Playing Protection Poker for Practical Software Security

Martin Gilje Jaatun and Inger Anne Tøndel

Department of Software Engineering, Safety and Security
SINTEF ICT
NO-7465 Trondheim, Norway
martin.g.jaatun@sintef.no
http://www.sintef.no/sos-agile

Abstract. Software security is about creating software that keeps performing as intended even when exposed to an active attacker. Secure software engineering is thus relevant for all software, not only security software. We describe Protection Poker, a tool for risk estimation to be used as part of the iterationplanning meeting, and discuss some preliminary experiences.

1 Introduction

Protection Poker is a security risk assessment technique for agile development teams proposed by professor Laurie Williams and colleagues at NCSU [1]. The idea is to play Protection Poker as part of every iteration[1] planning meeting, in order to rank the security risk of each feature to be implemented in that iteration, and possibly identify additional security mechanisms that have to be implemented to maintain an acceptable risk level.

1.1 Risk in Protection Poker

Protection Poker uses a slight variation of the traditional computation of risk:

$$risk = (\sum \text{value of assets that could be exploited}) \times (\text{the exposure}) \qquad (1)$$

Risk is always related to the requirements that are to be implemented in the next iteration, often this will be some new, enhanced or corrected functionality. Exposure relates to how hard or easy this change in functionality makes it to attack the system, and in the evaluation of exposure, one should consider the possible ways in which attackers can attack the system (attack surface), what type of breaches they can perform (confidentiality, integrity, availability) and the skill level required. For asset value, the value of the asset for various groups should be considered: the value of the asset for an attacker is important for attacker motivation, whereas the value of the asset for customers, users, the

[1] e.g., a "sprint" in Scrum

business, etc. highly determines the consequences that a successful attack may have. Assets are typically considered to be database tables or system processes that the new functionality controls.

Our dialect of Protection Poker[2] uses the following numbers to determine asset value or system exposure: <10, 20, 30, 40, 50, 60, 70, 80, 90, 100. In calculations, "<10" is counted as 10. To be able to prioritize between different requirements, it is important to be able to get a spread in the numbers assigned. This is to avoid that, e.g., high risk projects rate every requirement with a high number. That would make it very hard to prioritize within the project.

1.2 Calibration

With Protection Poker, the security risk of a requirement is compared to other requirements of the same system. The goal is not to establish a "perfect" and "universal" risk value, but to rate the security risk of the requirements in order to be able to better prioritize security effort. Before starting to play Protection Poker for a system, it is thus recommended to perform calibration in order to arrive at a common understanding of the end-points of the scale, i.e., what does a <10 or a 100 mean for this product? This is done the following way:

Asset value: The team asks itself: what assets are most important in this system, and what assets are of little value. The asset they can think of as most important is given a '100' and the asset they can think of with little value is given a '<10'.

Exposure: The team asks itself: what types of functional requirements can open up most for attacks, and which functional requirements can limit exposure, and assign a '100' and a '<10' accordingly.

In the evaluation of asset value and exposure, numbers should be assigned relative to these endpoints, as well as the values assigned for previously assessed requirements.

2 Playing the game

Protection Poker is played during an iteration planning meeting, and it is recommended that the full development team participates. One person should have the role as moderator, and this person will be responsible for leading the team through the game, and point the discussions in a good direction. Ideally, a separate person should be tasked with recording important security solutions and ideas that emerge during play. Focus is on the specific requirements the team will likely implement during the next iteration.

Step 1 – Common understanding of the requirements: The requirements to be implemented in the iteration are explained to the team (e.g., by the product manager or business owner) and the team members discuss or ask questions to clarify the requirements.

[2] `http://www.sintef.no/protection-poker`

Step 2 – Initial discussion of security implications: The team performs a first discussion of the security implications of the requirements. The moderator can ask leading questions, e.g., “Who would want to attack this system?”; “What would an attacker do if he got access to this data?”; or “What damage could an insider do with this functionality?”

Step 3 – Identify assets: Everybody together identify which assets are created or touched upon by the requirement under consideration. Some of these may have already been assigned a value, and then this value can be reused.

Step 4 – Assign value to assets: For identified assets that have not previously been assigned a value, each participant picks the Protection Poker card (individually and without telling anybody about which card has been picked) that best describes their understanding of the asset’s value. All participants show their selected card to the whole team, and the team discusses the rationale for selecting the cards; the team members with the highest and lowest cards explain them to the group, followed by an open discussion until the team is ready to revote (if there is disagreement). When the team has reached a consensus on the asset value (or when there is no use in discussing any further – in these cases the moderator is responsible for making a suggestion for what value to assign to the asset), the moderator notes the asset value. The team now moves on to the next asset (if there are more left to assess) or to the exposure evaluation.

Step 5 – Evaluate exposure: As for asset value, the team bids to evaluate to which extent the requirement increases the exposure of the system and assets to attack.

Step 6 – Calculate risk: The numbers assigned for asset values and exposure are used to calculate a risk value as given in Equation 1.

Step 7 – Compare risk related to other requirements: The risk value for the requirement can be compared to the risk value of other requirements to see for which requirements security should be given specific priority.

Step 8 – Prioritize security activities: Based on the risk value and the discussion decision should be made on how to address security for this requirement. The decision should be documented. If there is a need for specific security activities or functionalities, these should be documented together with other requirements (e.g. in the backlog).

3 Experiences and Challenges

We made some small adjustments to the Protection Poker cards and terminology. Whereas the original Protection Poker uses the term “ease of exploitation”, we found that this concept was distracting or not properly understood by some pilot players, e.g., leading them to focus too much on threats such as “shoulder surfing”. In order to focus more on how a feature increases the attack surface of an application, we decided to change it to “exposure”.

The original Protection Poker[3] uses the same cards as Planning Poker [2, 3] (also known as Scrum Poker), used for effort estimation in agile teams. Planning Poker cards follow a Fibonacci-like sequence, after the rationale that it is easier to have an opinion on whether a task takes 1 or 2 days than whether it takes 40 or 41 days. We argue, however that the same is not true when it comes to relative value of assets or degree of exposure, and since we are less concerned about small risks and more interested in the bigger risks, we opted for an even scale instead. This enables us to differentiate between big risks, not just the small ones.

We have tried out Protection Poker with representatives from various Norwegian organizations and in general it has been well received. However, some have indicated that they feel it takes too long to play the game, especially when considering that planning meetings already are perceived as being too full. Laurie Williams [1] found that the time required for playing dropped significantly after the team gained familiarity with the technique, but we need more experience with our partner companies to determine whether that will also be the case here. We remain open to the possibility of changing how and when Protection Poker is played in order to maximise the benefit.

For some development groups, we have observed that asset identification can be difficult, and particularly the *granularity* of assets can be challenging. It is important that at least within a development team, the assets have a consistent granularity, as this otherwise might skew the risk calculations.

Acknowledgment

This work was supported by the SoS-Agile: Science of Security in Agile Software Development project, funded by the Research Council of Norway, grant number 247678.

References

1. Williams, L., Meneely, A., Shipley, G.: Protection poker: The new software security game. IEEE Security and Privacy **8**(3) (2010) 14–20
2. Grenning, J.: Planning poker or how to avoid analysis paralysis while release planning. Hawthorn Woods: Renaissance Software Consulting **3** (2002)
3. Moløkken-Østvold, K., Haugen, N.C., Benestad, H.C.: Using planning poker for combining expert estimates in software projects. Journal of Systems and Software **81**(12) (2008) 2106 – 2117 Best papers from the 2007 Australian Software Engineering Conference (ASWEC 2007), Melbourne, Australia, April 10-13, 2007Australian Software Engineering Conference 2007.

[3] `http://collaboration.csc.ncsu.edu/laurie/Security/ProtectionPoker/`