Michael A. Borchardt

Web Application Security: Threat Modeling

March 9th, 2024

Objective and Scope

The purpose of this Web Application Security document is to better communicate the idea behind threat modeling, with context to web applications. In today's online environment, there are many threats a web application administrator may come into contact with while overseeing operations.

Foremost, this document is intended for personnel overseeing, administrating, or developing websites, web applications, or web services. This document is also intended for novice users, who may be curious about threat modeling. I do my best to avoid language that is too technical.

1. Introduction

The materials presented within the contents of this document have been collected from several sources across the Internet. These organizations include, but are not limited to, the Open Web Application Security Project (OWASP), the National Insitute for Standards and Technology (NIST), the SANS Insititute, and other recognized sources of industry best practices.

2. What Is Threat Modeling?

First, the main idea behind threat modeling is to identify vulnerabilities and risks in a web application before an initial production launch. Concerning web applications, threat modeling is mitigating vulnerabilities and risks associated with connecting to an online environment. When threat modeling web applications, the best threat models are generated from assuming the perspective of the attacking party.

There are many bad actors to be encountered when connecting an application to a public domain. While many of the risks associated with a public domain can be mitigated through proper threat modeling, there will, inevitably, be some risk that cannot be mitigated.

3. Three Steps of Threat Modeling.

There is a three-step process concerning threat modeling web applications. Foremost, the first step involves decomposing the application. While many applications are too complex to quickly decompose, this task may require a team of individuals. I understand decomposing the application may be perceived as vague and complex terminology. However, I understand decomposing the application as figuring out how the application interacts with the public environment.

Decomposing the application includes a number of activities, which I would consider essential. First, identify entry points to better understand where a potential attacker could interact with the application. Next, identify assets associated with the application that the attacker would be interested in. For example, some attackers may be interested in data held by an application's storage container.

The second step involved in constructing an appropriate threat model is to determine and rank threats. However, this may appear to be a difficult task. When ranking threats, an organization must consider which assets are most valuable. The most valuable assets have the highest-ranking threats.

The third step in constructing a threat model is to determine the countermeasures needed to mitigate the highest-ranking threats. Oftentimes, an organization will consider a cost-benefit analysis methodology before implementing mitigation strategies. How valuable is the asset they're trying to protect, and is the cost of protecting the asset worth the money? These are

questions best left to higher-ranking administrative staff members.

Depending on the web application, website, or web service, business continuity may not be entirely dependent upon the uptime of the public domain. In some scenarios, the business is a public web application, and business continuity is entirely dependent upon the use of the service. In such scenarios, guaranteeing server uptime is of mission-critical importance.

Good threat modeling makes use of appropriate countermeasures, as a preventative effort, to stop bad actors from impacting web application functionality and continuity.