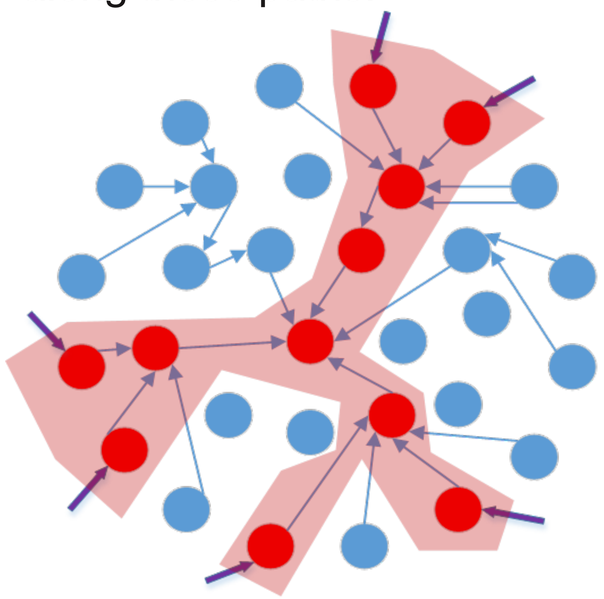
Automated Attack Surface Approximation

Christopher Theisen

Attack Surface

The *Attack Surface* of a system is the paths into and out of the system and the code and data along those paths.



Visualization of the attack surface of a system; red nodes appeared in stack traces

	Fuzzing	User Crashes
%binaries	0.9%	48.4%
%flaws	14.9%	94.6%

Attack Surface Approximation of Windows 8 and 8.1 in 2014. Shows percentage of binaries seen in stack traces, and how many security flaws were in those binaries [1].

Goal

The goal of this research is *to aid software engineers in prioritizing security efforts by approximating the attack surface of a system via stack trace analysis.*

Methods

1. Parse **any** stack traces from system
2. If code appears on a stack trace a minimum number of times, include on attack surface
3. Overlay security flaws; how many flaws included on attack surface?

Conclusions

- Code on stack traces correlates with security flaws
- Analysis can be performed on binary, file, function level
- Additional metrics based on stack traces may be useful
  - Graph shapes?
  - Boundary of a system?
- What kind of tools could help developers?
  - Visualization IDE plugin?
  - Statistical summaries?

	Files	Flaws	%Files	%Flaws	Precision	Recall
>= 1	4998	282	8.4%	72.1%	0.056	0.721
>= 30	1853	210	3.1%	53.7%	0.113	0.537
>= 140	969	162	1.6%	41.4%	0.167	0.414
All	59437	391	-	-	-	-

Attack Surface Approximation of Mozilla Firefox in 2010-2012. The more appearances on a stack trace, the more likely a file has a security flaw.

[1] C. Theisen, K. Herzig, P. Morrison, B. Murphy, and L. Williams, "Approximating Attack Surfaces with Stack Traces," in *Companion Proceedings of the 37th International Conference on Software Engineering*, 2015

NC STATE UNIVERSITY