Towards Analysis of Sophisticated Attacks, with Conditional Probability, Genetic Algorithm and a Crime Function

Wolfgang Boehmer

Towards analysis of sophisticated attacks, with conditional probability, genetic algorithm and a crime function

Wolfgang Boehmer

Technische Universität Darmstadt, Morneweg Str. 30,
CASED building, D-64293 Darmstadt, Germany
wboehmer@cdc.informatik.tu-darmstadt.de

Abstract. In this short article, a proposal to simulate a sophisticated attack on a technical infrastructure is discussed. Attacks on (critical) infrastructures can be modeled with attack trees, but regular (normal) attack trees have some limitation in the case of a sophisticated attack like an advanced persistent (sophisticated) attack. Furthermore, attacks can also be simulated to understand the type of attack, and in order to subsequently develop targeted countermeasures. In this case, a normal, and also a sophisticated attack, is typically carried out in three phases. In the first phase (I) extensive information is gathered about the target object. In the second phase (II), the existing information is verified with a target object scan. In the third phase (III), the actual attack takes place. A normal attack tree is not able to explain this kind of attack behavior. So, we advanced a normal attack tree, which uses conditional probability according to Bayes to go through a certain path - step by step - from the leaf to the root. The learning ability, which typically precedes an attack (phase II), is simulated using a genetic algorithm. To determine the attack, we used threat trees and threat actors. Threat actors are weighted by a function that is called criminal energy. In a first step, it proposes three types of threat actors. The vulnerabilities have been identified as examples for a laboratory network.

Keywords: Conditional probability; genetic algorithm; Bayes theorem; attack trees; threat actor; crime function; risk scenario technology

1 Introduction

The challenge of Homeland security has significantly changed during the last decade. One of the new challenges is so called advanced persistent threats (APT). APT's have a unique characteristic and current defense strategies have failed against this type of threat. The current main shortcomings of most common security technology are Network/-Host-based Intrusion Detection Systems and antivirus products. Antivirus products are excluded in this analysis. Intrusion detection Systems (IDS) are based on two typical items:

– *Signature-based* is still the most common technique and focuses on the identification of known patterns.

– *Anomaly-based*, which consists of monitoring system activity to determine whether an observed activity is normal or anomalous, according to a heuristic or statistical analysis.

Finally, a major challenge for current IDS is the limited window of time for which the connection state can be maintained, as all modern IDSs are focused on real-time detection. Only a few products, like ArcSight from Hewlett Packard, go beyond this limitation. Also the knowledge about sophisticated threats is very rare before a sophisticated attack is faced, but we can point out some typically properties, as Gartner and ISACA has published to name few as a representation. An APT can be characterized as a so called *mission impossible*.

1. APTs make frequent use of zero-day exploits or modify/obfuscate known ones and, thus, are able to evade the majority of signature-based end points and network intrusion detection solutions. Also, in general APTs are spread over a wide period of time and, as a result, are often outside the limited detection/correlation window of these systems.
2. APT attackers focus on a specific target and are willing to spend significant time and explore all possible attack paths until they manage to subvert its defense.
3. APTs are able to jump over an air-gap, like in the case of Stuxnet.
4. Based on the analysis of major APT attacks it is evident that some perpetrators are supported by nation-states that have significant enabling capabilities (intelligence collection, manufacturing, covert physical access) for this type of attacks.
5. APTs are highly selective. Only a small and carefully selected number of victims are targeted, usually in nontechnical departments of an organization, as they are less likely to identify and report an attack.

In this contribution we will deal with sophisticated attacks on a technical infrastructure. We will advance the typical attack tree development methodology with the conditional probability to go step by step from the leaf to root of target. The typical binary logic used on an attack tree is not sufficient for that analysis. Also the search phase of an attack is not represented on a normal attack tree. For this search phase we propose a genetic algorithm (GA) to find a solution to this complex search problem. A genetic algorithm is based on the process of evolution by natural selection, which has been observed in nature. With both of these advanced techniques we combine a threat actor with one possible path of an attack.

The rest of the article is organized as follows. In the next section 2 the related work is discussed and the limitation of a normal attack tree. Afterwards, in section 3, the new type of attack tree, and the strengthes and the weaknesses of this new type of attack tree are discussed. The article concludes with a brief summary, continuing considerations and proposals for further studies.

2 Related work

The idea of attack trees goes back to the article by Weis in 1991 [1]. In this article he describes *threat logical trees*. Generally, attacks are modeled with graphical, mathematical, decision trees. A few years later, the idea of threat trees was taken up by B.

Schneier, among others, and developed [2]. This work by B. Schneier led to extensive additions and improvements to this technology, such as those published by A.P. Moore et al. [3]. Several tools have been developed and published; representative of the work is [5, 6]; the authors provide an overview of the techniques and tools. The contribution of S. Mauw and M. Oostdijk [10] in 2005 formalizes the concepts informally introduced by Schneier [2]. This formalization clarifies which manipulations of attack trees are allowed under which conditions. The commonality of attack trees and game theory has been elaborated on in 2010 by Kordy et al. [11]. Thus, a similar approach for the threat scenarios and threat agent are performed using the game theory.

But all the variations of an attack tree are unable to explain a sophisticated attack, as we explained for APTs in the section 1 (Introduction).

3 A new type of attack tree

We follow the formal definition from [4] of attack graph for a given data structure used to represent all possible attacks.

Definition 3.01 *An attack graph in general, or AG, is a tuple* $G = (S, \tau, S_o, S_s)$*, where* S *is a set of states,* $\tau \subseteq S \times S$ *is a transition relation,* $S_0 \subseteq S$ *is a set of initial states, and* $S_s \subseteq S$ *is a set of success states.*

Based on this definition some automated attack graph generators were developed, to name few [8], [5]. But all of these automated attack graphs are unable to explain the infection path e.g. from the Stuxnet Attack. The infection with Stuxnet Virus was of the internal network via a USB stick. The internal network itself was not connected to the internet. Furthermore we want to use an attack graph to prove the possibility of reaching the target (root) up from a leaf. This possibility is often named as the safety condition. In the following definition we follow [9] page 106.

Definition 3.02 *A probabilistic attack graph (scenario) graph or PAG is a tuple* $G = (S_n, S_p, s_e, \tau, \pi, S_0, S_a, S_f, S, D)$ *where* S_n *is a set of nondeterministic states,* S_p *is a set of probabilistic states,* $s_e \in S_n$ *is a nondeterministic escape state* ($s_e \notin S_a$ *and* $s_e \notin S_f$)*,* $S = S_n \cup S_p$ *is the set of all states,* $\tau \subseteq S \times A \times S$ *is a transition relation,* $\pi : S_p \to S \to \mathcal{R}$ *are transition probabilities,* $S_0 \subseteq$ *is a set of initial states,* $S_a \subseteq S$ *is a set of acceptance states,* $S_f \subseteq S$ *is a set of finial states, and* $D : S \to 2^A (A = 2^{AP})$ *is a labeling of states with sets of alphabet letters.*

A probabilistic scenario graph (PSG) distinguishes between nondeterministic states (set S_n) and probabilistic states (set S_p). The sets of nondeterministic and probabilistic states are disjointed ($S_n \cap S_p = \oslash$). The function π specifies probabilities of transitions from probabilistic states, so that for all transitions (s_1, s_2) such that $s_1 \in S_p$, we have Prob $(s_1 \to s_2) = \pi(s_1)(s_2) > 0$. Thus, $\pi(s)$ can be viewed as a probability distribution on next states. Intuitively, when the system is in a nondeterministic state s_n, we have no information about the relative probabilities of the possible next transitions. When the system is in a probabilistic state s_p, it will choose the next state according to probability distribution (s_p) [9, 12] page 106.

In this work, we expand the idea of T. R. Ingoldsby [6] with conditional probability $\pi(s)$. Threat trees generally have a root or target. Different branches (nodes) can lead to this target, which are to be regarded as parts of goals and each start of a leaf.

Each leaf is initiated by an attacker with different motives. The leaves and branches are weighted and equipped with an actor [14]. The weighting corresponds to the criminal energy (Criminal power, Cp), and contains three functions. The assessment of these three functions reflects the exogenous knowledge which is required in the Bayesian statistics.

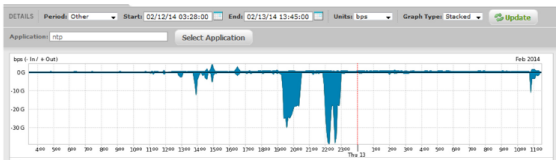

Fig. 1: Attack tree of a telecommunication provider on NTP protocol

The function of the criminal energy (Cp), which in turn is composed of three additive functions, represents the expert knowledge for the Bayesian risk analysis. The criminal energy is represented by the cost function (cost of attack *cf.* Fig. 2), the technical feasibility function (technical function, *cf.* 3) and the noticeability function, *cf.* Fig. 4. The three functions are mentioned by T. R. Ingoldsby [6] and have to be adjusted to the relevant inspection. The following Figures 2 - 4 describe the exogenous knowledge that focuses on the objective of a system.

Fig. 2 states that a threat agent (actor) for an attack is willing to spend money on tools. This willingness varies between 0 - 1 (axis of ordinates) and decreases with increasing costs (axis of abscissae). This can be explained simply because, on the internet, there are a number of free tools that are all well-suited to threaten a system.

Figure 3 indicates how the tools are used and the technical possibilities that exist. It is a statement about the complexity of the tools and the willingness to make use of this complexity. The curve indicates that the greater the technical possibilities and the complexity of the tools, the more the willingness drops. I.e. simple tools are preferred with simple operation. Figure 4 shows the noticeability function expressing how an actor wants to disguise his attack, so that he could not be discovered. From of these three functions, the threat of an attack is determined more precisely. This is called the criminal energy. These functions must be adapted to each situation and reflect the exogenous knowledge again, which is necessary for the conditional probability.

As an example, we can estimate the technical ability rating of 5 (out of 10), and expose the miscreant at a 0.3 noticeability.

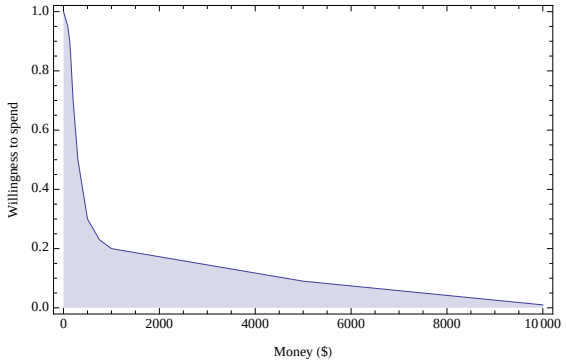

Fig. 2: Cost function

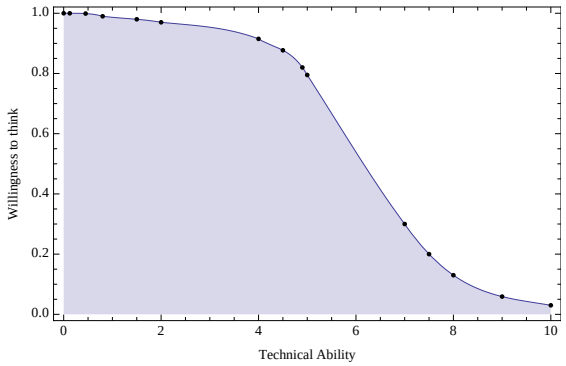

Fig. 3: Technical function

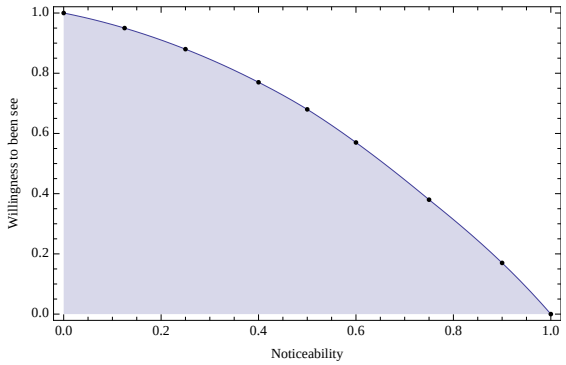

Fig. 4: Noticeability function

Using the utility functions shown, we discover that

$$f_{cost}\,(25) = 0.9 \tag{3.1}$$

$$f_{tech\,ability}\,(05) = 0.9 \tag{3.2}$$

$$f_{noticeabiity}\,(0.3) = 0.85 \tag{3.3}$$

and therefore the criminal energy with

$$CE = f_{cost} \cdot f_{tech\,ability} \cdot f_{noticeabiity} \tag{3.4}$$

$$CE = 0.6885 = 0.9 \cdot 0.9 \cdot 0.85 \tag{3.5}$$

With this function of the criminal energy, we could estimate the threat profile in conjunction with a specific threat agent (actor). The three functions of Figures 2 - 4 do not explain anything about the motivation and benefit of the threat agent, but the threat agent's motivation is correlated with attack benefits. These must also be taken into account in order to understand how desirable an attack appears to an adversary. The discussion of the motivation and benefits of a threat agent is not really covered in this article. The criminal function produces the population of the genetic algorithm.

The basic process for a genetic algorithm [13] is:

- Initialization - Create an initial population. This population is usually randomly generated and can be any desired size, from only a few individuals to thousands.
- Evaluation - Each member of the population is then evaluated and we calculate a 'fitness' for that individual. The fitness value is calculated by how well it fits with our desired requirements. These requirements could be simple, 'faster algorithms are better', or more complex, 'stronger materials are better but they shouldn't be too heavy'.
- Selection - We want to be constantly improving our populations overall fitness. Selection helps us do this by discarding the bad designs and only keeping the best individuals in the population. There are a few different selection methods, but the basic idea is the same: make it more likely that fitter individuals will be selected for our next generation.
- Crossover - During crossover we create new individuals by combining aspects of our selected individuals. We can think of this as mimicking how sex works in nature. The hope is that by combining certain traits from two or more individuals we will create an even 'fitter' offspring which will inherit the best traits from each of its parents.
- Mutation - We need to add a little bit randomness into our populations' genetics otherwise every combination of solutions we can create would be in our initial population. Mutation typically works by making very small changes at random to an individuals genome.
- Repeat all steps - Now we have our next generation we can start again from step two until we reach a termination condition.

But, typically the largest benefit of the threat agent is associated with achieving the tree's root node, or with side benefits occurring at the various intermediate nodes. Different threat scenarios run through different paths among leaf nodes and root node.

The threat agent's benefits may differ considerably depending on the threat scenario used. We will discuss different threat scenarios and different paths between leaf nodes and root in the next section.

4 Conclusion and further investigation

A comparison with real world attacks, like Stuxnet, shows that typical modeled attack graphs and their models in the literature are not able to create a real attack graph. In this paper we proposed a new methodology to use a genetic algorithm and the Bayes theorem to create an attack graph that could explain a real attack graph. The next step is to include not only a network into the model, but also the environment must be taken into account.

References

[1] J. D. Weis. A system security engineering process. Proceedings of the 14th National. Computer Security Conference, 1991.

[2] B. Schneier, Attack trees, Dr. Dobb' s Journal, vol. 24, no. 12, pp. 21– 29, 1999.

[3] A. P. Moore, R. J. Ellison, and R. C. Linger, Attack modeling for information security and survivability, Technical Note CMU/SEI-2001- TN-001, Carnegie Mellon University, 2001.

[4] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. Wing. Automated generation and analysis of attack graphs. In Security and Privacy, 2002. Proceedings. 2002 IEEE Symposium on, pages 273?284, 2002.

[5] O. Sheyner and J. Wing, Tools for Generating and Anaylzing Attack Graphs, pp. 344 – 371. FMCO 2003, LNCS 3188, Springer-Verlag Berlin Heidelberg, 2004.

[6] T. R. Ingoldsby, Fundamentals of Capabilities-based Attack Tree Analysis. Amenaza Technologies Limited, 406 – 917 85th St SW, m/s 125.

[7] S. Jha, O. Sheyner, and J. Wing. Two formal analyses of attack graphs. In Computer Security Foundations Workshop, 2002. Proceedings. 15th IEEE, pages 49–63, 2002.

[8] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. Wing. Automated generation and analysis of attack graphs. In Security and Privacy, 2002. Proceedings. 2002 IEEE Symposium on, pages 273 - 284, 2002.

[9] O. M. Sheyner. Scenario graphs and attack graphs. PhD thesis, University of Wisconsin, 2004.

[10] S. Mauw and M. Oostdijk, Foundations of attack trees, in International Conference on Information Security and Cryptology – ICISC 2005. LNCS 3935, pp. 186 – 198, Springer, 2005.

[11] B. Kordy, S. Mauw, M. Melissen, and P. Schweitzer, Attack-defense trees and two-player binary zero-sum extensive form games are equivalent, in Proceedings of the First international conference on Decision and game theory for security, GameSec'10, (Berlin, Heidelberg), pp. 245 – 256, Springer-Verlag, 2010.

[12] L. Wang, T. Islam, T. Long, A. Singhal, and S. Jajodia. An attack graph-based probabilistic security metric. In Proceeedings of the 22Nd Annual IFIP WG 11.3 Working Conference on Data and Applications Security, pages 283?296, Berlin, Heidelberg, 2008. Springer-Verlag.

[13] Darrel Whitley, A genetic algorithm tutorial, Statistics and Computing, June 194, Volume 4, Issue 2, pp65-85.

[14] N. Poolsappasit, Towards an Efficient Vulnerability Analysis Methodology for better Security Risk Management. PhD thesis, Colorado State University, July 2010.