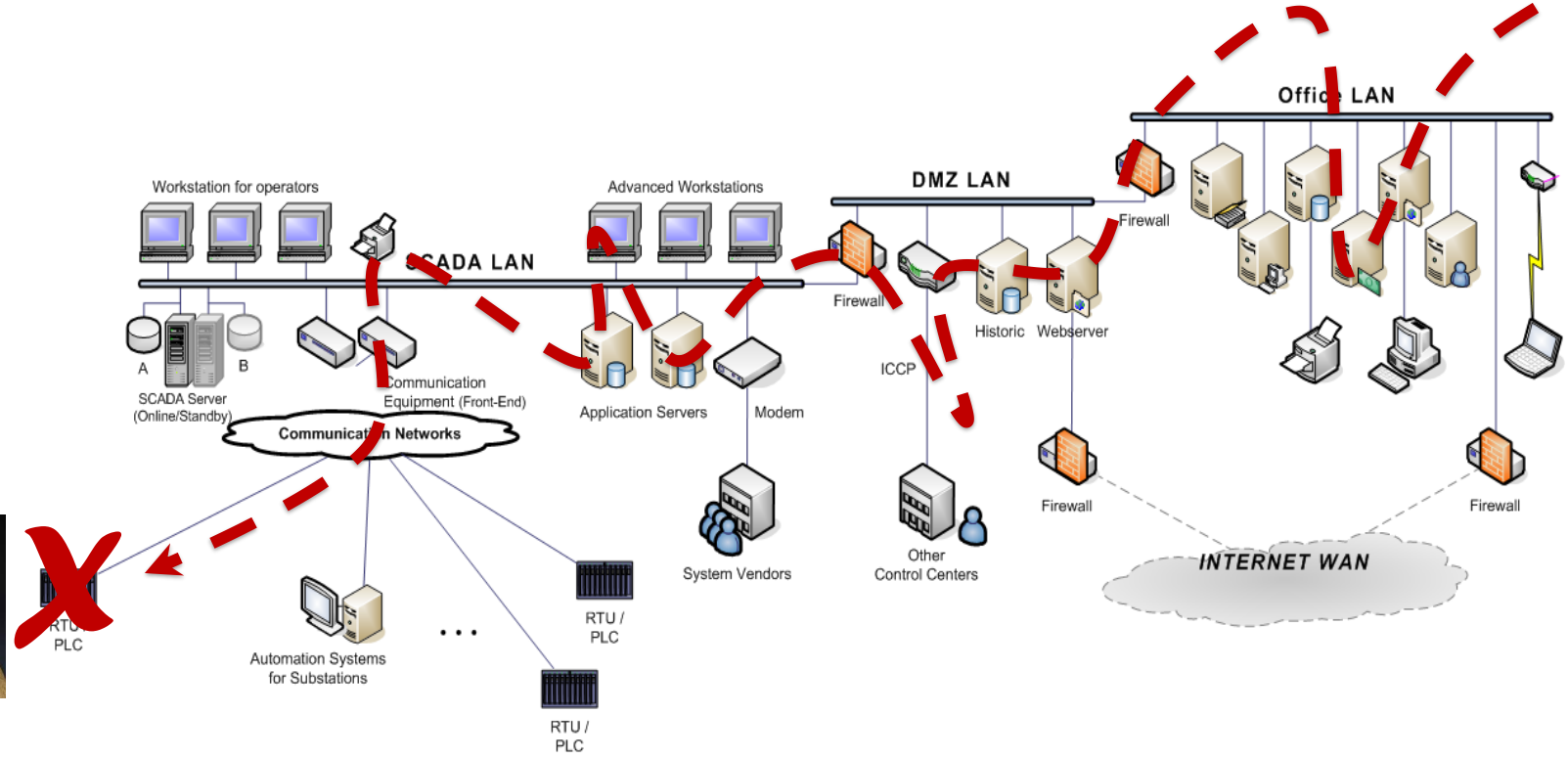
Viktor Engström, Giuseppe Nebbione, Mathias Ekstedt

CDIS Spring Conference 2025

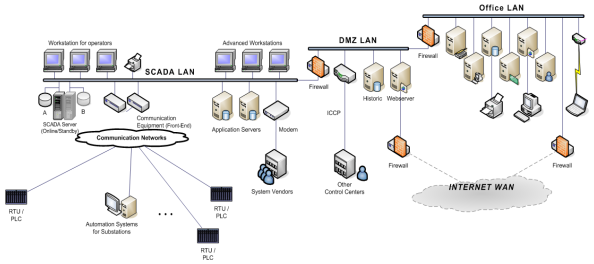
Is my architecture secure (enough)?



How do you get from A to B?

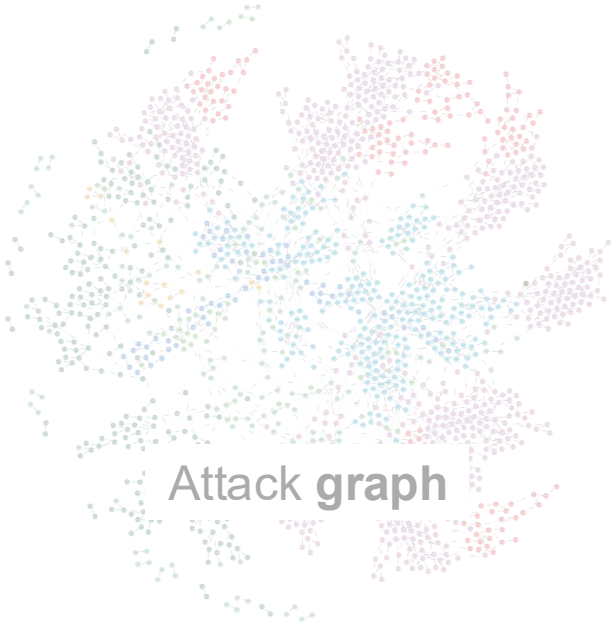


Attack graph construction



Real architecture

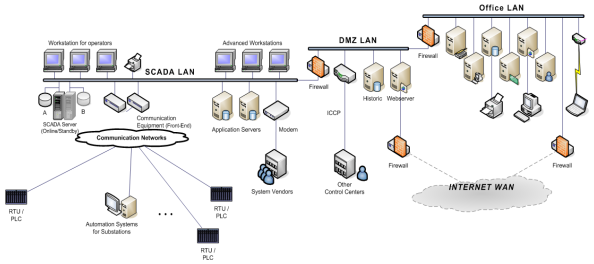
System **model**



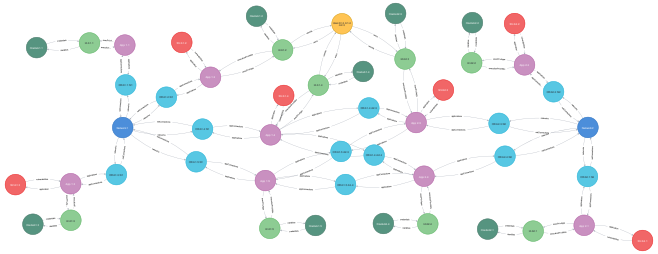
Attack **graph**

Meta Attack Language
MAL

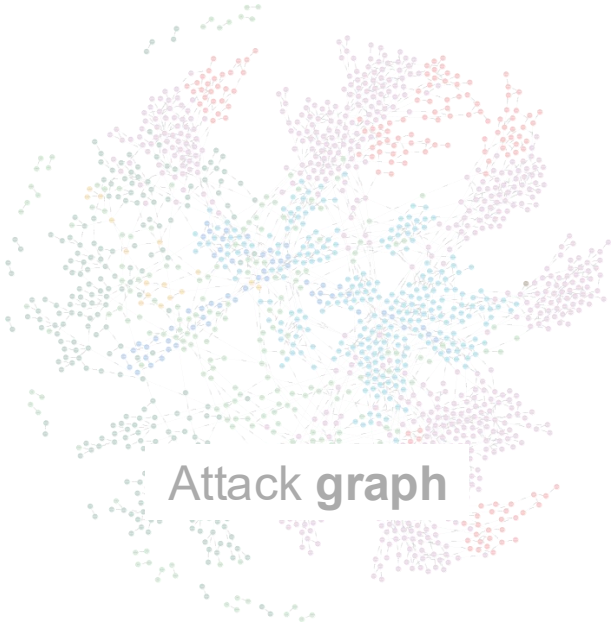
Attack graph construction



Real architecture



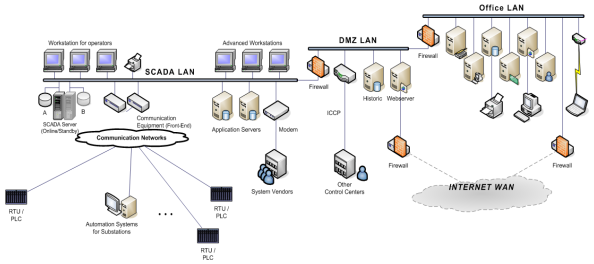
System **model**



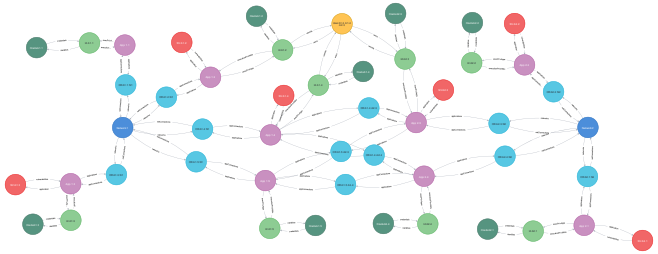
Attack **graph**

Meta Attack Language
MAL

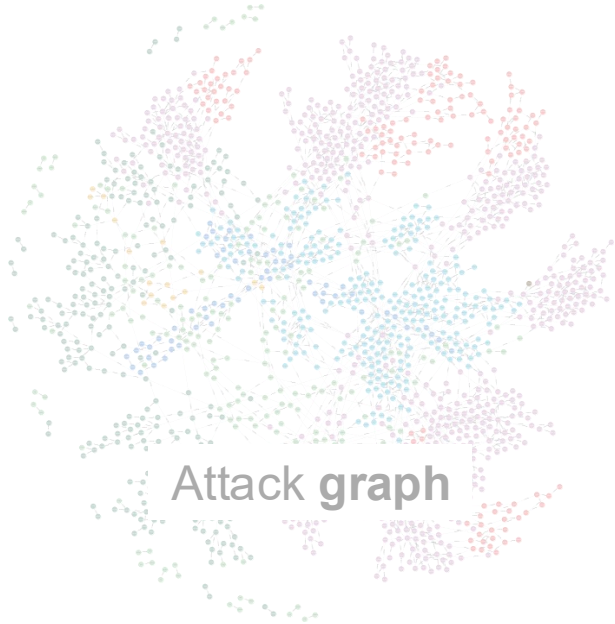
Attack graph construction



Real architecture



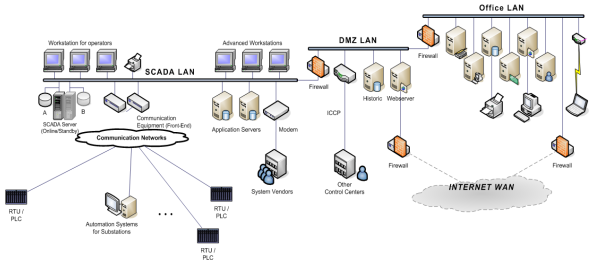
System **model**



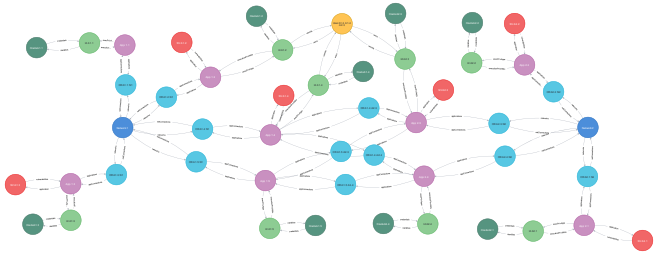
Attack **graph**

Meta Attack Language
MAL

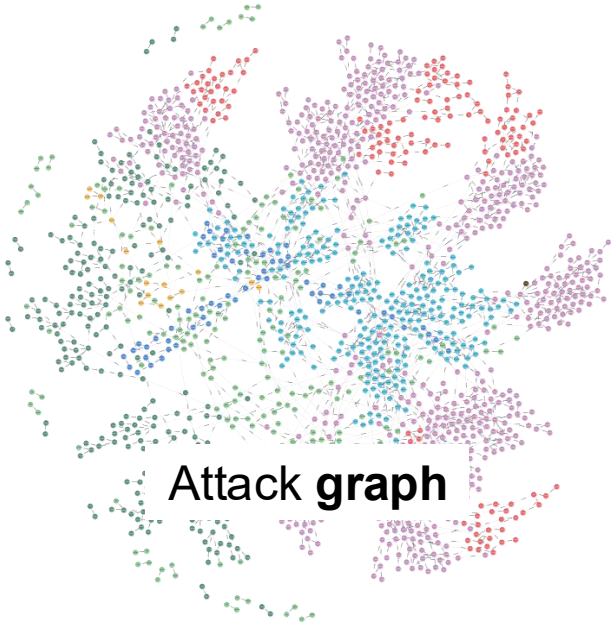
Attack graph construction



Real architecture



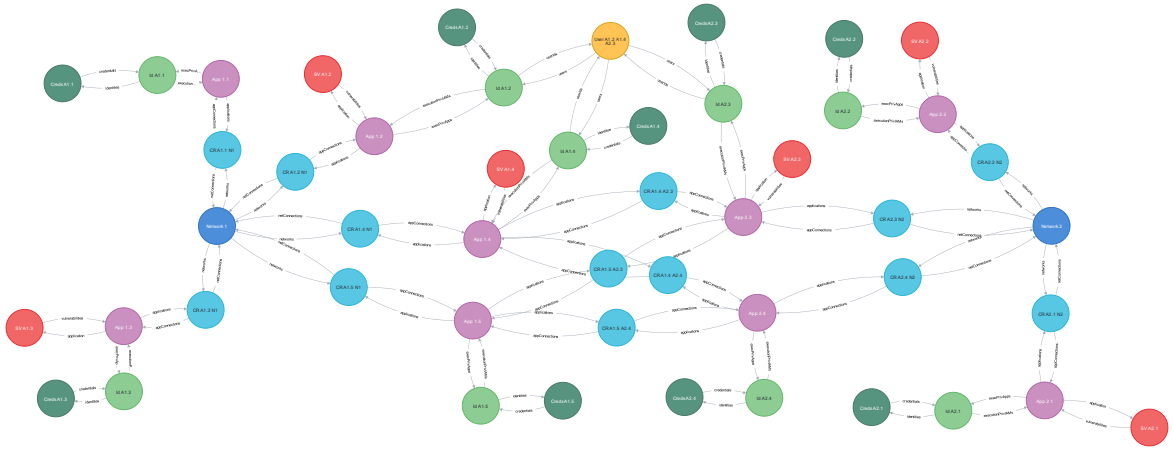
System **model**



Attack **graph**

Meta Attack Language
MAL

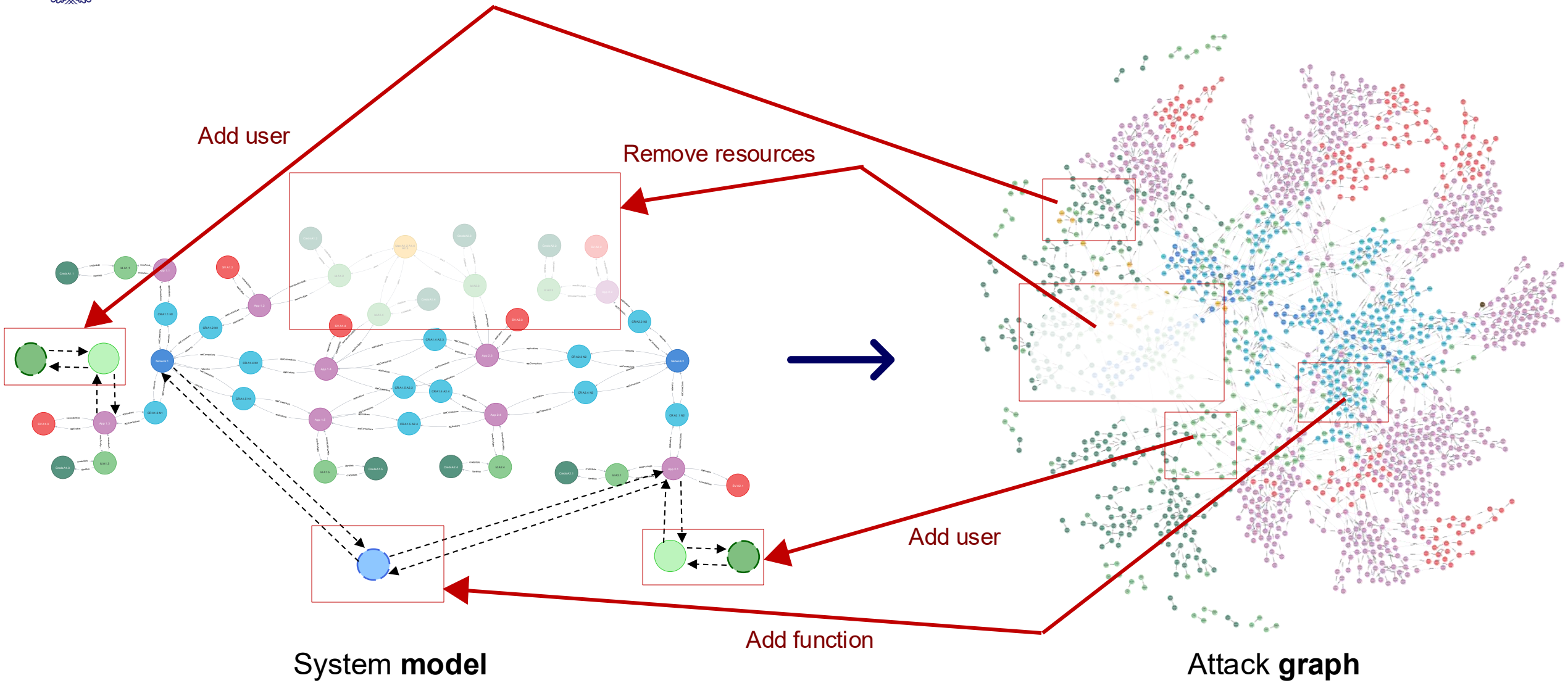
Problem



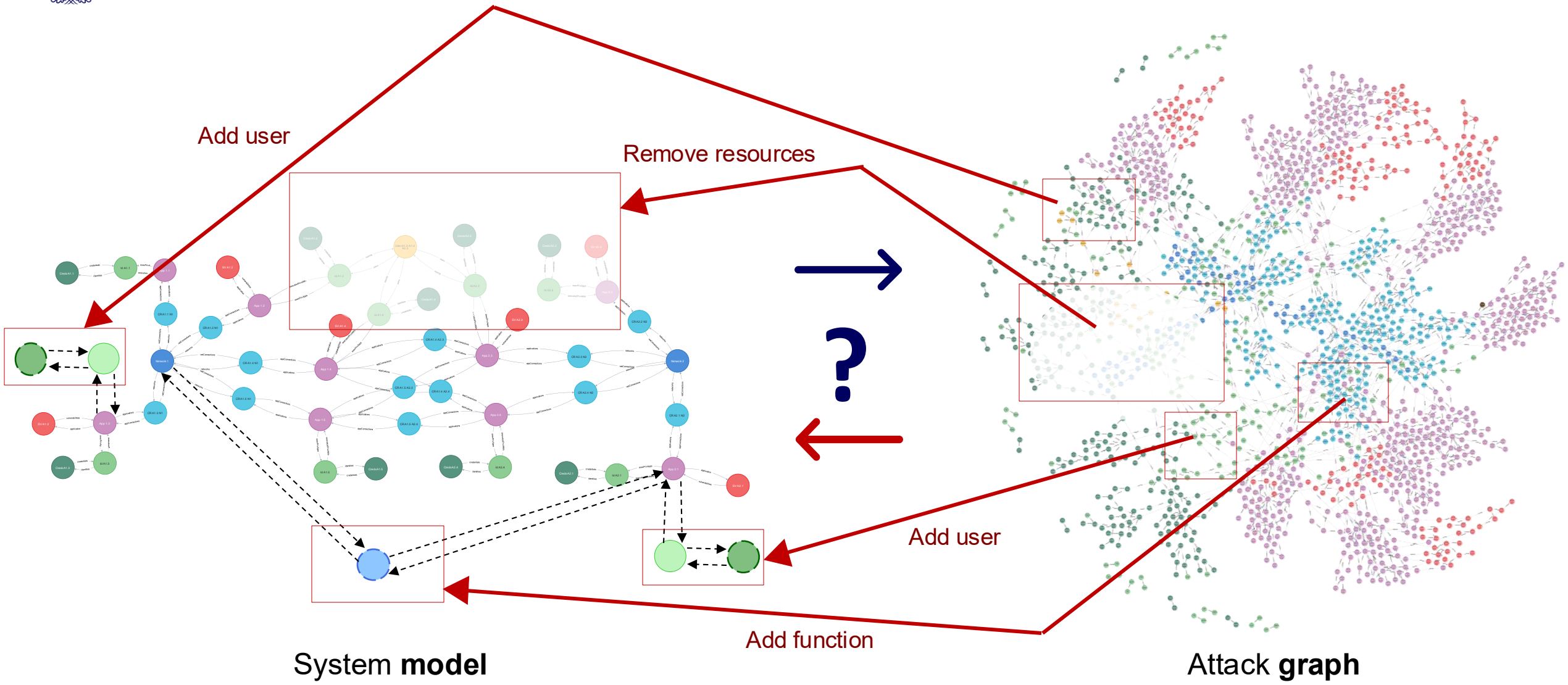
System **model**

Attack **graph**

Problem

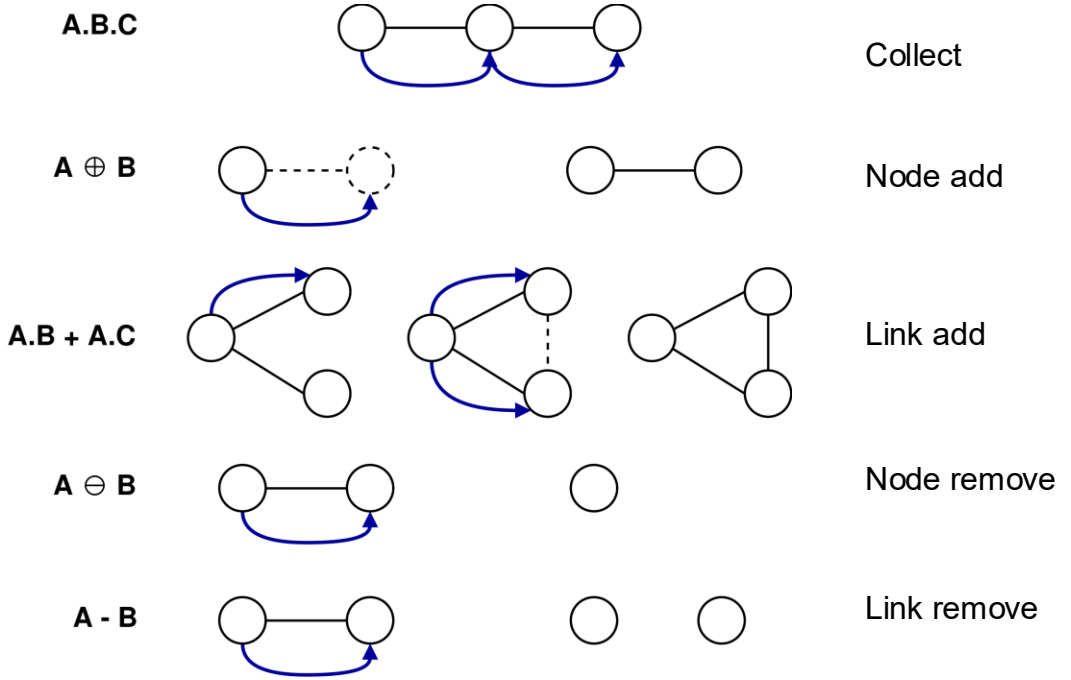

Add user
Remove resources
Add user
Add function
System **model**
Attack **graph**

Problem


Add user
Remove resources
?
Add user
Add function
System **model**
Attack **graph**

DynaMAL

- A low-level graph manipulation system
  - Selecting graph elements
  - Adding/removing graph elements
- A high-level metalanguage: DynaMAL
  - Domain-specific system modeling
  - Domain-specific graph manipulation rules

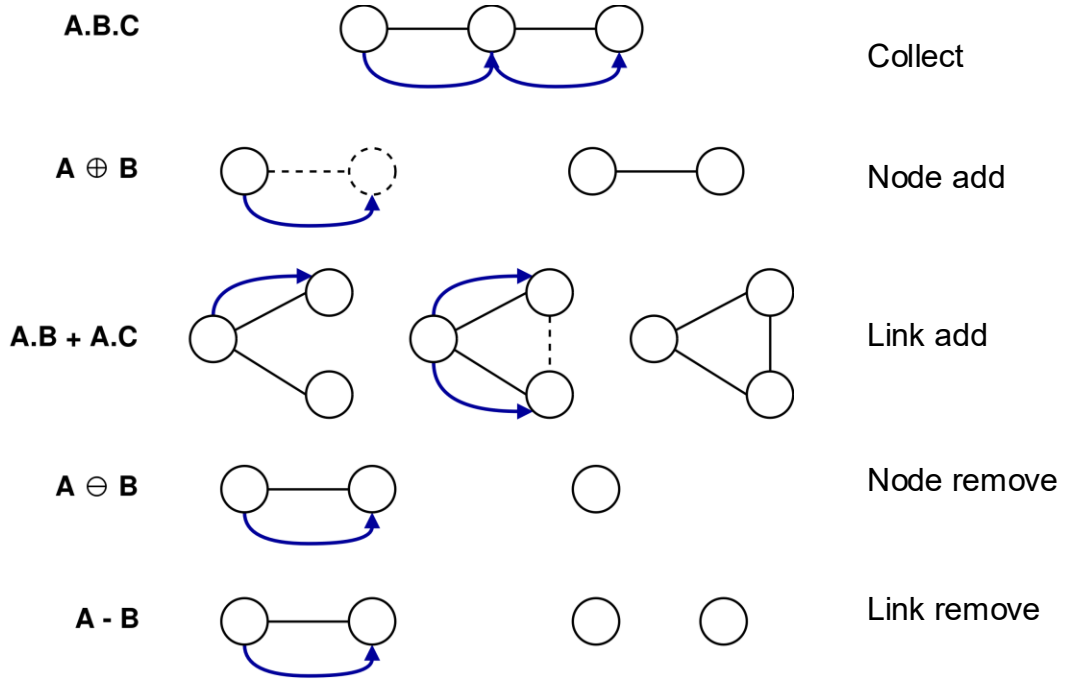



```
// DynaMAL
asset CreateFunction extends APIAction {
  | execute >
    A> Namespaces / resources[Function]
}

asset DeleteFunction extends APIAction {
  | execute >
    R> Namespaces / resource[Function]
}
```

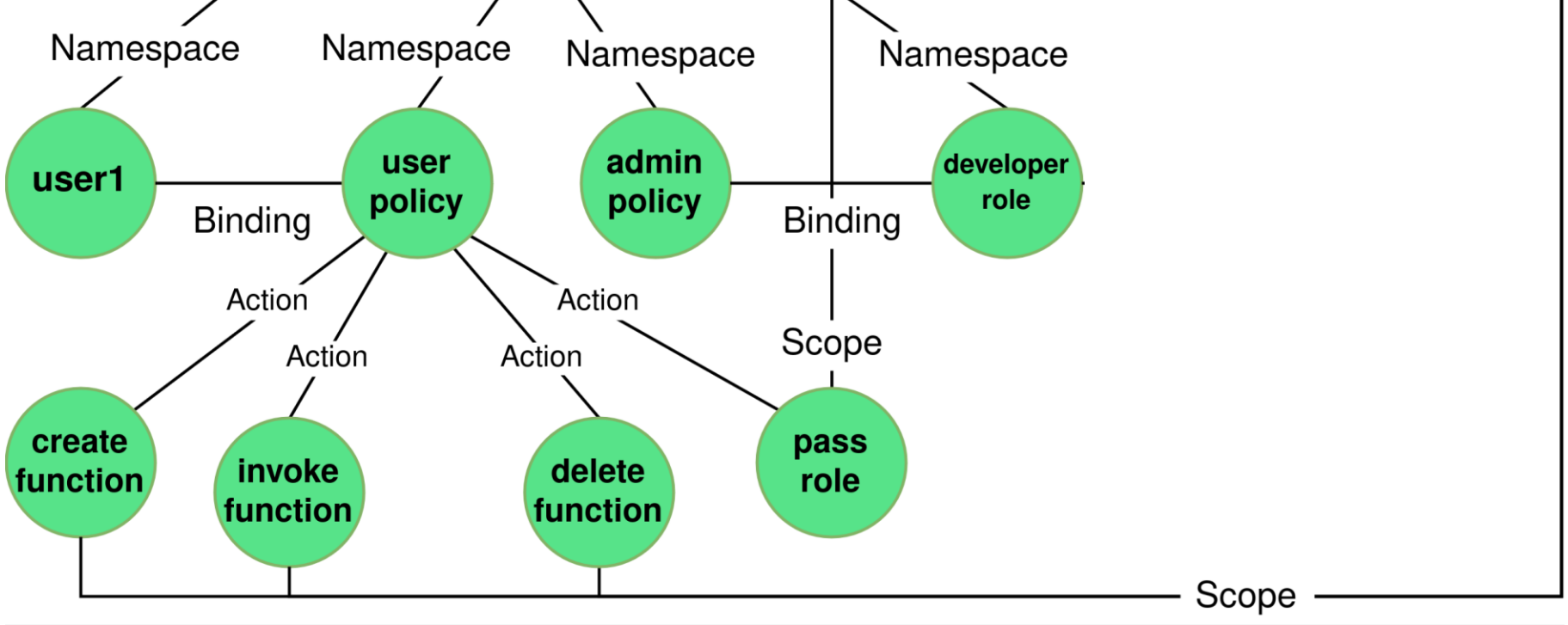
DynaMAL

- A low-level graph manipulation system
  - Selecting graph elements
  - Adding/removing graph elements
- A high-level metalanguage: DynaMAL
  - Domain-specific system modeling
  - Domain-specific graph manipulation rules



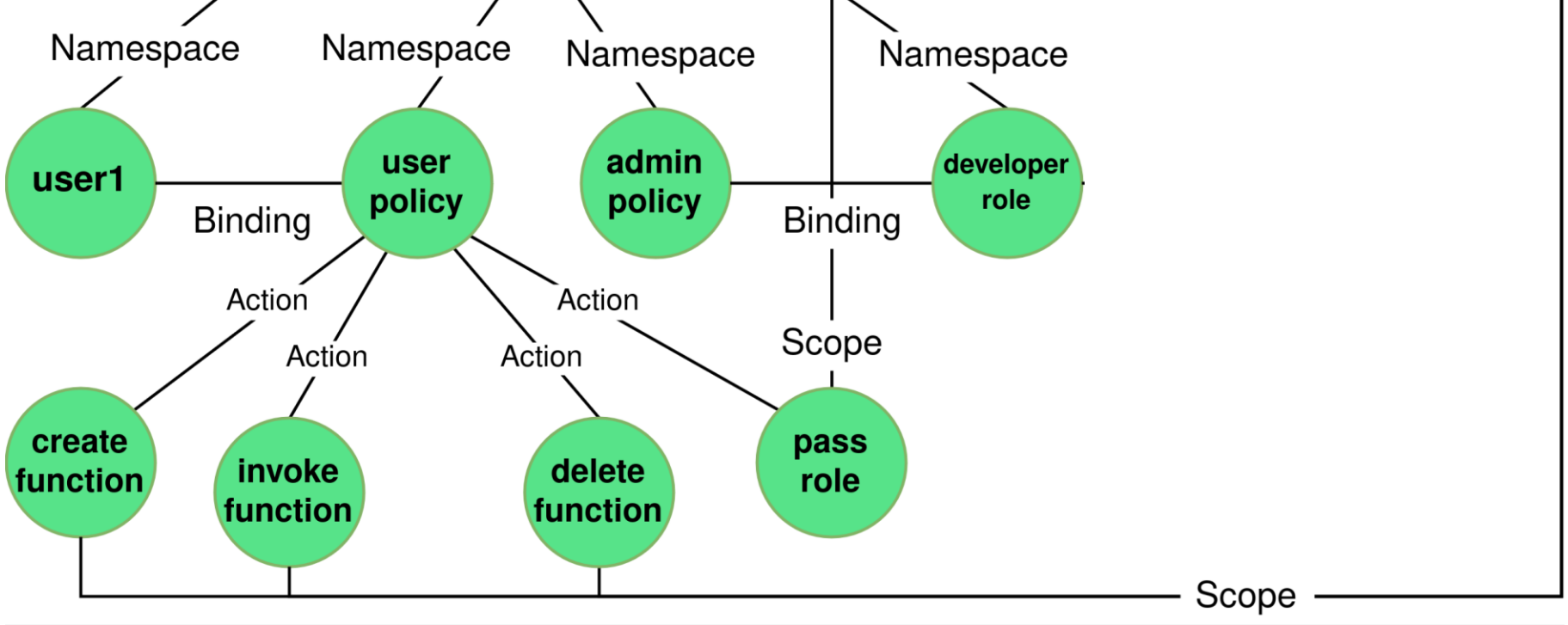

```
// DynaMAL
asset CreateFunction extends APIAction {
  | execute >
    A> Namespaces / resources[Function]
}

asset DeleteFunction extends APIAction {
  | execute >
    R> Namespaces / resource[Function]
}
```

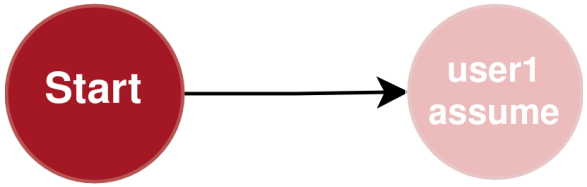


System **model**

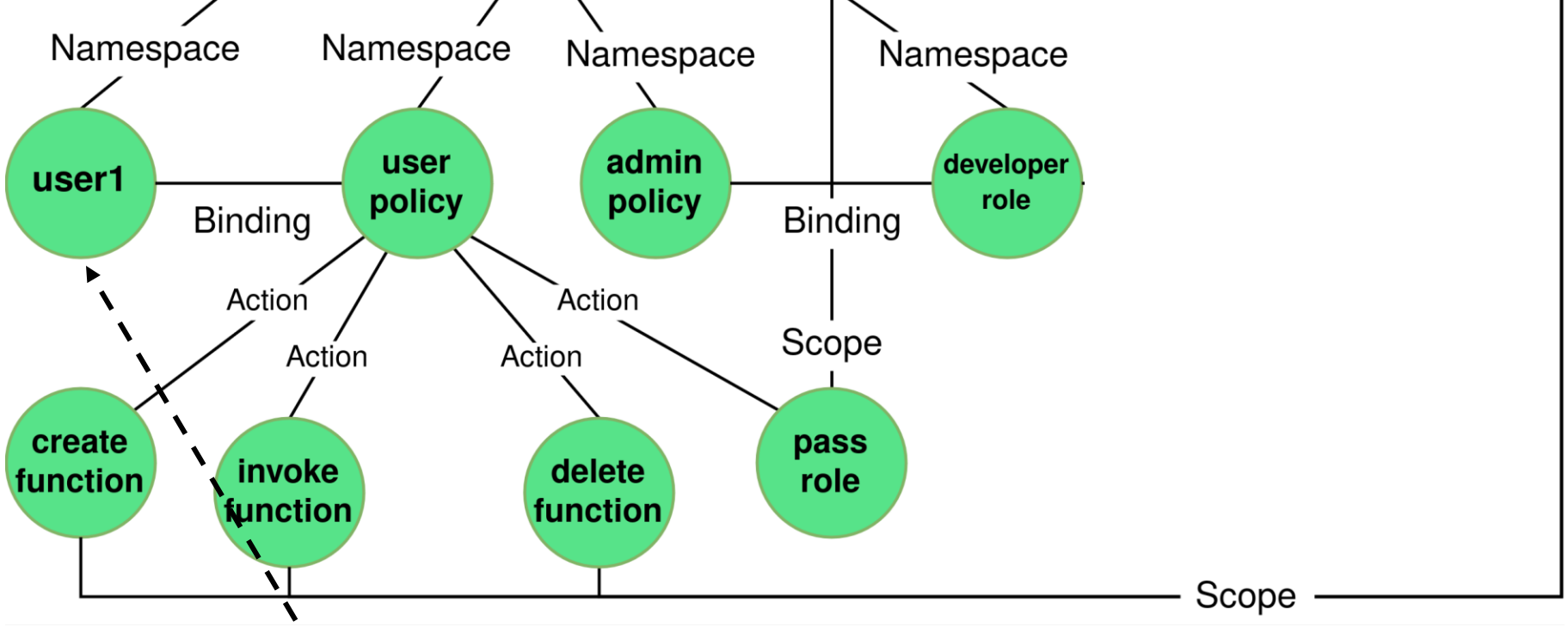
Attack **graph**



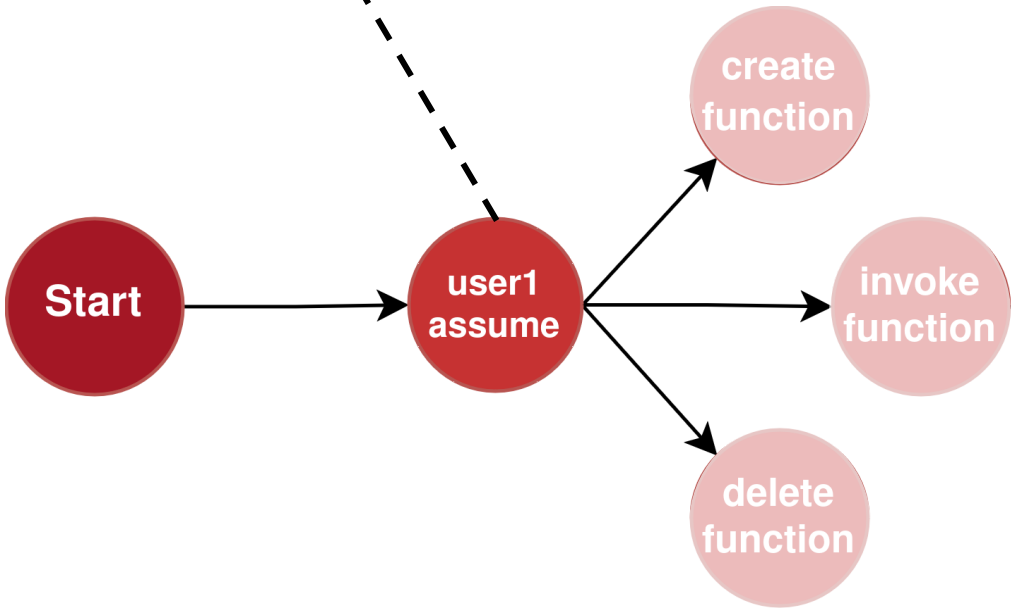
System **model**



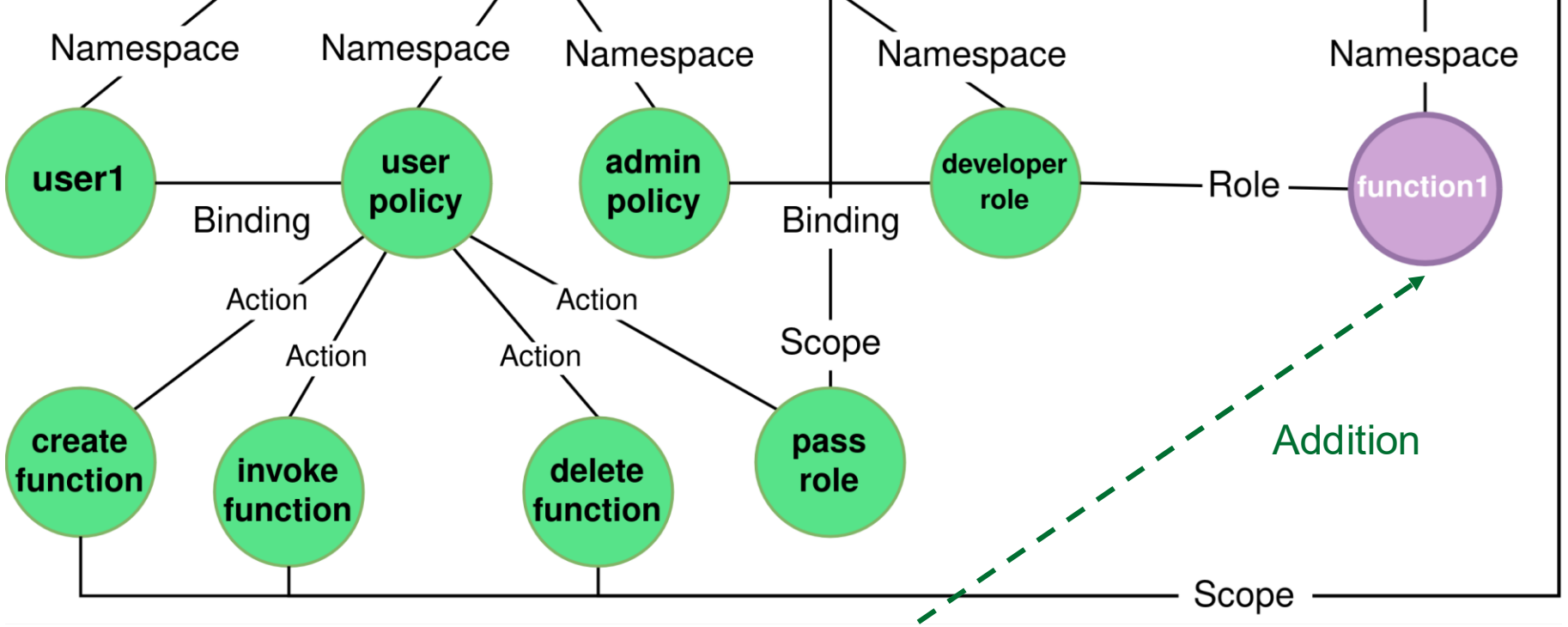
Attack **graph**



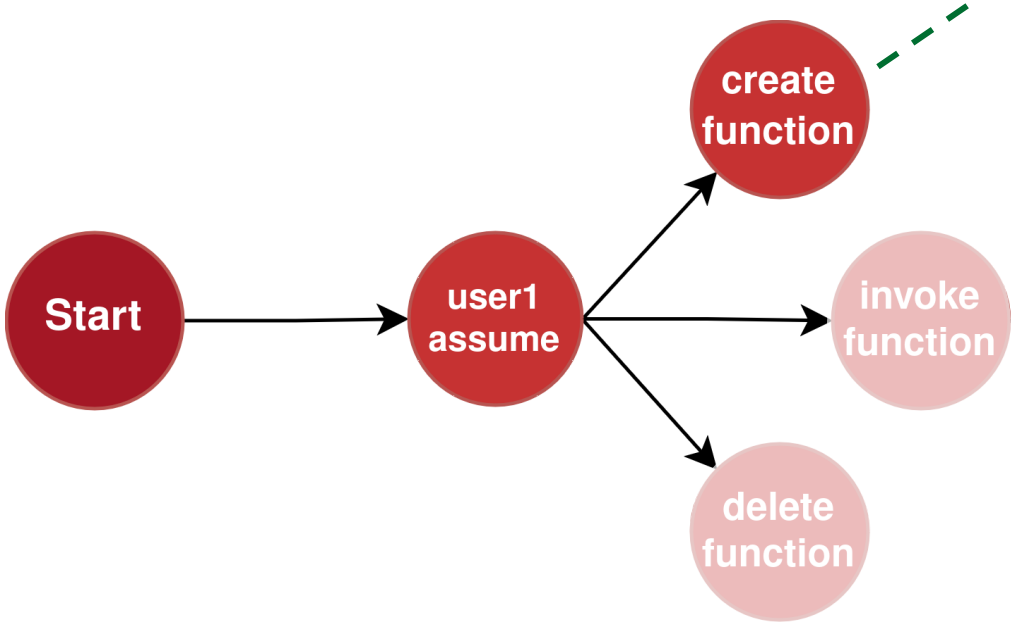
System **model**



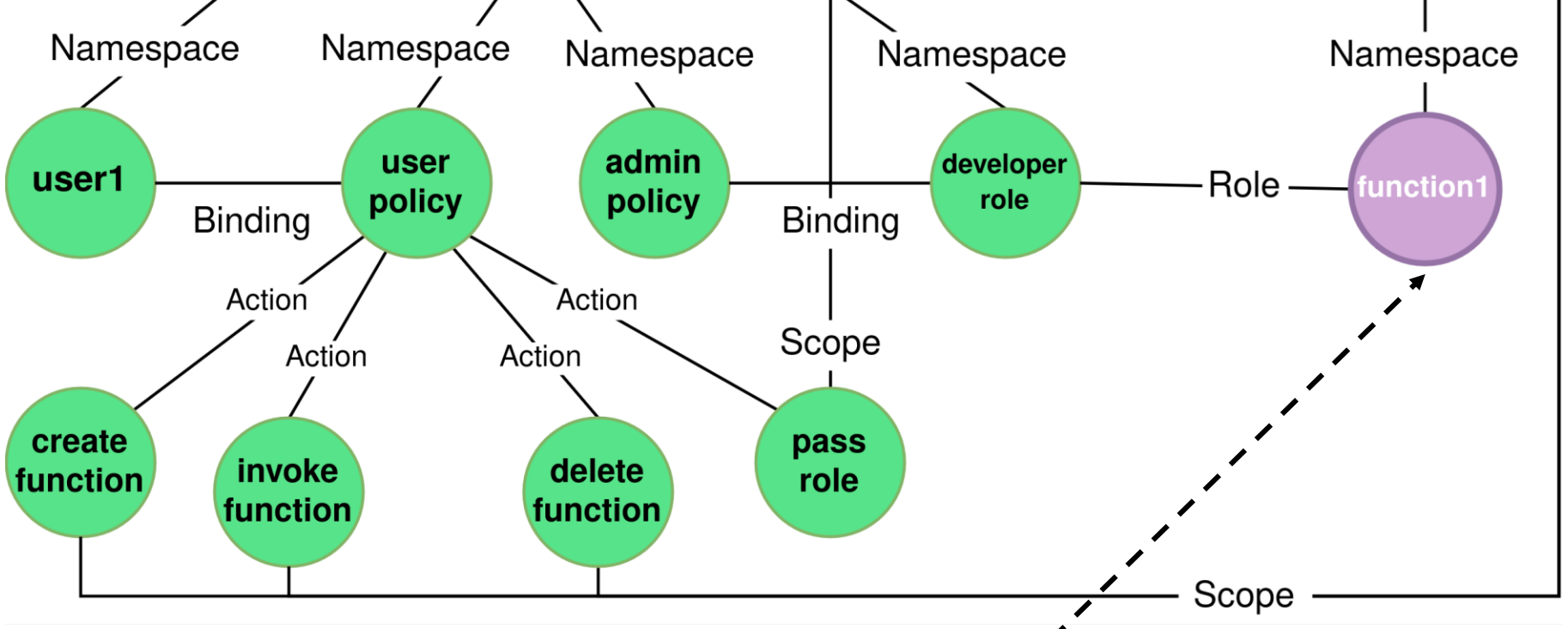
Attack **graph**



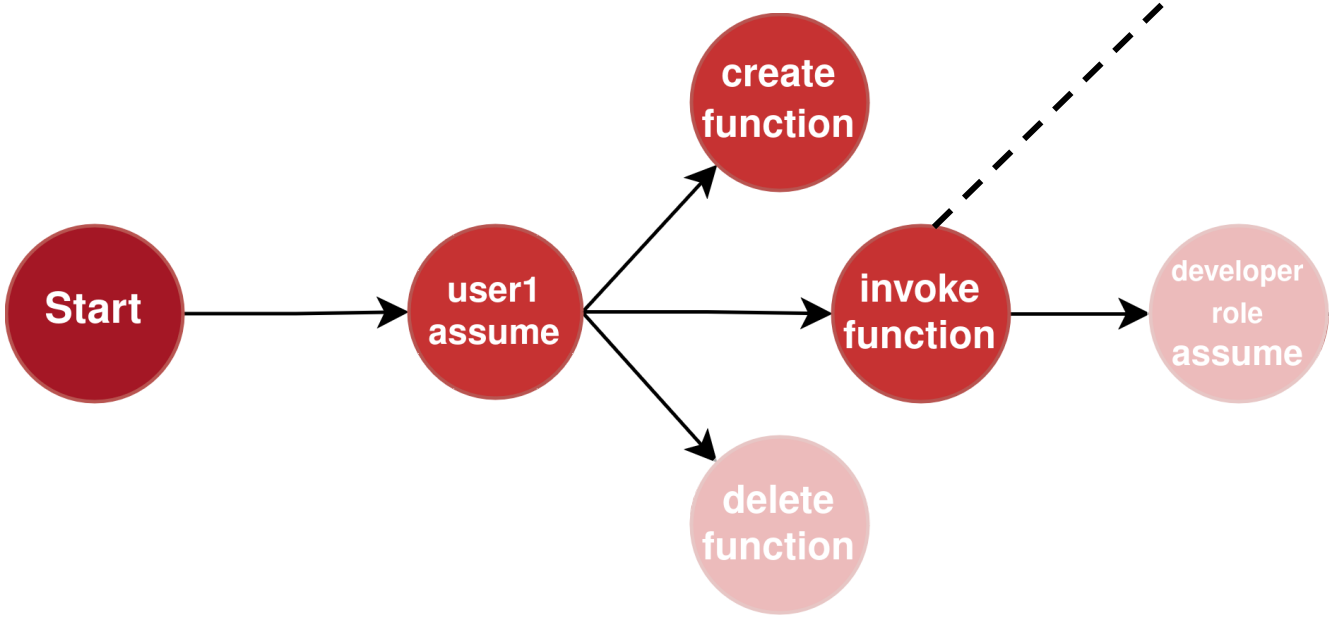
System **model**



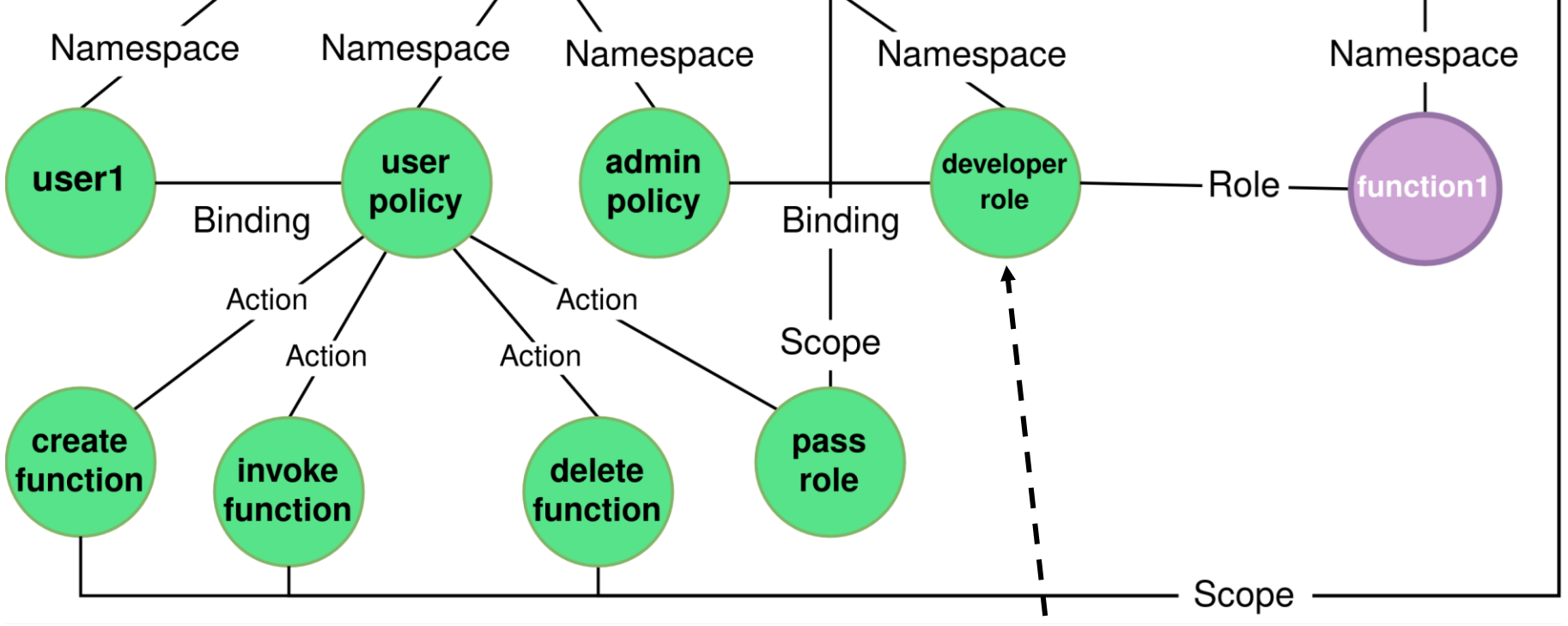
Attack **graph**



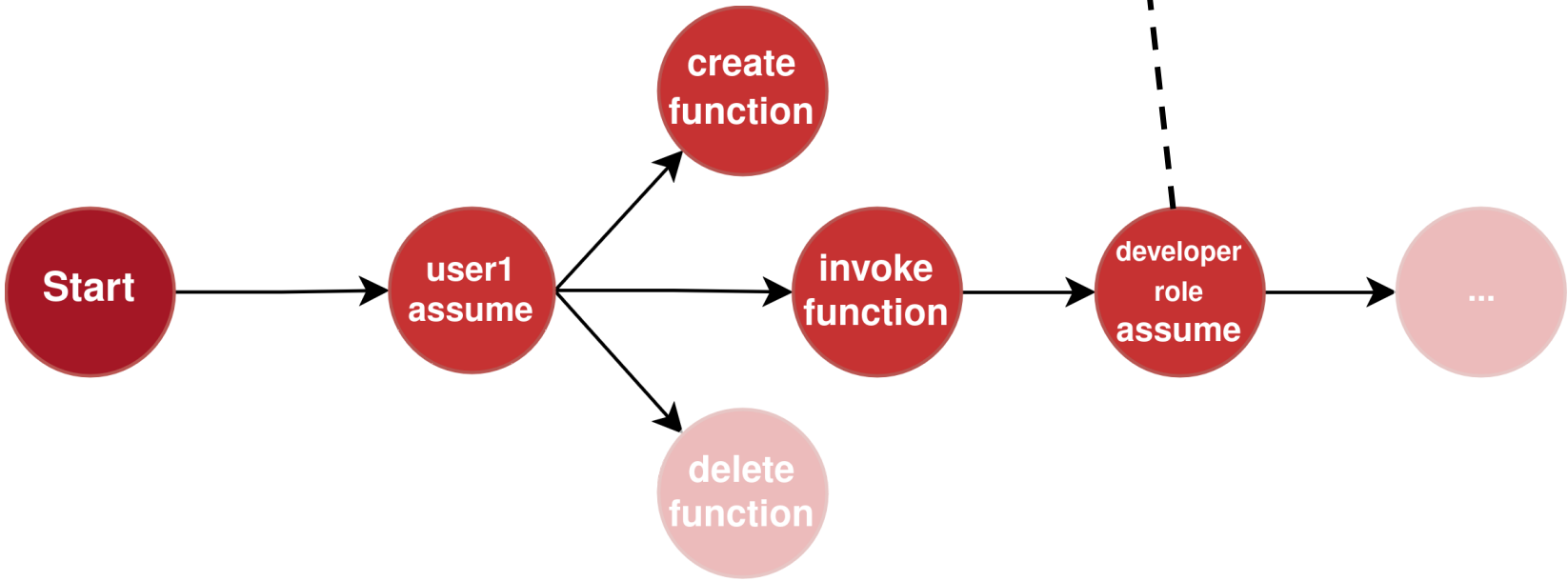
System **model**



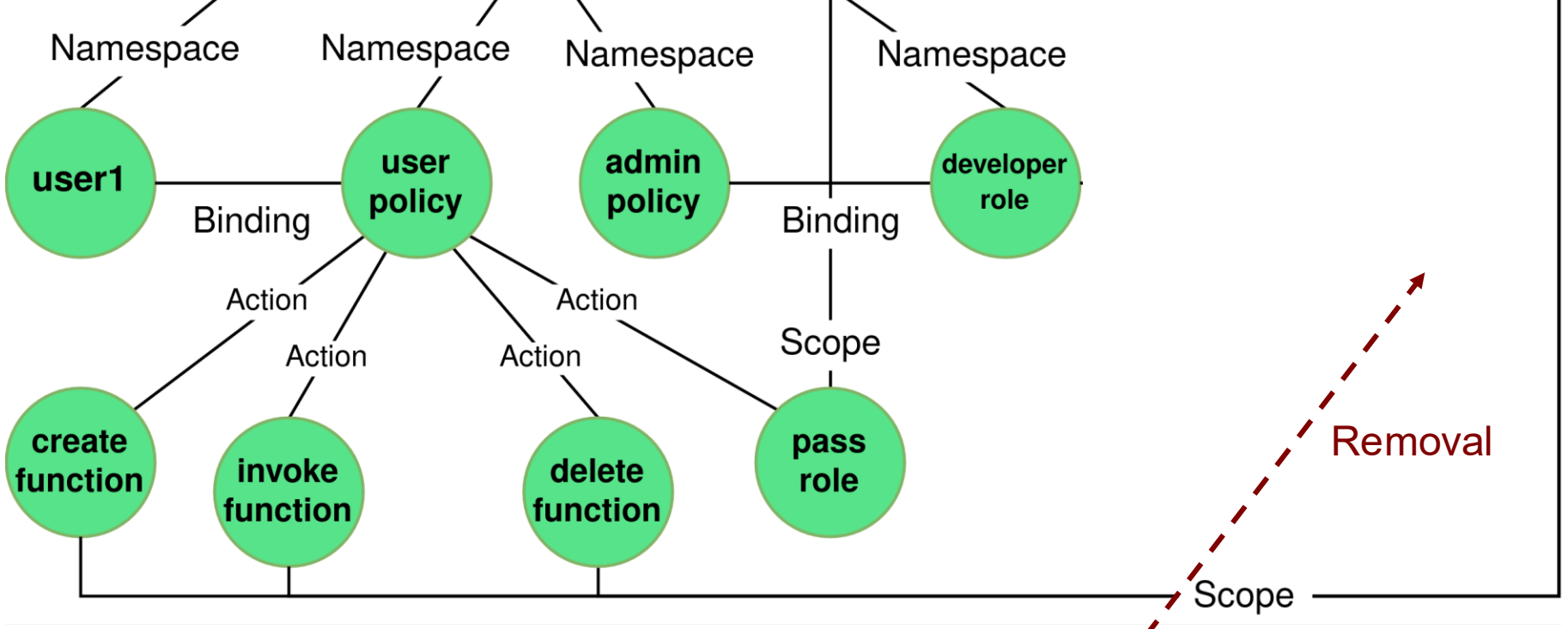
Attack **graph**



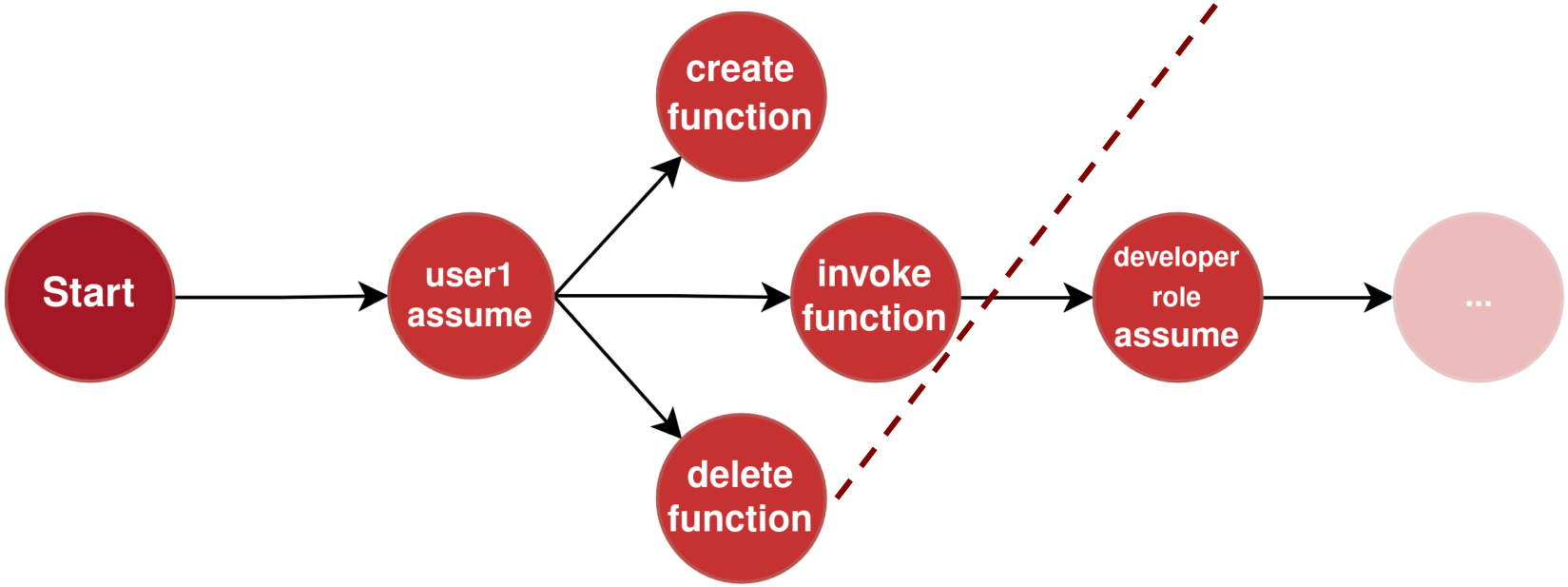
System **model**



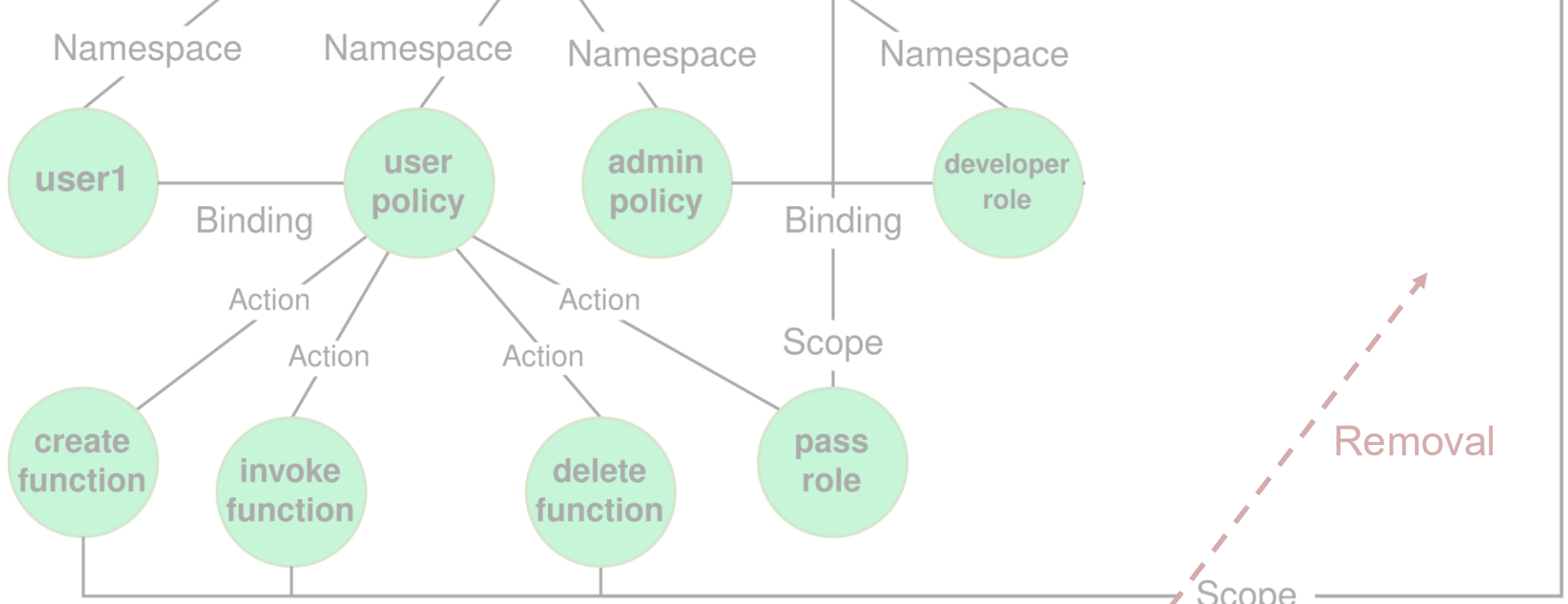
Attack **graph**



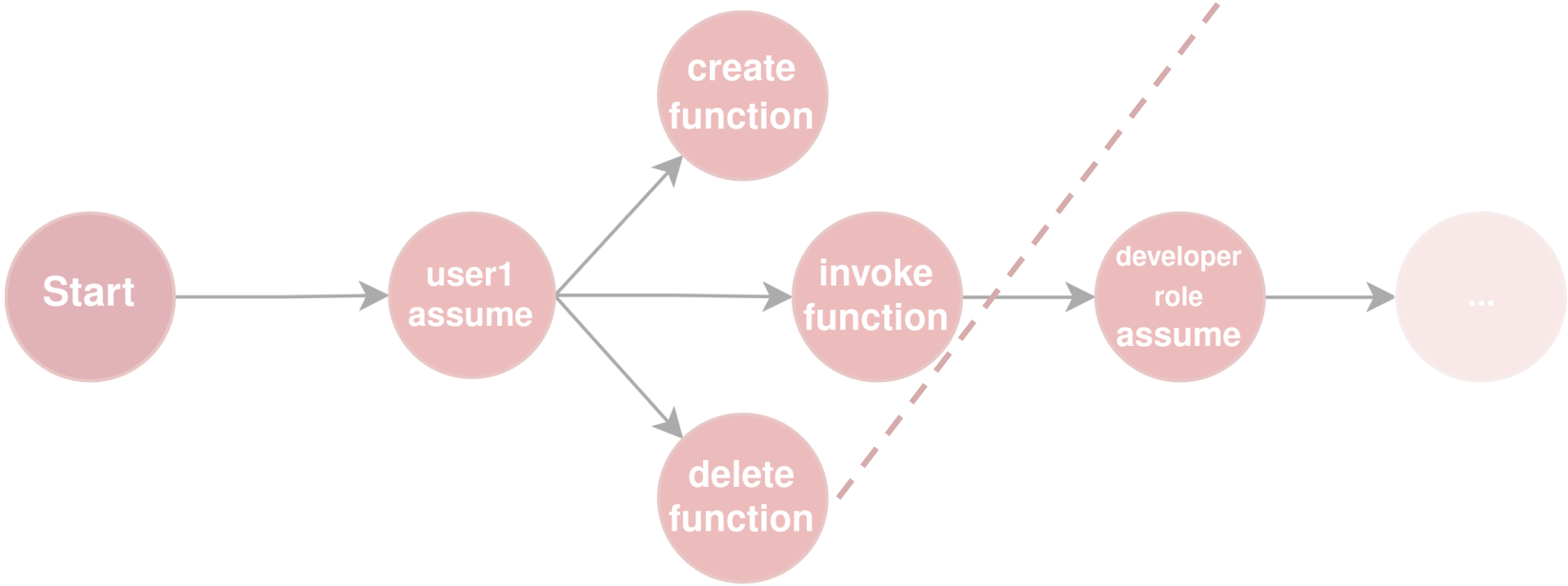
System **model**



Attack **graph**



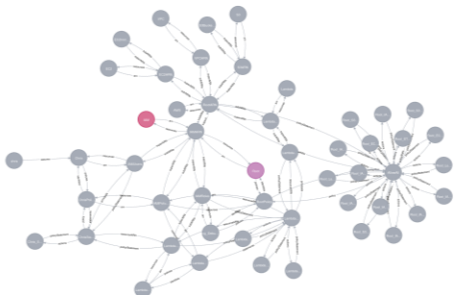
System **model**



Attack **graph**

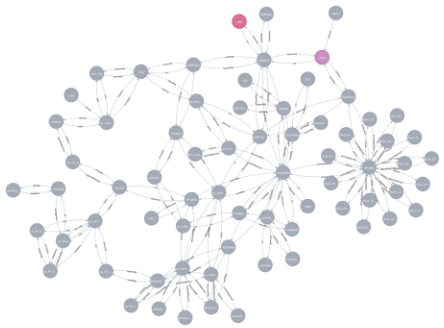
Scenario setup

Lambda



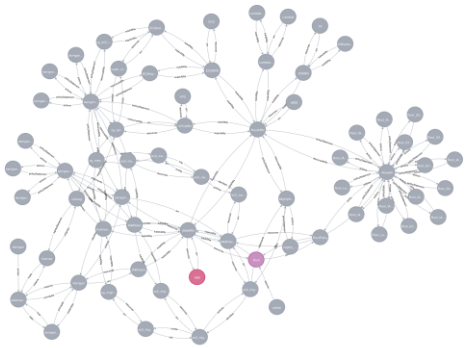
(Easy)

CodeBuild



(Medium)

Attachment



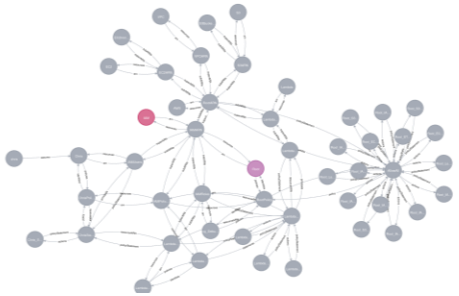
(Hard)

Scenario source: https://github.com/RhinoSecurityLabs/cloudgoat

A search method

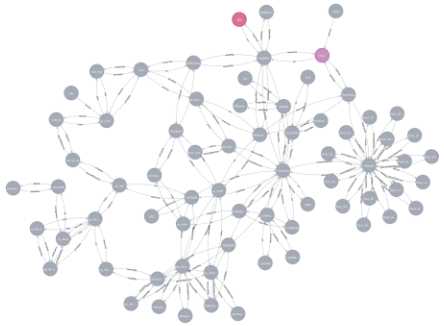
Ant colony optimization

Lambda



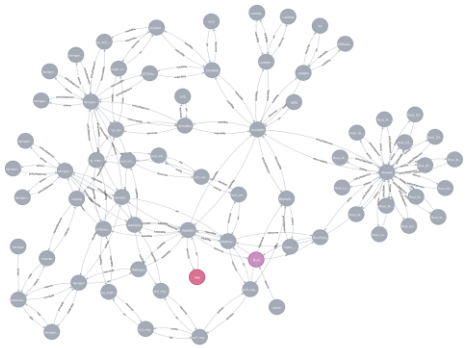
(Easy)

CodeBuild



(Medium)

Attachment



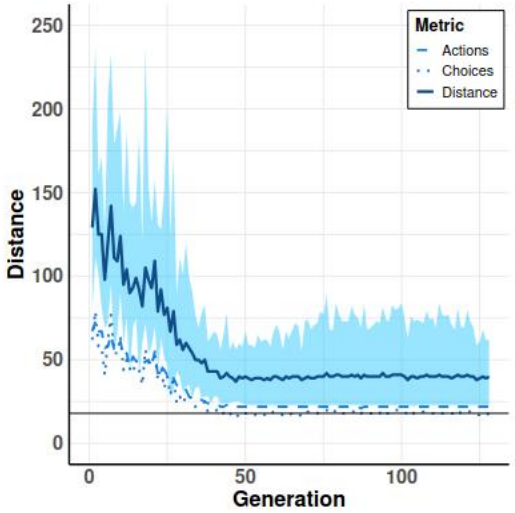
(Hard)

Scenario source: https://github.com/RhinoSecurityLabs/cloudgoat

Results

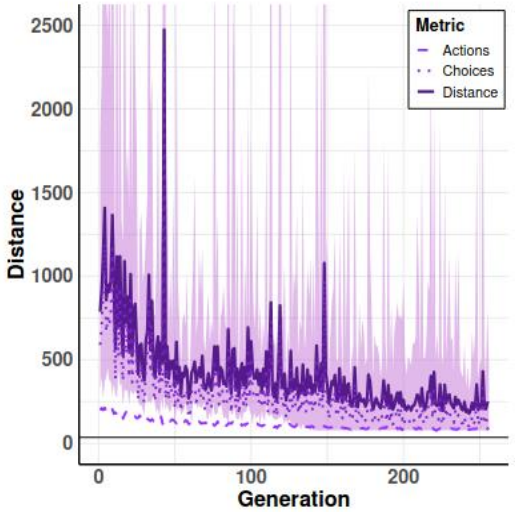
Ant colony optimization

Lambda



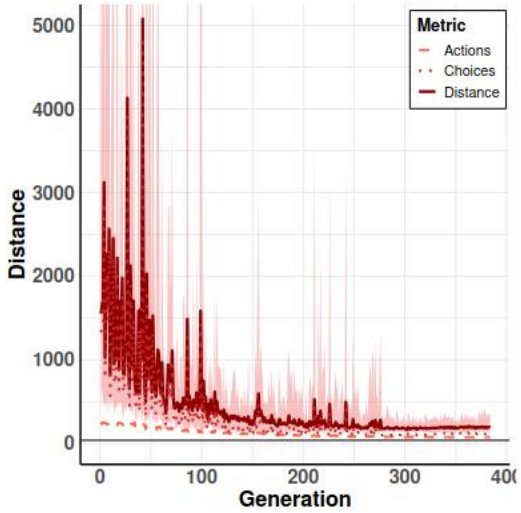
(Easy)

CodeBuild



(Medium)

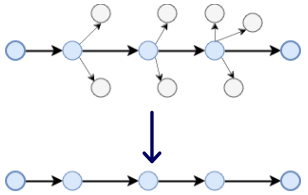
Attachment



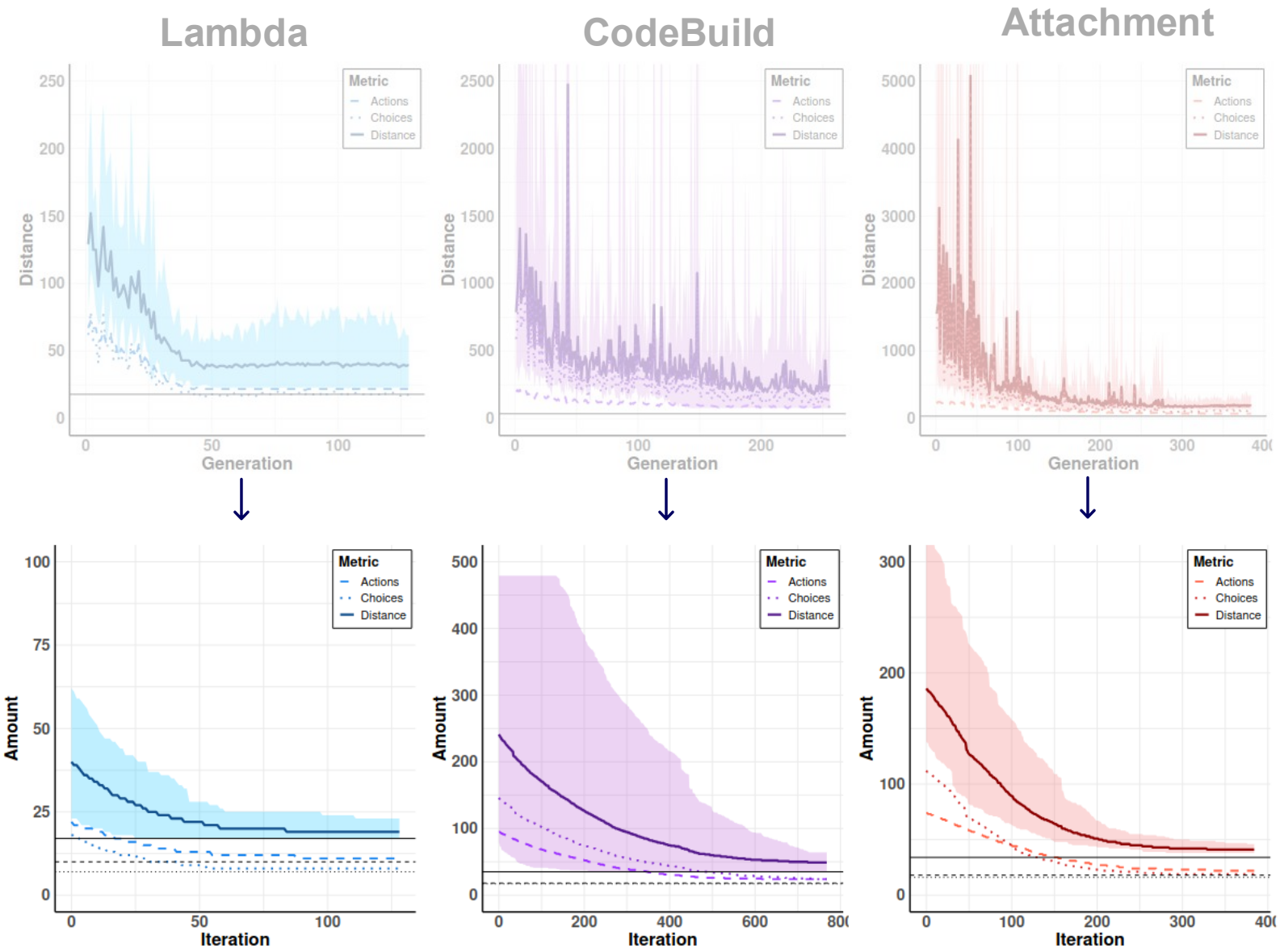
(Hard)

Refining the results

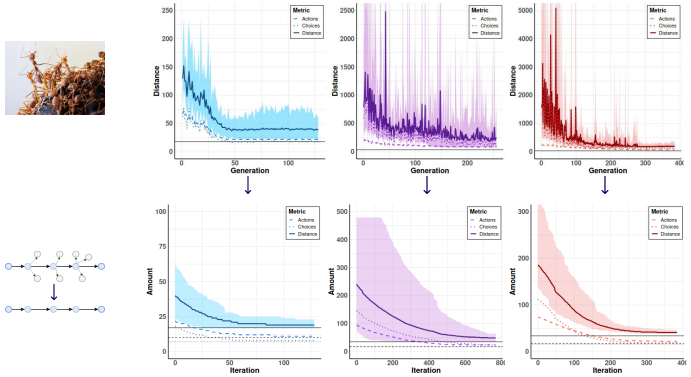
Ant colony optimization



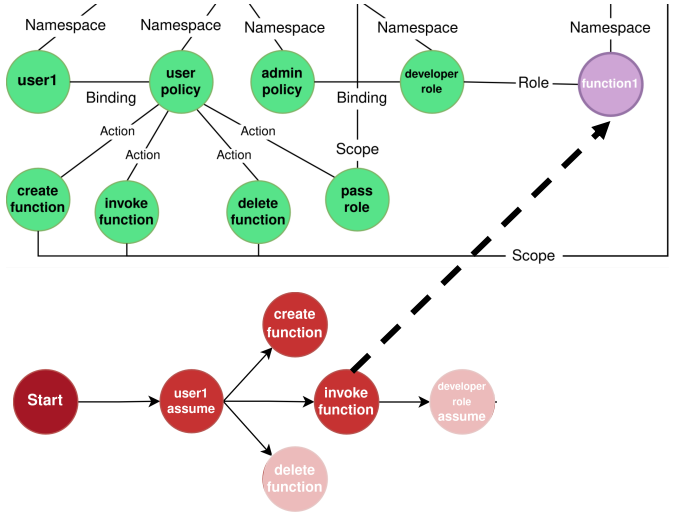
Simulated annealing

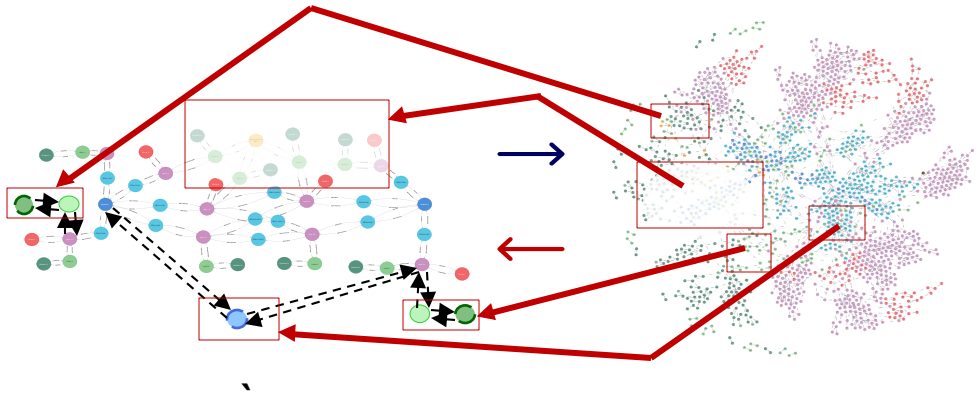


Conclusions



- Solved real hacking challenges automatically
- Implemented adversary-driven dynamics
- Replicated a missing aspect of real attacks





- DynaMAL: https://gitlab.com/kth-ssas/dynamal-group
- Contact: vengs@kth.se