# Poster: Advanced Metrics for Risk-Based Attack Surface Approximation

Christopher Theisen
North Carolina State University
Department of Computer Science
890 Oval Drive, Campus Box 8206 Raleigh, NC 27695
crtheise@ncsu.edu

Laurie Williams
North Carolina State University
Department of Computer Science
890 Oval Drive, Campus Box 8206 Raleigh, NC 27695
lawilli3@ncsu.edu

## ABSTRACT
Despite a growing number of threats, the software engineering community still faces a critical deficit of trained security professionals for defending against cyber attacks. To combat this shortage, efficient prioritization of the effort of security professionals is needed. To address this issue, we present Risk Based Attack Surface Approximation (RASA), which uses crash dump stack traces to approximate the attack surface of a system. In our poster, we will describe several RASA metrics that could help security effort prioritization. These metrics include temporal metrics (how the attack surface changes over time), shape metrics (how code artifacts are interconnected), and depth metrics (how far code artifacts are from the entry and exit points of a system).

## CCS Concepts
D.2.8 [**Software Engineering**]: Metrics – *Process Metrics*, *Product Metrics.*

## Keywords
Stack traces; attack surface; security

## 1. INTRODUCTION
The "attack surface" of a software system is defined by the Open Web Application Security Project as "the paths in and out of a system, the data that travels those paths, and the code that protects both.[1]" Identifying the attack surface of a software system, however, is nontrivial. Manadhata et al. described the attack surface using API entry and exit points [1]. Munaiah et al. described the attack surface of a system as adjacency to nodes that appear on a call graph for the target system. [2]. Younis et al. explored the concept of attack surface entry points to describe the risk of vulnerability exploitation [3].

Providing a set of code artifacts that appear on the attack surface to software practitioners for security effort prioritization could help efficiently identify potential security issues. To that end, we have developed Risk-Based Attack Surface Approximation (RASA), which uses crash dump stack traces from a target system to identify potentially vulnerable code. Previous research into RASA [4, 5] determined that the approach shows significant promise in helping security professionals prioritize their efforts.

In our poster, we will describe our ongoing efforts in validating RASA. The goal of this poster is to *help software engineers prioritize their security efforts by deriving metrics used for effort prioritization from crash dump stack traces.*

## 2. CONTENT
Our poster will contain information on our ongoing validation efforts with RASA. We will highlight several recent results for RASA, including the ability for RASA to identify changes in the attack surface of a system over time. The change in the Windows operating system attack surface from 2014 to 2015 for versions 8 and 8.1 are seen in Table 1.

Our poster will also feature descriptions of RASA metrics, including shape metrics, which describe how many different incoming and outgoing calls there are from particular code artifacts, and depth metrics, which describe how far code artifacts are from the edge of a software system. These metrics could help identify particular places in a software system that would benefit from security inspection.

| System | Year | %Code Coverage | %Vulnerabilities |
|---|---|---|---|
| Windows 8 | 2014 | 11.0% | 17.6% |
| | 2015 | 11.9% | 17.8% |
| Windows 8.1 | 2014 | 8.1% | 16.3% |
| | 2015 | 10.0% | 18.1% |

**Table 1. Percentage of files appearing on crash dump stack traces and vulnerabilities appearing on crash dump stack traces for operating system/year pairings.**

## 3. REFERENCES
[1] P. Manadhata, J. Wing, M. Flynn, and M. McQueen, "Measuring the attack surfaces of two FTP daemons," in *2nd ACM workshop on Quality of Protection*, ed, 2006, pp. 3-10.

[2] N. Munaiah and A. Meneely, "Beyond the Attack Surface: Assessing Security Risk with Random Walks on Call Graphs," presented at the Proceedings of the 2016 ACM Workshop on Software PROtection, Vienna, Austria, 2016.

[3] A. A. Younis, Y. K. Malaiya, and I. Ray, "Using Attack Surface Entry Points and Reachability Analysis to Assess the Risk of Software Vulnerability Exploitability," in *2014 IEEE 15th International Symposium on High-Assurance Systems Engineering*, ed, 2014, pp. 1-8.

[4] C. Theisen, K. Herzig, P. Morrison, B. Murphy, and L. Williams, "Approximating Attack Surfaces with Stack Traces," in *IEEE/ACM 37th IEEE International Conference on Software Engineering*, ed. Florence, Italy, 2015.

[5] C. Theisen, "Automated Attack Surface Approximation," in *The 23rd ACM SIGSOFT International Symposium on the Foundations of Software Engineering*, ed. Bergamo, Italy, 2015, pp. 1063-1065.

[1] https://www.owasp.org/index.php/Attack_Surface_Analysis_Cheat_Sheet