

Tekeli-li

The Great Deep Project

Team Members

Michael Weatherby

Michael Fitch

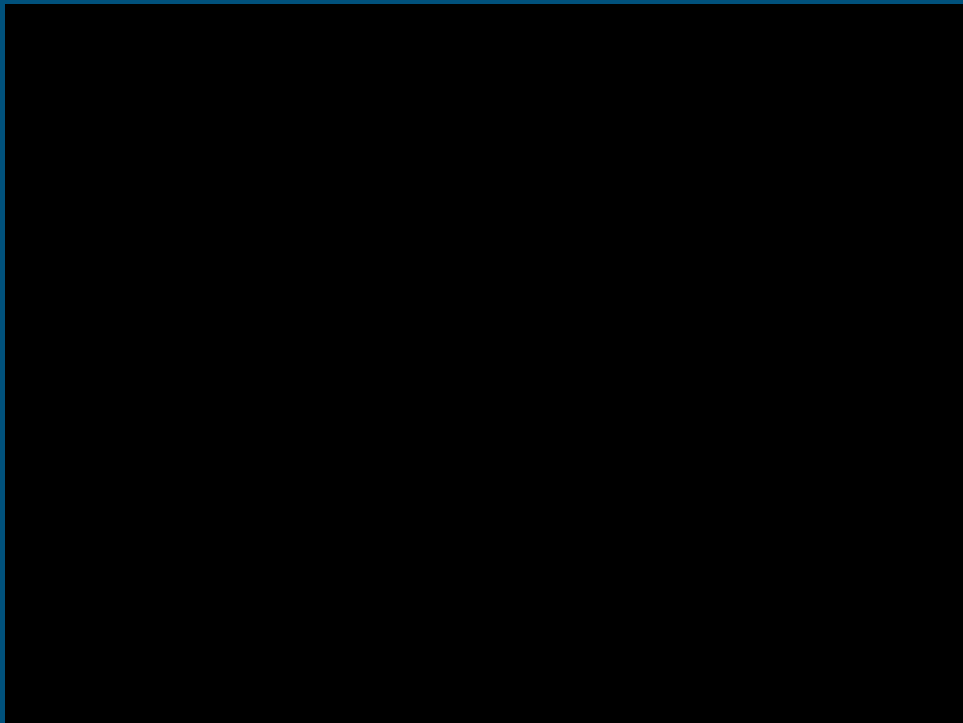
Blake Janes

Eric Del Valle



Shoggoth

- Built on QEMU
- Designed for high performance Dynamic Binary Instrumentation
- Key additions to QEMU include:
 - User-made plugins
 - OS Handlers
 - Rapid Analysis tool
- Intended as a complete starting program for binary exploitation research on operating systems



Main Project Goals

- Test Shoggoth as a “typical user” to provide feedback on:
 - Difficulty to successfully setup and run
 - Clarity of documentation
 - Accessibility of key features
- Write Documentation for:
 - Clarify existing documentation where appropriate
 - Completed goals that lack documentation
 - Problems preventing normal usage, including bugs
 - Anything else a potential future user may need to know

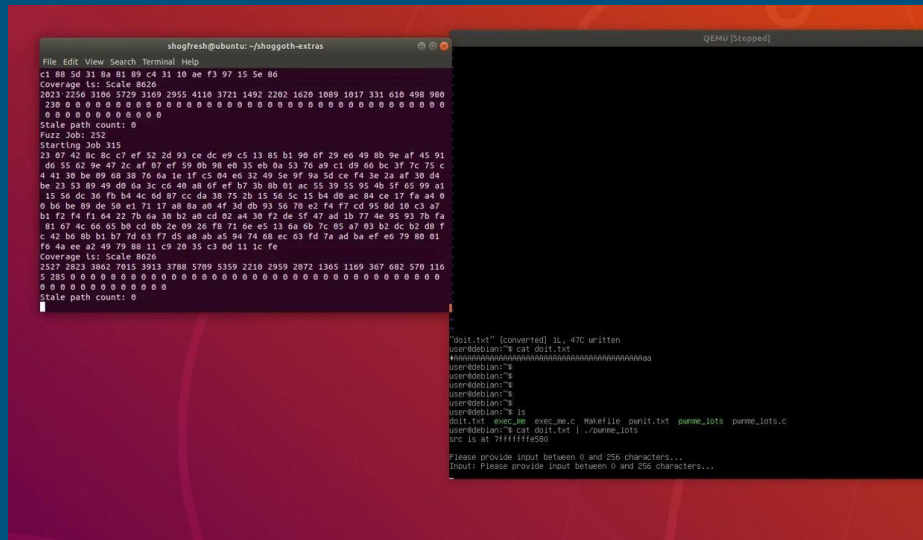
100

- [illegible]

Shoggoth running a simple fuzzer

Completed - Advanced Fuzzing

- Coverage Guided Fuzzer
 - Uses program coverage analysis to try and reach each line of code
 - Rare edge cases may cause a crash if the fuzzer manages to reach them
- File Fuzzer
 - Reads input parameters from a file
 - Intended to work alongside AFL
- Documentation created for both fuzzers



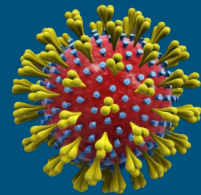
Completed - Create OS Handler

- Setup Untangle in Shoggoth
 - Untangle is a linux based firewall
 - Searched for potential attack vectors in Untangle
- Found bug in Shoggoth's "find_mm_offset()" function
 - Newer linux kernels (post v4.1) have a different offset for this struct
 - Discovered workaround- offset can be hardcoded if location is known
 - Attempted a few fixes on the function, but none succeeded



Untangle running in Shoggoth

Difficulties



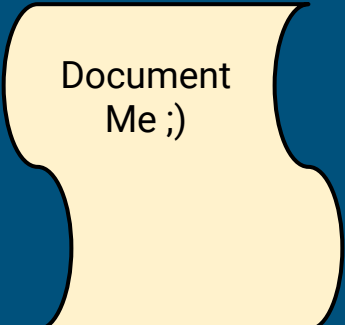
- The big one - Covid-19
 - Caused a temporary breakdown in communication with Cromulence
 - Delayed the project moving forward for awhile
 - Forced more remote collaboration, which was not initially planned
- Too ambitious
 - Initially, the team wanted to rush through the middle parts of the project (setting up basic systems and writing documentation) to try and find a vulnerability
 - A bit too late, it was realized that the middle parts of the project were the most important

Lessons Learned

- Understand your capabilities
 - At the beginning, the group planned to do too much in the timeframe given, so the group should have better understood their capabilities
- Set up communications as fast as possible
 - The best way to learn what the sponsor wants is to ask them, and the group took too long to set up reliable communications with them- especially after Covid-19
 - At minimum, have weekly meetings

Future Potential

- Create more advanced plugins- potentially intended for redistribution
- Use Shoggoth to search for real vulnerabilities in an OS
- Add additional OS handlers
- And of course... documentation! And more documentation!



Document
Me ;)