Poster: Quantitative Underpinnings of Secure, Graceful Degradation

Ryan Wagner
Carnegie Mellon University
rrwagner@cs.cmu.edu

David Garlan
Carnegie Mellon University
garlan@cs.cmu.edu

Matt Fredrikson
Carnegie Mellon University
mfredrik@cs.cmu.edu

ABSTRACT

System administrators are slowly coming to accept that nearly all systems are vulnerable and many should be assumed to be compromised. Rather than preventing all vulnerabilities in complex systems, the approach is changing to protecting systems under the assumption that they are already under attack.

Administrators do not know all the latent vulnerabilities in the systems they are charged with protecting. This work builds on prior approaches that assume more *a priori* knowledge. [5]. Additionally, prior research does not necessarily guide administrators to gracefully degrade systems in response to threats [4]. Sophisticated attackers with high levels of resources, like advanced persistent threats (APTs), might use zero day exploits against novel vulnerabilities or be slow and stealthy to evade initial lines of detection.

However, defenders often have some knowledge of where attackers are. Additionally, it is possible to reasonably bound attacker resourcing. Exploits have a cost to create [1], and even the most sophisticated attacks use limited number of zero day exploits [3].

However, defenders need a way to reason about and react to the impact of an attacker with existing presence in a system. It may not be possible to maintain one hundred percent of the system's original utility; instead, the attacker might need to gracefully degrade the system, trading off some functional utility to keep an attacker away from the most critical functionality.

We propose a method to "think like an attacker" to evaluate architectures and alternatives in response to knowledge of attacker presence. For each considered alternative architecture, our approach determines the types of exploits an attacker would need to achieve particular attacks using the Datalog declarative logic programming language in a fashion that draws adapts others' prior work [2][4]. With knowledge of how difficult particular exploits are to create, we can approximate the cost to an attacker of a particular attack trace. A bounded search of traces within a limited cost provides a set of hypothetical attacks for a given architecture. These attacks have varying impacts to the system's ability to achieve its functions. Using this knowledge, our approach outputs an architectural alternative that optimally balances keeping an attacker away from critical functionality while preserving that functionality. In the process, it provides evidence in the form of hypothetical attack traces that can be used to explain the reasoning.

This thinking enables a defender to reason about how potential defensive tactics could close off avenues of attack or perhaps enable an ongoing attack. By thinking at the level of architecture, we avoid assumptions of knowledge of specific vulnerabilities. This enables reasoning in a highly uncertain domain.

We applied this to several small systems at varying levels of abstraction. These systems were chosen as exemplars of various "best practices" to see if the approach could quantitatively validate the underpinnings of general rules of thumb like using perimeter security or trading off resilience for security. Ultimately, our approach successfully places architectural components in places that correspond with current best practices and would be reasonable to system architects. In the process of applying the approach at different levels of abstraction, we were able to fine tune our understanding attacker movement through systems in a way that provides security-appropriate architectures despite poor knowledge of latent vulnerabilities; the result of the fine-tuning is a more granular way to understand and evaluate attacker movement in systems.

Future work will explore ways to enhance performance to this approach so it can provide real time planning to gracefully degrade systems as attacker knowledge is discovered. Additionally, we plan to explore ways to enhance expressiveness to the approach to address additional security related concerns; these might include aspects like timing and further levels of uncertainty.

CCS CONCEPTS

• **Security and privacy → Software and application security**;

KEYWORDS

Self-Adaptive Systems, Security, Advanced Persistent Threat

ACM Reference Format:
Ryan Wagner, David Garlan, and Matt Fredrikson. 2018. Poster: Quantitative Underpinnings of Secure, Graceful Degradation. In *Proceedings of HotSoS conference (HotSos'18).* ACM, New York, NY, USA, Article 4, 1 page. https://doi.org/https://doi.org/10.1145/3190619.3191695

HotSos'18, April 10–11, 2018, Raleigh, NC, USA

ACM ISBN 978-1-4503-6455-3/18/04…$15.00
https://doi.org/https://doi.org/10.1145/3190619.3191695

REFERENCES

[1] 2013. The digital arms trade. *Economist* (30 March 2013).
[2] Stefano Ceri, Georg Gottlob, and Letizia Tanca. 1989. What you always wanted to know about Datalog (and never dared to ask). *IEEE transactions on knowledge and data engineering* 1, 1 (1989), 146–166.
[3] Nicolas Falliere, Liam O Murchu, and Eric Chien. 2011. W32.Stuxnet dossier. *Symantec Corp., Security Response* (Feb. 2011).
[4] Xinming Ou, Sudhakar Govindavajhala, and Andrew W Appel. 2005. MulVAL: A logic-based network security analyzer.. In *USENIX Security Symposium.* Baltimore, MD, 8–8.
[5] Oleg Sheyner and Jeannette Wing. 2003. Tools for generating and analyzing attack graphs. In *International Symposium on Formal Methods for Components and Objects.* Springer, 344–371.