



UNIVERSITY POLITEHNICA OF BUCHAREST
FACULTY OF AUTOMATIC CONTROL AND COMPUTERS
COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
Advanced Cybersecurity Master Program

OpenCRS: Attack Surface Approximation, Vulnerabilities Discovery, and Automatic Exploitation of Binaries

Thesis Advisors

Adrian-Răzvan Deaconescu
Constantin-Eduard Stăniloiu

Student

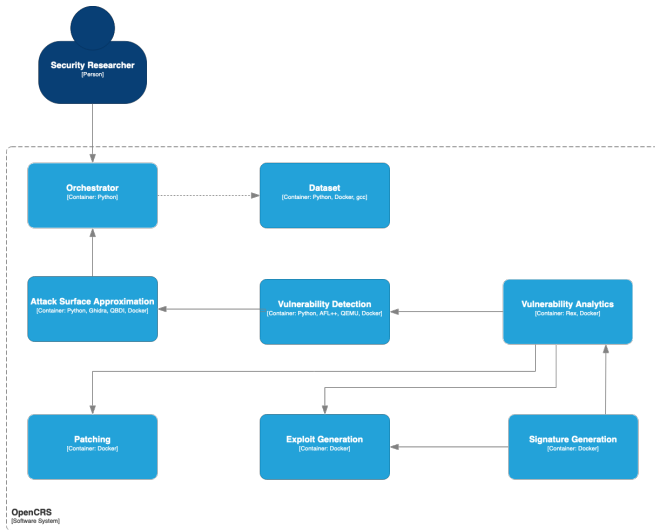
George-Andrei Iosif

Bucharest
2023

- ▶ Commercial software as an use cases for binary-only testing
- ▶ DARPA's Cyber Grand Challenge
- ▶ Retired open source versus active commercial CRSes

- ▶ Open source¹ cyber reasoning system
- ▶ Updated capabilities
 - ▶ Finding and patching vulnerabilities
 - ▶ Creating and detecting exploits
- ▶ Focus
 - ▶ Executable and Linkable Format (ELF)
 - ▶ C codebase
 - ▶ i386
 - ▶ Arguments, stdin, or files

¹<https://github.com/CyberReasoningSystem>



The Attack Surface Approximation Module



- ▶ Discovery of input streams' indicators
 - ▶ Decompilation and AST parsing
 - ▶ Imported functions' bucketing
- ▶ Generation of dictionaries with heuristics
- ▶ Arguments discovery and role attachment
 - ▶ Execution tracing
 - ▶ Monitoring of file openings

The Vulnerability Detection Module



- ▶ Detector recommendation
- ▶ Fuzzing with afl++
 - ▶ Already-implemented for files and `stdin`
 - ▶ With custom harness for arguments
- ▶ Emulation

The Automatic Exploit Generation Module



- ▶ Exploiter recommendation
- ▶ BoF exploitation with forked **Zeratool**
- ▶ Mitigations bypass



- ▶ Three integrated test suites
- ▶ Automatic building by test suite parsing
- ▶ Dataset querying, with filter

Heuristic	Arguments Count
Generation	62
Manuals' parsing	6701
Binary pattern matching on <code>uname</code>	37

Table 1: Arguments Dictionaries, by Heuristic

Executable	Arguments Stream	Files Stream	stdin Stream
<code>null_pointer_deref_args.elf</code>	Detected (TP)	N/A	N/A
<code>null_pointer_deref_files.elf</code>	Detected (TP)	Detected (TP)	Detected (FP)
<code>null_pointer_deref_stdin.elf</code>	Detected (TP)	Detected (FP)	N/A
<code>multiple_inputs_streams.elf</code>	Detected (TP)	Detected (TP)	Detected (TP)

Table 2: Accuracy in Detecting the Input Streams

```
$ opencrs-surface fuzz --elf /bin/uname --dictionary uname.dict
Several arguments were detected for the given program:
```

Argument	Role

-	FLAG
-a	FLAG
-a string	STRING_ENABLER
-i	FLAG
-i string	STRING_ENABLER
-m	FLAG
-m string	STRING_ENABLER
-n	FLAG
-n string	STRING_ENABLER
-o	FLAG
-o string	STRING_ENABLER
-p	FLAG
-p string	STRING_ENABLER
-r	FLAG
-r string	STRING_ENABLER
-s	FLAG
-s string	STRING_ENABLER
-v	FLAG
-v string	STRING_ENABLER

Figure 3: Arguments Fuzzing for /bin/uname

Weakness	Count
Stack-based Buffer Overflow	13836
Heap-based Buffer Overflow	11088
Integer Overflow or Wraparound	3960
Mismatched Memory Management Routines	3564
Integer Underflow	2952
Free of Memory not on the Heap	2680
Use of Externally-Controlled Format String	2410
Buffer Underflow	2048
Buffer Under-read	2048
OS Command Injection	1921

Table 3: Executables in `opencrs_dataset`², by CWE

²https://github.com/CyberReasoningSystem/opencrs_dataset

- ▶ Successful vulnerability discovery
 - ▶ Different input streams
 - ▶ Different vulnerabilities
 - ▶ Tainted format string
 - ▶ NULL pointer dereferencing
 - ▶ Stack buffer overflow
- ▶ Three executables for which exploits were generated

Future Work

- ▶ New input streams
- ▶ Pair-wise testing of arguments
- ▶ Labels for input streams
- ▶ New vulnerability detection techniques
- ▶ Exploitation of tainted format string

Conclusion

- ▶ Open source cyber reasoning system
- ▶ Four functional modules
- ▶ Additional public dataset
- ▶ Promising results

