Tool-supported premortems with attack and security patterns.

FAILY, S., LYLE, J. and PARKIN, S.

2012

Tool-supported Premortems with Attack and Security Patterns

Shamal Faily *, John Lyle * and Simon Parkin †
* Department of Computer Science, University of Oxford. *first.last@cs.ox.ac.uk*
† School of Computing Science, Newcastle University. *s.e.parkin@ncl.ac.uk*

***Abstract*— Security patterns are a useful technique for packaging and applying security knowledge. However, because patterns represent partial knowledge of a problem and solution space, there is little certainty that addressing the consequences of one problem won't introduce or exacerbate another. In this abstract, we suggest that rather than using patterns exclusively to explore possible solutions to security problems, we should use them to carry out a *premortem* on why they instead cause problems. We present the approach taken to devise and tool-support such a process using data from the EU FP 7 *webinos* project.**

***Index Terms*—security pattern; attack pattern ; premortem ; CAIRIS**

I. Motivation

Because security knowledge isn't always readily available in design situations, there is value in codifying and packaging it. Given both the adversarial nature of security, and the resulting dangers of over or under commensurate treatment of security, it seems useful to package knowledge about attacks as patterns as well. It is surprising, therefore, that, despite an abundance of examples of how security knowledge can be codified as patterns, e.g. [1], and the claim that building attack patterns is evidence of organisational security maturity [2], there is a dearth of work describing the application of attack patterns in security design.

While attempts have been made to characterise attack and misuse patterns, e.g. [3], such patterns may not be effective until we understand the abstractions used by attackers, as well as defenders, to reason about a system. One way of tackling this problem involves getting a better understanding of the attackers themselves. Steps towards this goal are being made using profiling techniques [4], and the reuse of open-source intelligence for building attacker personas [5].

Like patterns in general, attacker representations can only provide a partial representation of an attacker's knowledge; if these are to act as an impetus for motivating attack patterns then the qualitative data used to build these personas needs to be relevant to the design context. For example, if we develop personas for the possible attackers of an online book-store, there is no certainty that these personas will be equally useful when considering the potential attackers of an electronic voting system. Moreover, given that recent work has shown that the data used to develop personas can be useful for informing secure system design in its own right [6], one might argue that if context specific qualitative data was readily available then we would simply use it to identify criteria for selecting a security pattern, thereby eliminating the need for attacker representations, and attack patterns in general.

To better appreciate the value that attack patterns might have in design, we need to consider security as, what social planners call, a *wicked problem* [7].

II. Patterns as an exploratory tool

Security can be considered an example of a wicked problem because we lack clarity about what it means to secure systems, tests for proving a system is secure, and a grasp of all possible solutions for satisfying a specified security problem [8]. Making any design decision has consequences on the underlying system. This makes security patterns useful because pattern templates describe the consequences of their use. This is important because the wicked nature of security means that we may never have the assurances that we would like about a pattern's efficacy; while a pattern may be one possible solution to a problem, we can never be completely sure that this solution itself doesn't introduce complications yet to be identified. Nonetheless, applying security patterns remains useful because, as designers, they force us to make value judgements about possible design solutions, and these help us delimit the solution space.

Interestingly, the value associated with applying patterns to delimit the problem space is obtained whether or not they successfully address the problem we had in mind. While it seems paradoxical that we would apply a security pattern knowing that it will fail, the value the failure provides in delimiting the problem space is arguably greater than its success. This is because analysing the failure may lead to more reflection about why the failure occurred so that subsequent candidate solutions can avoid any identified pitfalls. Such an approach is analogous to a *premortem*. In business scenario planning, these operate on the assumption that a solution has failed; rather than reflecting on what may go wrong with a design, planners instead generate plausible reasons for explaining why a solution has already failed [9]. Although the known structure, motivation, and consequences of security patterns provide some insight into the causes of such a failure, when combined with attack patterns, they allow reflection on the motivations of a perceived attacker, and how his capabilities and motivations lead to an exploit identified in a failed security pattern; this can then be considered in subsequent patterns exploring the same problem. If the mapping between patterns is unclear, the lack

```
<?xml version="1.0"?>
<!DOCTYPE attack_pattern SYSTEM "attack_pattern.dtd">

<attack_pattern name="Request footprinting" likelihood="Occasional"
severity="Critical">
  <intent>Glean an understanding of what resources are available on a
device by eavesdropping on requests. </intent>
  <motivation goal="confidentiality" value="Low">
    <description>Ethan wants to get a better understanding of what
resources are under policy control. </description>
  </motivation>
  <applicability environment="Complete" />
  <structure attack="Network Eavesdropping" exploit="Missing XML
Validation" />
  <participant name="Ethan">
    <motive name="System resource theft" />
    <responsibility name="Technology" value="Medium" />
    <responsibility name="Software" value="Medium" />
    <responsibility name="Knowledge/Methods" value="Medium" />
  </participant>
  <collaboration>
    <target name="Access Requestor" />
    <exploit name="Access Request" />
  </collaboration>
  <consequences>Impact of attack</consequences>
  <implementation>This scenario describes how Ethan carries out request
footprinting</implementation>
  <known_uses>None</known_uses>
  <related_patterns>None</related_patterns>
</attack_pattern>
```

Fig. 1. XML document of an attack pattern

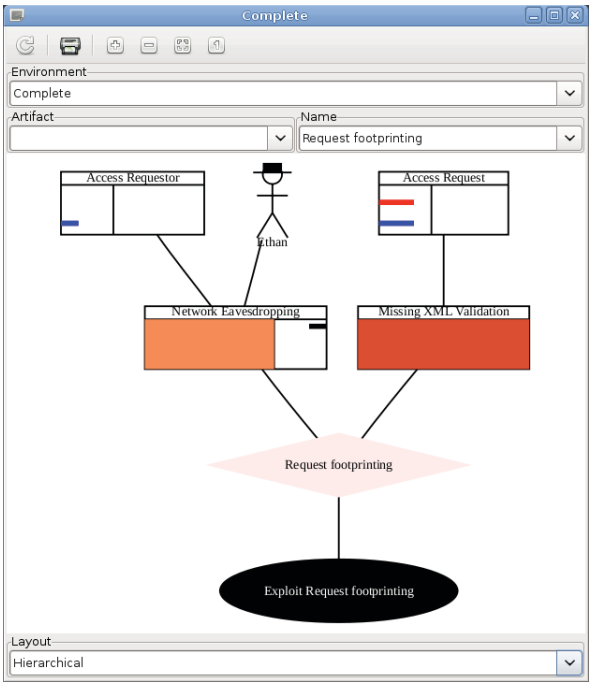

Fig. 2. CAIRIS Risk Analysis model of security and attack pattern elements

of data also provides clues about what additional evidence is needed before a "cause of death" can be established.

III. A TOOL-SUPPORTED PREMORTUM PROCESS

At the University of Oxford, we have explored how such a premortem process might be tool-supported. Using the EU FP 7 *webinos* project as an exemplar, we have imported project requirements, use cases, personas, and open-source threat data from the OWASP [10] project into the open-source CAIRIS design tool [11]. Using the canonical Design Patterns template prescribed by [12], we concurrently specified security and attack patterns that were relevant to webinos in XML documents; an example of the template used for attack patterns is illustrated in Figure 1. Each element of the security and attack patterns was aligned with elements of the IRIS meta-model [13], upon which CAIRIS was built. Once the patterns were created, we first imported the relevant attack patterns into the tool before introducing a security pattern we wish to analyse into a CAIRIS model. In addition to generating a risk analysis model, such as that illustrated in Figure 2, extensions to CAIRIS were also added to automate an attack resistance analysis. This form of analysis was proposed by McGraw [14] as part of an architectural risk analysis process but, instead of using it to demonstrate the viability of known attacks against a security pattern, we instead used the technique to understand why the security pattern failed to mitigate the attack pattern.

We are currently evaluating both this process and the tool-support by using it to support the design of the security architecture for *webinos*.

IV. ACKNOWLEDGEMENT

The research described in this paper was funded by the EU FP7 *webinos* project (FP7-ICT-2009-05 Objective 1.2).

REFERENCES

[1] M. Schumacher, E. Fernandez, D. Hybertson, and F. Buschmann, *Security Patterns: Integrating Security and Systems Engineering*. John Wiley & Sons, 2005.

[2] G. McGraw, B. Chess, and S. Migues, "Building security in maturity model," September 2011.

[3] E. B. Fernandez, N. Yoshioka, H. Washizaki, J. Jürjens, M. VanHilst, and G. Pernul, "Using security patterns to develop secure systems," in *Software Engineering for Secure Systems: Industrial and Research Perspectives*, H. Mouratidis, Ed. IGI Global, 2011, ch. 2, pp. 16–31.

[4] R. Chiesa, S. Ducci, and S. Ciappi, *Profiling hackers: the science of criminal profiling as applied to the world of hacking*. Auerbach Publications, 2009.

[5] A. Atzeni, C. Cameroni, S. Faily, J. Lyle, and I. Fléchais, "Here's Johnny: a Methodology for Developing Attacker Personas," in *Proceedings of the 6th International Conference on Availability, Reliability and Security*, 2011, pp. 722–727.

[6] S. Faily and I. Fléchais, "User-centered information security policy development in a post-stuxnet world," in *Proceedings of the 6th International Conference on Availability, Reliability and Security*, 2011, pp. 716–721.

[7] H. W. J. Rittel and M. M. Webber, "Dilemmas in a general theory of planning," *Policy Sciences*, vol. 4, no. 2, pp. 155–169, 1973.

[8] S. Faily and I. Fléchais, "To boldly go where invention isn't secure: applying Security Entrepreneurship to secure systems design," in *Proceedings of the 2010 New Security Paradigms Workshop*. ACM, 2010, pp. 73–84.

[9] G. Klein, "Performing a project premortem," *Harvard Business Review*, vol. 85, no. 9, pp. 18–19, 2007.

[10] Anonymous, "Open Web Application Project (OWASP) web site," http://www.owasp.org, August 2011.

[11] S. Faily, "CAIRIS web site," http://www.comlab.ox.ac.uk/cairis, June 2011.

[12] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design patterns: elements of reusable object-oriented software*. Addison-Wesley, 1995.

[13] S. Faily and I. Fléchais, "A Meta-Model for Usable Secure Requirements Engineering," in *Proceedings of the 6th International Workshop on Software Engineering for Secure Systems*. IEEE Computer Society, 2010, pp. 126–135.

[14] G. McGraw, *Software Security: Building Security In*. Addison-Wesley, 2006.