



Limiting Recertification in Highly Configurable Systems

Analyzing Interactions and Isolation among Configuration Options

Christian Kästner, Jürgen Pfeffer

Institute for Software Research
School of Computer Science
Carnegie Mellon University

Configuration Options as Problem for Security

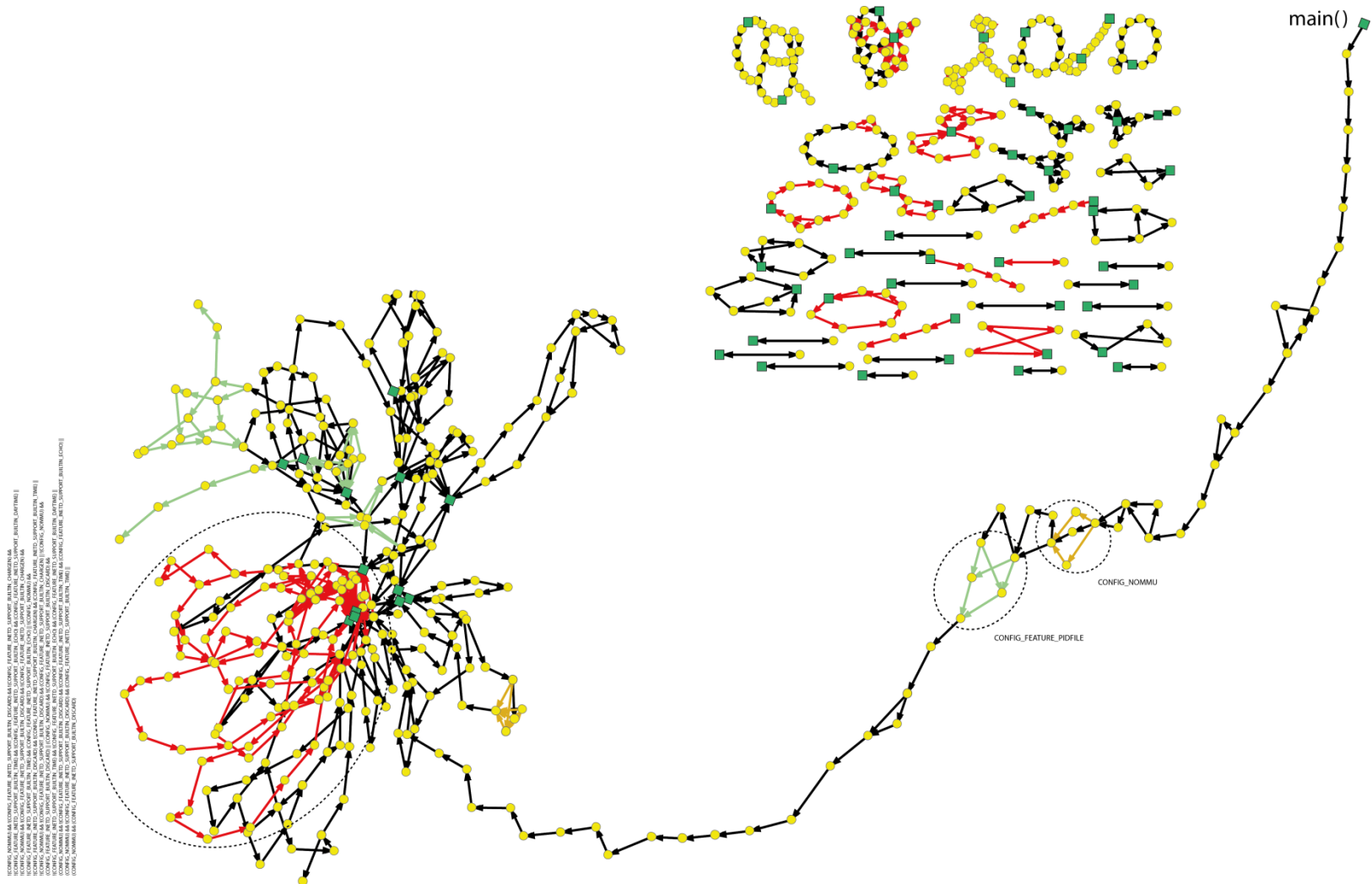
- Configuration space grows exponentially with configuration options
- Challenging to analyze and certify for security properties
- Vulnerabilities may hide in specific configuration combinations
- Current approaches:
 - Work on specific snapshots
 - Require expensive re-certifications
- *Example Linux kernel: Disabling unneeded functionality could avoid over 50% of CVE security vulnerabilities for most users!*

Our Approach

- Track configuration options with TypeChef infrastructure
- Build control-flow graphs for the system with all configurations
- Extract network information from the source code
- Determine for every code structure a presence condition
- Represent the structure for all configurations at once (in graphs)
- Apply network analysis metrics to software dependency graphs

Case Study: Linux & SELinux (project), Busybox (here)

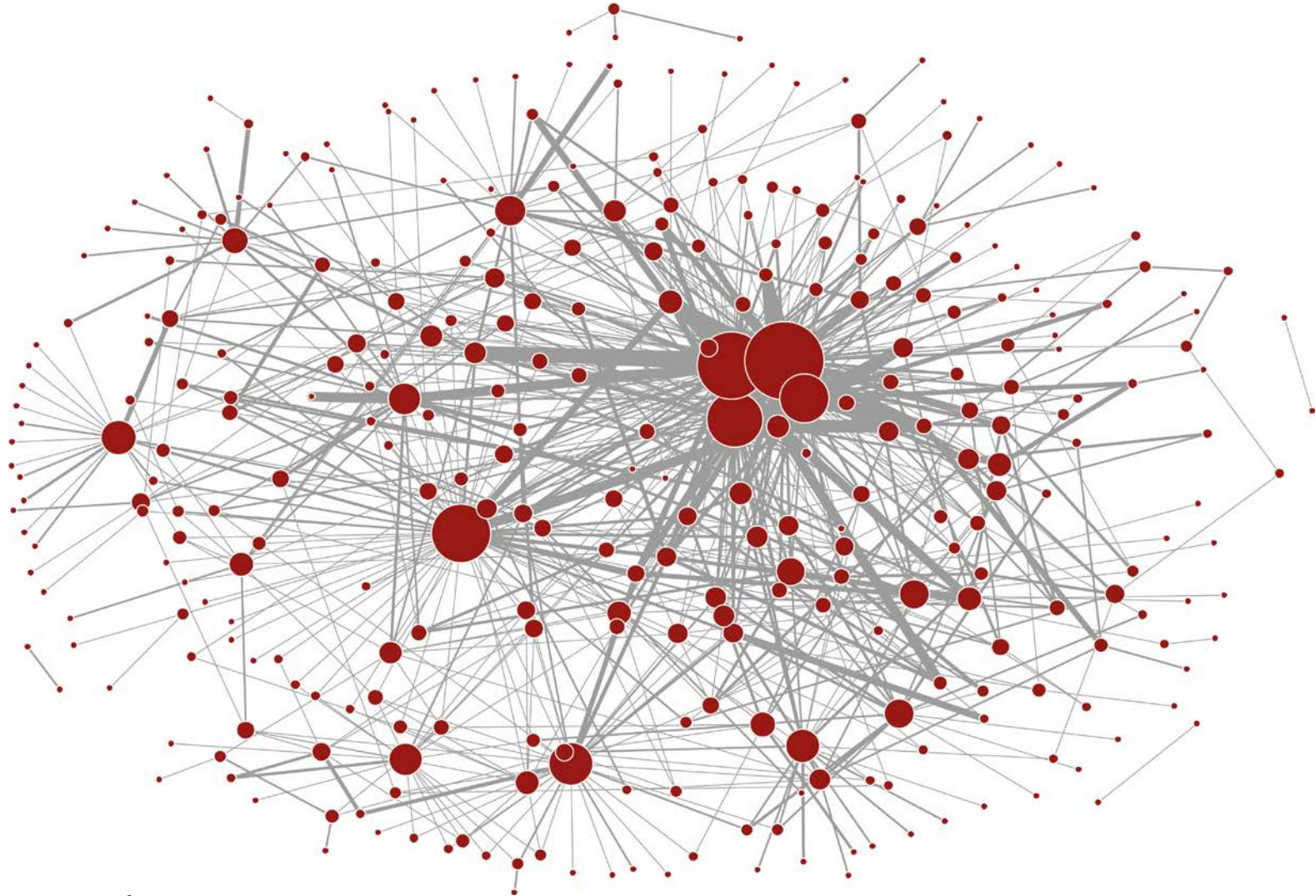
Control Flow Graph with Presence Conditions



Feature Connectivity



File Connectivity



Challenges

- Multi-level analysis (statements in functions in files in directories)
- Study graphs with complex (propositional) presence conditions
- Cross validation with known vulnerabilities and severities (NIST National Vulnerability Database)
- Over Time Analysis: Opportunities for intervention

