Security Management Using Planning Domain Definition Language: A Case For Ransomware Mitigation

Afaq Inayat, Muriel F. Franco, Eder J. Scheid, Lisandro Z. Granville

Institute of Informatics (INF) – Federal University of Rio Grande do Sul (UFRGS)
Av. Bento Gonçalves, 9500, Porto Alegre – RS – Brazil

[afaq.inayat, mffranco, ejscheid, granville]@inf.ufrgs.br

Abstract. *Network automation is essential for efficiently managing complex networks, enhancing reliability, and reducing human errors. This work proposes a novel approach to automate security planning and management using the Planning Domain Definition Language (PDDL). PDDL is used to generate an action plan to mitigate imminent threats to companies' networks. The work demonstrates how PDDL can be applied to security planning in a dedicated scenario where there is a need to mitigate a ransomware attack targeting a company.*

1. Introduction

Automation plays a crucial role in planning cybersecurity strategies to prevent and mitigate cyberattacks. There are different efforts in both academia and industry focusing on automating different security tasks [Pandey et al. 2022, Franco et al. 2023]. For example, attack graphs have been proposed as a tool to help network administrators understand the potential weaknesses of their networks [Bezawada et al. 2019]. However, there is still room for novel solutions that explore state-of-the-art techniques to improve the process of security planning and automate the mitigation of cyberattacks.

A promising approach is the Planning Domain Definition Language (PDDL) [Haslum et al. 2019], an Artificial Intelligence (AI) planning tool that enables efficient problem-solving by defining domains, predicates, and actions. This can be applied to security planning solutions, such as providing an attack tree to identify adversarial strategies [Falco et al. 2018] and penetration test planning actions [Obes et al. 2013].

In this work, we employ a PDDL as an AI planner to provide a plan of action for security management actions, such as automated incidence response and strategic planning. For that, our approach applies PDDL to determine the sequence of actions to achieve a specific goal, such as avoiding cyberattacks or mitigating an imminent threat. All planning is based on the domain and actions, which can be specified, for example, as a network security problem. For the approach's evaluation, an illustrative scenario is defined in a hypothetical company under a Ransomware attack infecting and compromising its devices. The PDDL is then used to propose a plan and recommend actions to avoid the spread of Ransomware in the network.

The remaining of this work is organized as follows. Section 2 presents related work on PDDL, Section 3 describes the approach, Section 4 presents the evaluation, and Section 5 presents the conclusion.

2. Related Work

AI is becoming vital in computer networking, especially for network security, management, and system evaluation [Wan et al. 2021]. Over the past decade, significant efforts have been devoted to developing flexible and adaptable management solutions to support autonomous and self-managed networks [Pang et al. 2020]. Automated planning is also needed because it offers a range of compelling advantages and can be an ally for the automation of networking management, thus, enhancing not only the operational efficiency but also the cost-effectiveness of managing next-generation networks.

AI Planning, a sub-field in AI, tackles various problems. It involves selecting goal-driven actions based on a high-level world description [Choi et al. 2021]. In [Obes et al. 2013], authors introduce an attack model that generates attack paths for penetration testing, using PDDL for path generation, and validates them through real network exploits. Similarly, [Bezawada et al. 2019] introduces AGBuilder, an AI planning tool for generating, updating, and refining attack graphs. Likewise, [Falco et al. 2018] suggests using AI planners to create attack trees, identifying adversarial strategies for compromising critical infrastructure systems.

3. Approach

Figure 1 presents the overall PDDL approach, including its components and their relationships. The approach takes as an example the case of an attack on a network, where we isolate the infected system and safeguard the rest of the systems. The components composing the solution and their definitions are described in the rest of this section.

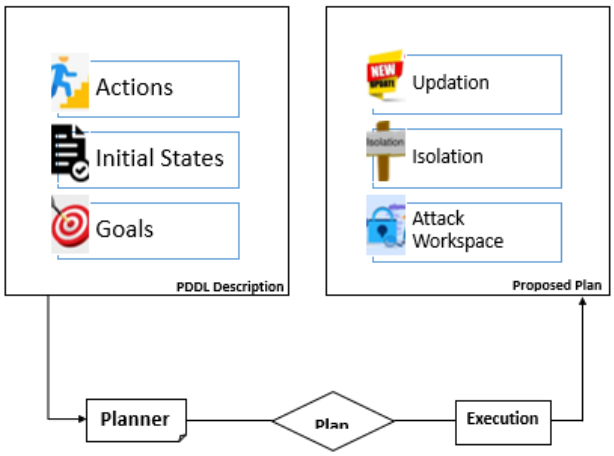

Figure 1. Approach's Flow, Based on [Obes et al. 2013]

The Planner is a tool that takes as input the description of PDDL composed of a Domain and a Scenario/Problem that includes actions, initial states, and goals. The output of the Planner is a Plan, and the execution of a plan is a sequence of actions for the attack workspace that leads to the completion of the goal in case all the actions are successful. In our case, the output of the Planner will be the steps (*e.g.*, isolation and updation) that safeguard the system from the attack. AI Planner has the advantage of requiring no data or datasets to train. Instead, an entity within a domain is modeled, describing the relevant features of an environment, the goals, constraints, and the actions available to the entity.

PDDL can be a valuable tool when defining and implementing Planners. PDDL is designed for task planning, and PDDL-based planners are widely used for a variety of planning problems. PDDL-based planners perform better on problems with longer solutions. To generate a plan for an attack the PDDL definition is composed of two key parts: (1) a PDDL domain definition and (2) a PDDL problem description.

A **PDDL Domain** is a high-level description of a set of problems and the corresponding actions and constraints involved. In the PPDL domain, we specify the requirements (such as device types) that will support the available actions for the attack, the pre-conditions and post-conditions are the effect of actions. The planner will only accept the domain if it supports all the requirements described in the domain.

The domain definition starts with defining a planning domain of a specific scenario, such as a Ransomware defense. Requirements or features will be used by the domain to define the types of objects within the domain. Within this domain, there are specific types of devices, namely routers, switches, and computers.

In PDDL, predicates are used to describe properties of objects or relations between objects that can be true or false. In the provided code shown in Figure 2, the predicates are used which are essential in defining the initial and goal state for the Ransomware attack (predicates such as Infected, isolated, patched, and clean). For example, the predicate *(infected ?d - device)* indicates that a specific device, represented by the variable *?d*, is infected by Ransomware because we want to keep track of the infected device.

```
(:predicates
  (infected ?d - device)
  (connected ?d1 ?d2 - device)
  (isolated ?d - device)
  (patched ?d - device)
  (cleaned ?d - device)
  (all-patched)
)
```

Figure 2. Predicates Defined to Describe Devices States Using PDDL

Model-related actions that satisfy the predicates in the right cases are required in the PDDL specification. For example, in Figure 3 we need to perform the action ”isolate” for the isolation of the computer in a scenario of Ransomware attack for this we need to define parameters *(i.e., (?d - device))*. In this case, there is the *?d* is a variable of type device. This indicates that the action can be applied to any device.

The precondition section inside the action ”isolate” specifies the conditions that must be true for the action to be applicable or executable *(i.e., :precondition (and (infected ?d) (not (isolated ?d)))*. In precondition ”and” is a logical conjunction, meaning all conditions inside must be true. *(infected ?d)* means that the device *?d* must be infected for the action to be applicable.

In this case, the device is infected by Ransomware. *(not (isolated ?d))* states that the device *?d* should not already be isolated. Collectively, these preconditions state that the action isolate can be applied to a device *?d* if and only if Ransomware infects the device and it is not already isolated. The effect section *(i.e.,* :effect (isolated ?d)) describes the changes in the world state that will result after the action is successfully

```
(:action isolate
  :parameters (?d - device)
  :precondition (and (infected ?d) (not (isolated ?d)))
  :effect (isolated ?d)
)
(:action patch
  :parameters (?d - device)
  :precondition (and (not (infected ?d)) (not (patched ?d)))
  :effect (patched ?d)
)
(:action all-devices-patched
  :parameters ()
  :precondition (forall (?d - device) (or (patched ?d) (infected ?d)))
  :effect (all-patched)
)
(:action clean
  :parameters (?d - device)
  :precondition (and (infected ?d) (isolated ?d) (all-patched))
  :effect (and (not (infected ?d)) (cleaned ?d))
)
```

Figure 3. Definition of Security Actions Using PDDL

executed. *(isolated ?d)* means that after the execution of the isolate action on device *?d*, the device will be in an isolated state. Similarly in Figure 3, three other actions are defined: "patch", "clean", and "all_devices_patched".

A PDDL problem contains an initial state: a set of predicates set to true initially, and a goal state: a set of predicates that may or may not be true with the actions defined in the domain. It is needed then to define the problem (*i.e.*, Ransomware-attack) and the domain that associated the problem with the domain (*e.g.*, Ransomware-defense).

The objects section lists specific objects (or instances) in a Ransomware attack scenario, including routers (r), switches (s), personal computers (pc), and a gateway (gw) for the network. The initial state describes what is true about the system when the planner starts. Figure 4 shows the Initial states and relationships. Figure 5 shows the setup and network connectivity based on Figure 4. This states that initially the pc3 is infected by Ransomware and also shows the topology of the network, like how the network devices are connected. Different devices are involved in this topology (*i.e.*, routers, switches, and computers).

```
(:init
  (infected pc3)
  (connected r1 s1)
  (connected r2 s2)
  (connected r1 gw1)
  (connected s1 s3)
  (connected s3 pc1)
  (connected s3 pc2)
  (connected s3 pc3)
  (connected s1 pc4)
  (connected s2 pc5)
  (connected s2 pc6)
)
```

Figure 4. Initial States and Relationships Defined using PDDL

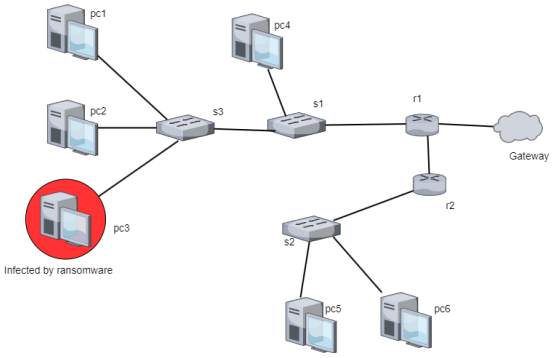

Figure 5. Network Topology Generated from Initial Configurations Defined in Figure 4

In PDDL, the *(: goal ...)* section defines the desired state or conditions that the planner should aim to achieve. The goal state denotes the desired final state of the world for the given Ransomware attack. Figure 6 shows an example of a goal that aims to

isolate the infected device, patch all other vulnerable devices in the network to stop the Ransomware from spreading, and finally, clean the infected system.

Also, it states that for all devices *(?d)* certain conditions must be true, such as a device must be patched or cleaned for successful plan execution. These conditions are important because it is used by the **PDDL Planner** to provide a plan that achieves all of the goals described (*i.e.*, all infected and connected devices must be patched or cleaned).

```
(:goal
  (and
    (forall (?d - device) (or (patched ?d) (cleaned ?d)))
    (not (exists (?d1 ?d2 - device) (and (infected ?d1) (connected ?d1 ?d2))))
  )
)
```

Figure 6. Definition of Ransomware Mitigation as a Goal Using PDDL

The **PDDL Planner** aims to solve a PDDL problem by finding a plan that satisfies it. A successful plan is a sequence of actions from those specified in the PDDL problem for a given initial state. The output of the planner is a plan that will reach us to the desired goal for the problem of Ransomware attack.

The **PDDL Plan** is a sequence of actions generated from the initial state in the PDDL problem to the final state in the PDDL problem. A plan can also be thought of as a sequence of transitions from the initial state to the goal state. The plan after the execution of the Planner will be evaluated in Section 4.

4. Evaluation

The evaluation was conducted in an illustrative scenario, as depicted in Figure 5. It highlights the WannaCry Ransomware infecting a hypothetical organization's network. To contain and mitigate the impact of the attack, the organization utilized PDDL to create a comprehensive plan. The organization's network comprises routers (r1, r2), switches (s1, s2, s3), personal computers (pc1, pc2, pc3, pc4, pc5, pc6), and a gateway (gw1). Following the approach outlined in Section 3, the PDDL plan was executed, involving isolating infected computers and patching devices. The PDDL plan to mitigate WannaCry's impact is as follows:

(isolate pc3) (patch r1) (patch r2) (patch s1) (patch s2) (patch s3) (patch pc1) (patch pc2) (patch pc4) (patch pc5) (patch pc6) (all-devices-patched) (clean pc3)

The plan begins by isolating pc3, a crucial step to prevent WannaCry from spreading. Then, it focuses on patching routers and switches, ensuring Ransomware cannot exploit device vulnerabilities. Next, it patches all other computers to reduce infection risk, and the action (all-devices-patched) verifies complete patching for safety. This ensures no devices remain vulnerable.

Finally, it cleans pc3 to remove WannaCry remnants. PDDL's use for a comprehensive WannaCry mitigation plan has proven highly effective, containing the outbreak by isolating infected devices, patching all network components and computers, and verifying successful patch implementation, preventing further damage.

5. Conclusion

In conclusion, we employed a PDDL planner to address Ransomware's impact within the organization. The use case described in this paper proved that the PDDL-based Ransomware mitigation plan effectively contains the outbreak by isolating infected devices, patching network components and computers, and verifying patches. This systematic approach minimized immediate damage in our hypothetical scenario.

We argue that PDDL can be used for different security scenarios, such as penetration tests and defense planning. Therefore, PDDL is not only restricted to incidence response (*e.g.*, Ransomware case analysis) but can be explored within different contexts and tasks related to security and network management. However, applying PDDL to large, complex networks with numerous devices and infrastructure components might present challenges. PDDL is primarily designed for symbolic planning tasks and the time to generate a plan could be prohibitive.

Future work involves expanding PDDL for various attacks scenarios, streamlining cybersecurity workflows, and integrating Machine Learning (ML) algorithms for improved threat detection and response in the evolving cybersecurity landscape.

References

Bezawada, B., Ray, I., and Tiwary, K. (2019). AGBuilder: an AI tool for automated attack graph building, analysis, and refinement. In *Data and Applications Security and Privacy (DBSec 2019)*, pages 23–42, Charleston, SC, USA. Springer.

Choi, T., Ko, R. K., Saha, T., Scarsbrook, J., Koay, A. M., Wang, S., Zhang, W., and St Clair, C. (2021). Plan2defend: Ai planning for cybersecurity in smart grids. *2021 IEEE PES Innovative Smart Grid Technologies-Asia (ISGT Asia)*, pages 1–5.

Falco, G., Viswanathan, A., Caldera, C., and Shrobe, H. (2018). A master attack methodology for an ai-based automated attack planner for smart cities. *IEEE Access*, 6:48360–48373.

Franco, M. F., Granville, L. Z., and Stiller, B. (2023). CyberTEA: a Technical and Economic Approach for Cybersecurity Planning and Investment. In *36th IEEE/IFIP Network Operations and Management Symposium (NOMS 2023)*, pages 1–6.

Haslum, P., Lipovetzky, N., Magazzeni, D., Muise, C., Brachman, R., Rossi, F., and Stone, P. (2019). *An introduction to the planning domain definition language*, volume 13. Springer.

Obes, J. L., Sarraute, C., and Richarte, G. (2013). Attack planning in the real world. *arXiv preprint arXiv:1306.4044*.

Pandey, A. B., Tripathi, A., and Vashist, P. C. (2022). A survey of cyber security trends, emerging technologies and threats. *Cyber Security in Intelligent Computing and Communications*, pages 19–33.

Pang, L., Yang, C., Chen, D., Song, Y., and Guizani, M. (2020). A survey on intent-driven networks. *IEEE Access*, 8:22862–22873.

Wan, H., Liu, G., and Zhang, L. (2021). Research on the application of artificial intelligence in computer network technology. In *5th International Conference on Electronic Information Technology and Computer Engineering*, pages 704–707.