

Static Analysis @ CTI

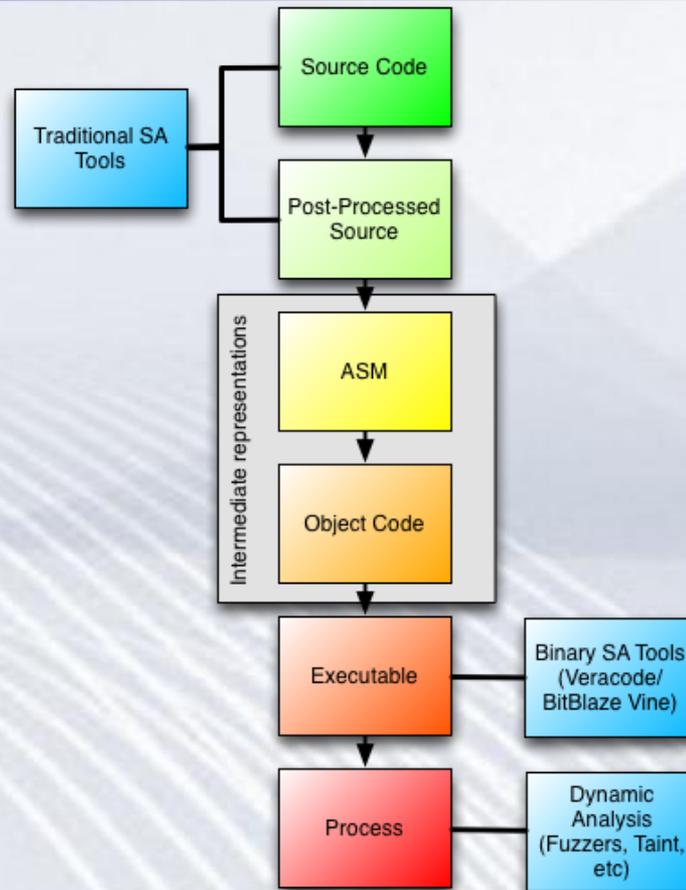
Richard Carback
<rtcarba@cti-usa.net>

- Not just eliminating vulnerabilities:
 - Compliance Issues
 - Supply Chain Problem
 - Functional Verification

- Focus: static binary analysis (WYSINWYX)

- Combining Source and Static Binary Analysis Techniques
- Utilizing Big Data Capabilities
- Dynamic Whole-System Analysis

Combining Source and Binary Static Analysis Techniques

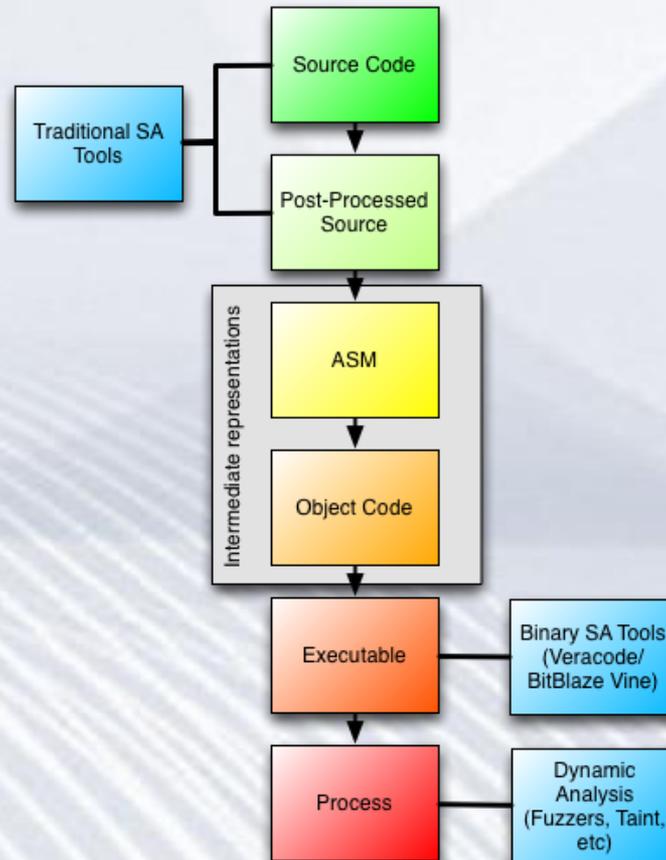


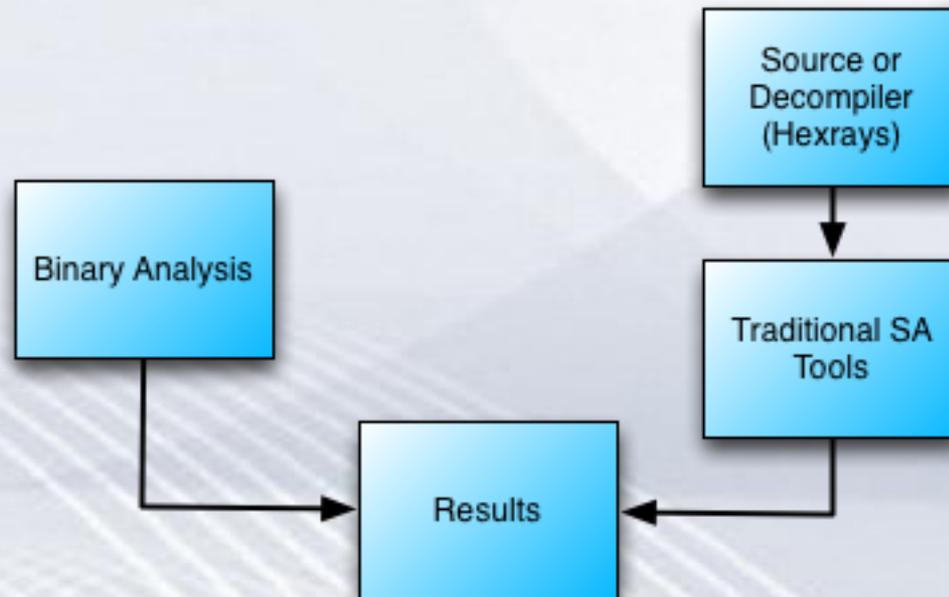
```
...  
memset (password,  
        '\0', len);  
free (password)  
...
```

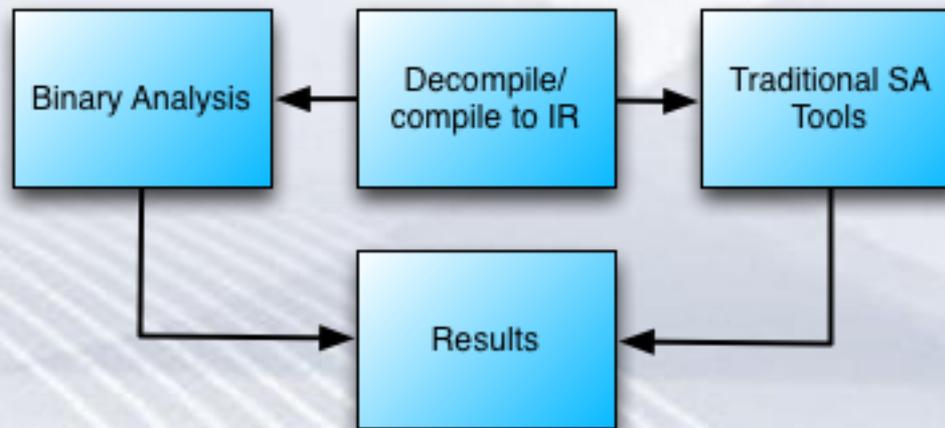
```
...  
movq -8(%rbp), %rdi  
movl $0, %eax  
call _free  
...
```

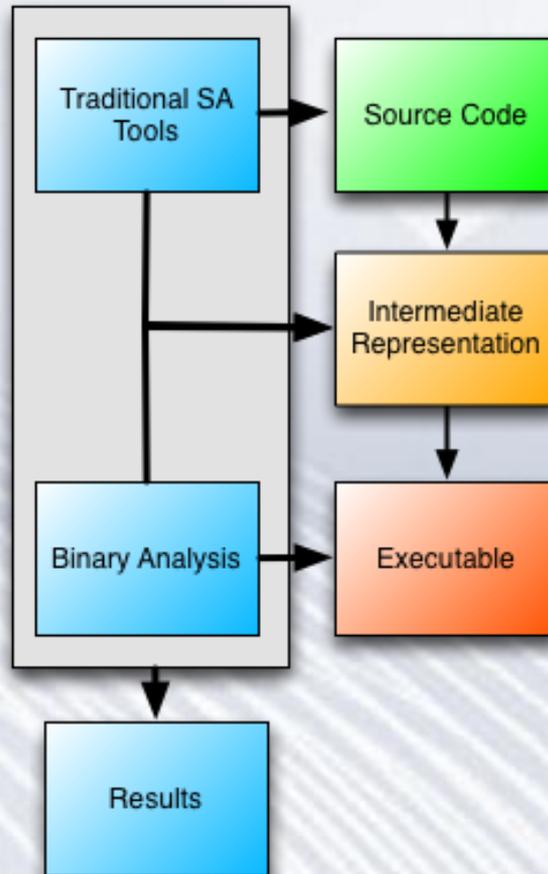
Binary Analysis is Also Limited

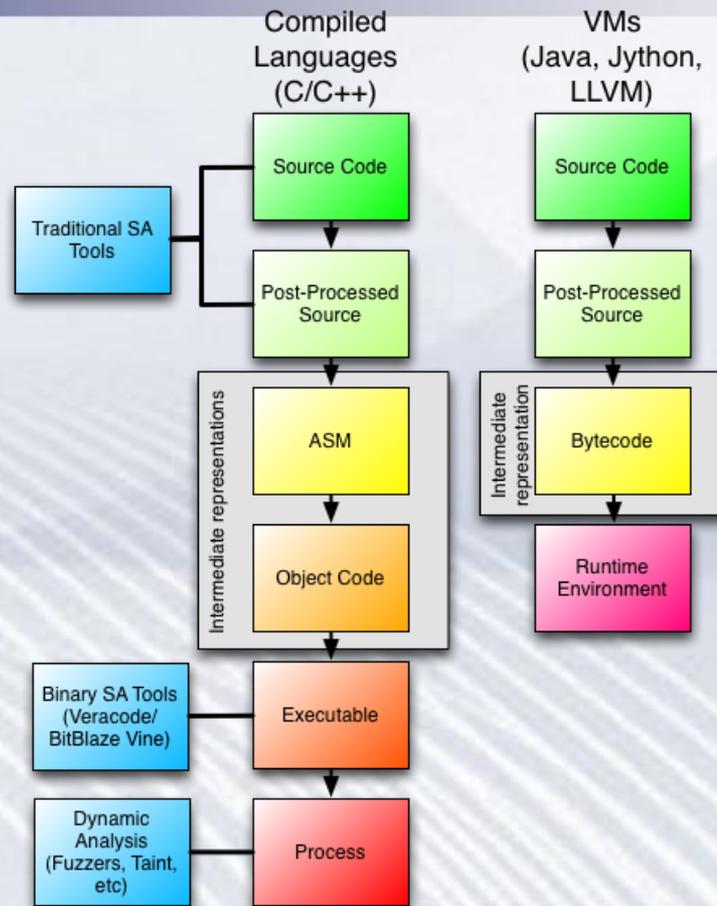
- Obviously, it is much harder
 - Indirect control flow can get expensive
- Limited utility
 - Only this compilation is analyzed, what about updates? Other Architectures?
- How do I fix problems?











Utilizing Big Data Techniques

- What compiler?
- What (static) libraries?
- Is there any copied code?
- Is there any (known) malicious code?
- Is this a new version of a program I've analyzed previously?
 - What changed?

- Break up by function
- Remove pre/post ambles
 - focus on what is unique in each function
- Convert to intermediate machine code representation
 - Things which do the same thing collapse to the same representation

- Map
 - Scan for matches
 - Calculate match score
 - Return if “good enough”
- Reduce
 - Return best answer(s)

Dynamic System Analysis

System State Monitoring Problem

- Detect when state is “compromised”
- Come up with a good way to create a virus signature on the fly to prevent further infection.

- **Measure State**
 - Watch system under normal operations with test data
 - Run the system through a fuzzer
- **Record**
 - Current function
 - Stack frames
 - Heap usage

- Watch network traffic
- When state does not match observed record...
 - Record anomaly and send alert
 - Grab traffic and look for shell/exploit code
 - Use static analysis to look for known patterns
 - Generate signature if possible
 - Record any new processes on system

- Think bigger
 - SA has much wider applicability than looking for clean code
- Think framework
 - Can we agree on a common (pseudo-compiled) representation for all architectures?