Dynamic Attack Trees

Aliyu Tanko Ali, Damas Gruska

Comenius University, Mlynská dolina, 842 48 Bratislava, Slovak Republic

Abstract

Safety-critical assets are facing enormous threats, and their continue existence lies in the ability to identify and mitigate these threats before they are realised. One good example of such assets is autonomous vehicles, a form of (application) AI that is equipped with self-awareness mechanisms that allow it to interact with a set of objects from the physical surrounding. However, this interaction can result in exposing the asset to a new set of undesired vulnerabilities that are difficult to identify. One of the traditional threat analysis methods used for threat modelling of different kinds of assets is attack trees, a well-known formalism used in security, safety, as well as risk analysis. In this paper, we propose an extension of attack trees, called *dynamic attack trees*. This allows us to model and analyse assets with dynamic threat environments that can interacts with external objects over time.

Keywords

Attack Trees, Cyber-Physical Systems, Security, Formal Modelling, Artificial Intelligence

1. Introduction and motivation

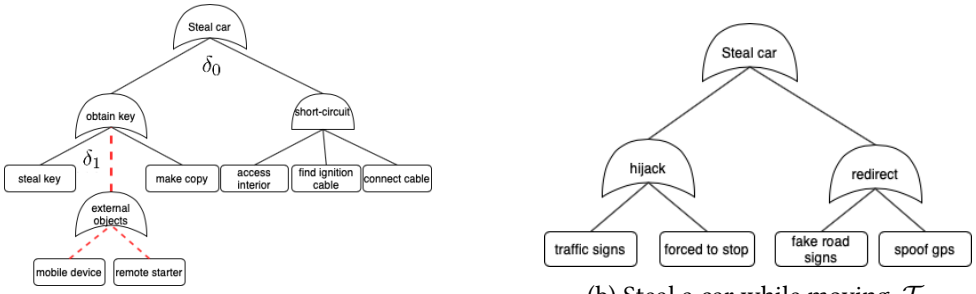
In recent years, safety-critical infrastructures (CI) become widely adopted as part of digital transformation. These involve using computing systems, networks, and robotics to form the backbone of a nation's economy. Technologies such as industrial control systems, embedded systems, IoT systems, artificial intelligence (AI), and cyber-physical systems (CPS) etc., emerged to improve performance, efficiency and users satisfaction. Protecting CI's ecosystem against potential attacks is crucial as most legacy systems and infrastructures integrate emerging technologies while maintaining their original design and characteristics. For example, traditional infrastructures/systems (i.e. vehicles, homes, doors) that integrates advanced technology to operate as CPS (i.e. autonomous vehicles, smart-homes, smart-doors/locks) are visible everywhere. However, the threat analysis approach used for the legacy systems is still adopted for the CI (i.e. CPS), even though the threat environments, as well as the attacker's knowledge about the new technology's operations, differs [1, 2] (i.e. legacy assets are designed as single entity, while CI consists of different (sub)systems that can sometimes run concurrency).

One such approach for threat analysis methods is the attack trees [3], a popular approach used for threats [4], and risk analysis for different kinds of assets [5, 6, 7]. Using this formalism, varying ways an asset may be compromised are modelled as a tree. The root of the tree is the attack target, it is further broken down (decomposed) into smaller units that are easier to solve, known as internal nodes (sometimes referred to as sub-goals). This process is repeated until the

OVERLAY 2021: 3rd Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis, September 22, 2021, Padova, Italy

aliyu.ali@fmph.uniba.sk (A. T. Ali); gruska@fmph.uniba.sk (D. Gruska)

CEUR Workshop Proceedings (CEUR-WS.org)



(a) Steal a car while parked, $\mathcal{T}_1$

(b) Steal a car while moving, $\mathcal{T}_2$

Figure 1: A simple example of how to steal a car at different threat environments $\mathcal{T}_1$ and $\mathcal{T}_2$

sub-goal(s) can no longer be broken down, and a set of leaf nodes is reached. The leaf nodes represent an atomic action that an attacker can execute to initiate an attack. Over the past years, attempts has been made to extend attack trees to analyse different kinds of assets and attack scenarios. Some of these extensions includes attack-defence trees, attack countermeasures, sequential and parallel attack trees, attack protection trees [5, 6, 8, 9, 10] etc. However, modern asset's operations (e.g. autonomous vehicles) heavily depend on interacting with physical surrounding. For example, an autonomous vehicle can interact with the surrounding objects (i.e. cars, infrastructures, traffic signs) to perform different activities ranging from avoiding collision, energy synchronisation, and identifying danger. Even though these objects are not an integral part of the vehicle, they can be used as an attack vector to launch dangerous attacks such as DoS, message falsification attack, message spoofing [11, 12] etc.

Problem statements. An attack tree is a tree-based formalism inspired by *fault trees* [13], and they are used to model (mostly static) threats. On the other hand, assets nowadays are hybrid [14], integrating the physical world to the digital world. This integration poses a great challenge when applying attack trees for the threat modelling. The *threat environment* is ever-changing due to agents behaviour (e.g. interaction with (sub)systems that are public or/and less secure), and the *vulnerability landscape* is erratic due to infrastructure updates (e.g. interaction with other components to complete a task). If such assets are to be analysed using attack trees, the estimated annotations of the tree (e.g. nodes) needs to be updated regularly to reflect both behaviours.

2. Dynamic attack trees

$(\mathcal{Q}, \mathcal{E})$ is a tree, where $\mathcal{Q} = \{q_0\} \cup \mathcal{Q}_S \cup \mathcal{Q}_L$ is the sets of nodes in the tree. $q_0 \in \mathcal{Q}$ denotes its root, $\mathcal{Q}_S$ denotes set of internal nodes (sometimes refer to as sub-goals), and $\mathcal{Q}_L$ denotes set of leaf nodes. These sets of nodes are connected by a set of edges $\mathcal{E} \subseteq \mathcal{Q} \times \mathcal{Q}$. Each node in the attack tree (except for leaf nodes) has a gate refinement. A gate indicates (the fashion) how a node can be achieved (compromised). The interpretation of this fashion is on a node at a level above the current node (parent node). Two of the widely used gates refinement are the AND and OR gates. The AND gate required that all the child nodes linked to a parent node must be compromised before the parent node is reached, while an OR gate can be reached if any

of the child nodes can be compromised. Formally we associate gates to nodes by the mapping $\mathcal{G} : \{q_0\} \cup \mathcal{Q}_S \rightarrow \{AND, OR\}$.

We assume a set $\lambda = \{\iota_1, ..., \iota_n\}$ of external systems/objects from the physical environment (i.e. IoT devices, traffic signs etc.) can communicate with an asset (for example, when they are within proximity via sensors). Intuitively, the sets of nodes in an attack tree represents areas/components of an asset that a malicious user can compromise to attain a malicious goal, this set of objects can also serve as attack vector to any asset they can interact with. For example, fake traffic signs can be malicious to vehicles.

Example 1. *Shown in Figure 1 (a) is a simple tree describing how to steal a car. This can be done by either obtain key or short-circuit. Now let us focus on obtain key sub-goal. It can be achieved through steal key or make copy. However, vehicles nowadays can be operated with external devices such as remote starter and mobile phones to unlock doors, and start the engine.* δ_0, δ_1 *are time points (explain later) that shows default tree and when an external device is added respectively.*

Now, before we formulate how this new nodes can be added to the tree, we first assume that the threat environment of an asset can change due to agent(s) behaviour. Informally, a threat environment is a state of operation which exposed an asset to a (new) set of vulnerabilities and, these vulnerabilities can only be exploited while the asset remains in that state. For example, Figure 1 (a) shows how to steal a car while it is parked. A change in the threat environment e.g the car starts moving, an attacker can steal the car via other means i.e shown in Figure 1 (b).

Let $\Delta = \{\delta_0, \delta_1, ..., \delta_h\}$ be a set of (analysis) time points. We will use this set of time points to mark when a new vulnerability emerged in the asset (i.e. external device connect, disconnect).

Definition 1. *Let* $\{q_0\} \cup \mathcal{Q}_S$ *be a root node and a set of internal nodes in an attack tree,* λ *be a set of external objects from the physical environment, and* Δ *be a set of analysis time points. A connection between the nodes in the tree and external objects is given as a function mapping* $f : (\{q_0\} \cup \mathcal{Q}_S) \times \lambda \times \mathbb{R}^+ \rightarrow \{0, 1\}$ *and* $f(q, \iota, \delta_i) = f(q, \iota, r)$ *for* $\delta_i \leq r < \delta_{i+1}$*, for* $0 \leq i < h$*, i.e. value of* f *changes with time points from* Δ*.*

If $f(q, \iota, \delta) = 1$, for simplicity we write $q_\delta(\iota)$, interpreting as connection via an intermediate node $q \in \mathcal{Q}_S$ of the tree with an external system $\iota \in \lambda$ and δ is the time point at which the connection occur. And if $f(q, \iota, \delta) = 0$, we simply write q_δ to denote that the node (sub-goal) does not connect with any external system/object. In a case of $q_\delta(\iota)$, the internal node q is said to include ι as a child (leaf) node (an edge linking the sub-goal of the asset with external system/object). The attack tree in Figure 1 (a) shows some external devices which can connect to the sub-goal and be used as attack vectors.

Example 2. *Let's refer back to Figure 1 (a), the analysis time points* δ_0 *and* δ_1 *indicate when the external device(s) shown in dotted lines connects to the car. As such, become possible ways (when exploited) to reach the root node. At* δ_0*, there is no connected objects to the car.*

Definition 2. *A dynamic attack tree is a tuple* $\mathcal{T}_r = (\mathcal{Q}, q_0, \mathcal{E}, \mathcal{G}, \Delta, \Lambda, f)$ *where,* $\mathcal{Q}$ *is a set of potential nodes that always create a tree,* q_0 *is a fixed root node that represents the attacker's target,* $\mathcal{E} \subseteq \mathcal{Q} \times \mathcal{Q}$ *is a set of potential edges that linked the nodes,* $\mathcal{G} : \{q_0\} \cup \mathcal{Q}_S \rightarrow \{AND, OR\}$ *is a mapping that associate the root node and set of sub-goals with the gates refinement,* Δ *is a set of*

analysis time points for the connections, Λ is a set of objects from the physical environment, and f is a function such that for each time point $\delta \in \Delta$ and each device $\iota \in \Lambda$, the sub-tree of $(\mathcal{Q}, \mathcal{E})$ consisting of 1.) all the nodes $q \in \mathcal{Q}_S$ such that $f(q, \iota, \delta) = 1$ and, 2.) all the leaves to which these are connected (together with the corresponding restriction of $\mathcal{E}$) still forms a tree, 3.) attack tree at time point δ_i will be denoted as a threat environment $\mathcal{T}_i$.

Example 3. *Given a set of threat environment $\{\mathcal{T}_1, \mathcal{T}_2..., \mathcal{T}_n\}$, notice that the attack paths in each threat environment differs. For instance, Obtain key or short-circuit are set of leaf nodes that represents atomic actions to steal a car while it is parked, these same vulnerabilities cannot be exploited in another threat environment (e.g moving).*

Now, let Act be a set of attack actions, an attack actions are set of efforts that a malicious user can perform in order to compromise vulnerabilities. Let $\mathcal{Q}_L$ be a set of leaf nodes that represent vulnerabilities/attack paths. We define an attack as a function $\mathcal{A} : (\mathcal{Q}_L \times \Delta) \rightarrow Act \cup \{Nil\}$, associating leaf nodes in the tree at a given (analysis) time point with a set of actions or *Nil* (when no action). An attack will not succeed if an attacker cannot complete executing the attack actions and the threat environment changed. Suppose an attacker fails to complete an attack process while the asset is at a particular threat environment $\mathcal{T}_i$ (i.e. car parked), any change in the threat environment to $\mathcal{T}_{i+1}$ (i.e. car moving) means that, the attack process started at previous threat environment $\mathcal{T}_i$ will be terminated since the vulnerabilities are no longer in existence at threat environment $\mathcal{T}_{i+1}$.

3. Dynamic attack trees analysis

The Work in [15] introduced how attack trees can be translated into a network of parallel timed automata (TA), with each automaton representing a linear path from a leaf node to the root node. The sets of nodes in the tree represents a set of states in the timed automata, the root node represents a final state, the set of edges represents a set of transition relations, and the set of attack actions represents a set of events. There exists an integrated formal verification tool UPPAAL [16], that accepts a version of TA as its formal language. Using UPPAAL, we can check whether an attacker $\mathcal{B}$ can reach the root node (in other words, whether the asset can be compromised) by the means of reachability checking in the TA. We can also perform other queries to see if the tree/attacker satisfies some properties such as; if an attacker $\mathcal{B}$ will always/eventually reach the root node when the asset is at a particular threat environment or when some external device is connected. This attacker $\mathcal{B}$ can also be modelled as a (separate) timed automaton. In our previous work [10], we demonstrated how attack trees can be translated into a network of parallel timed automata. Given a dynamic attack tree $\mathcal{T}_r$, a set of threat environment $\{\mathcal{T}_1, ..., \mathcal{T}_n\}$, and an attacker $\mathcal{B}$. The root node of the tree is reachable by $\mathcal{B}$ iff corresponding TA plus timed automaton running with the attacker can reach the final location.

Conclusion and future work. The work described here is still in-progress, additional work is being conducted. An attack tree for a big and complex asset is usually very big; adding set of external objects that can be incorporated to the tree over time adds more complexity and number of states. In our next work, we will formally define, and extend dynamic attack trees

translate that will include multiple threat environments. We will model (using UPPAAL) and analyse some working projects.

References

[1] F. Arnold, H. Hermanns, R. Pulungan, M. Stoelinga, Time-dependent analysis of attacks, in: International Conference on Principles of Security and Trust, Springer, 2014, pp. 285–305.

[2] A. Roy, D. S. Kim, K. S. Trivedi, Cyber security analysis using attack countermeasure trees, in: Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research, 2010, pp. 1–4.

[3] B. Schneier, Attack trees, Dr. Dobb's journal 24 (1999) 21–29.

[4] V. Saini, Q. Duan, V. Paruchuri, Threat modeling using attack trees, Journal of Computing Sciences in Colleges 23 (2008) 124–131.

[5] M. Fraile, M. Ford, O. Gadyatskaya, R. Kumar, M. Stoelinga, R. Trujillo-Rasua, Using attack-defense trees to analyze threats and countermeasures in an atm: a case study, in: IFIP Working Conference on The Practice of Enterprise Modeling, Springer, 2016, pp. 326–334.

[6] C.-W. Ten, C.-C. Liu, M. Govindarasu, Vulnerability assessment of cybersecurity for scada systems using attack trees, in: 2007 IEEE Power Engineering Society General Meeting, IEEE, 2007, pp. 1–8.

[7] M. A. Siddiqi, R. M. Seepers, M. Hamad, V. Prevelakis, C. Strydis, Attack-tree-based threat modeling of medical implants., in: PROOFS@ CHES, 2018, pp. 32–49.

[8] F. Arnold, D. Guck, R. Kumar, M. Stoelinga, Sequential and parallel attack tree modelling, in: International Conference on Computer Safety, Reliability, and Security, Springer, 2014, pp. 291–299.

[9] A. T. Ali, D. P. Gruska, Attack protection tree., in: CS&P, 2019.

[10] A. T. Ali, Simplified timed attack trees., in: RCIS, 2021, pp. 653–660.

[11] Y.-C. Sun, G.-H. Yang, Event-triggered resilient control for cyber-physical systems under asynchronous dos attacks, Information Sciences 465 (2018) 340–352.

[12] B. Nassi, M. Sror, I. Lavi, Y. Meidan, A. Shabtai, Y. Elovici, Piping botnet-turning green technology into a water disaster, arXiv preprint arXiv:1808.02131 (2018).

[13] P. J. Brooke, R. F. Paige, Fault trees for security system design and analysis, Computers & Security 22 (2003) 256–264.

[14] T. A. Henzinger, The theory of hybrid automata, in: Verification of digital and hybrid systems, Springer, 2000, pp. 265–292.

[15] R. Kumar, E. Ruijters, M. Stoelinga, Quantitative attack tree analysis via priced timed automata, in: International Conference on Formal Modeling and Analysis of Timed Systems, Springer, 2015, pp. 156–171.

[16] J. Bengtsson, K. Larsen, F. Larsson, P. Pettersson, W. Yi, Uppaal—a tool suite for automatic verification of real-time systems, in: International hybrid systems workshop, Springer, 1995, pp. 232–243.