Fuzzing in the Corporate World

Gadi Evron – Beyond Security

23C3, Berlin – December 2006

Who am I…

- My name is…
- I work for…
- My interests are…
- How I got into fuzzing…

What is fuzzing?

"**Fuzz testing** or **fuzzing** is a software testing technique. The basic idea is to attach the inputs of a program to a source of random data ("fuzz"). If the program fails (for example, by crashing, or by failing built-in code assertions), then there are defects to correct."

- Wikipedia.

The evolution of fuzzing tools

- Random input
- Sensible input
- Block-based input
- Protocol-based input

Emerging:

- Session-based input (and later on logical/implementation fuzzing)
- Monitoring (passive) and “engine driving” (active)

Smart fuzzing vs. “silly” fuzzing

- What is smart fuzzing and why all the buzz?
- Is smart fuzzing up to the task? Exhaustive vs. targeted
- Software testing vs. security testing

Advances in the fuzzing world (not technology)

- New more reliable tools
- New frameworks
- Commercial fuzzing tools

In the past…

- Fuzzing enthusiasts
- Corporations with fuzzing enthusiasts

Cisco, Microsoft, etc.

Home-grown tools for the use of the coders who built them.

And today…

- Cisco, Microsoft, Juniper, AT&T, Symantec, etc.

All use fuzzing extensively.

This usage is no longer limited to one elite group or another, but rather integrated into their development process.

Integration into the development cycle

- With other security QA tools
- As stand-alone tools
- Where do you integrate the tools?
- Challenges with integration in a QA environment

Fuzz before release

There are little to no excuses left for vendors to release software without prior security-oriented testing.

Such testing can not eliminate all bugs of all types, but it can eliminate most of the “lower hanging fruit” and some of the more difficult to discover and exploit vulnerabilities.

It’s about raising the bar.

Fuzzing as a process for vendors

What one vendor looks for (with examples from Microsoft and the “Security Development Lifecycle“):

- Fuzz each protocol on its own
- Maximize your testing – find all you can. Reach coverage as close to 100% as possible
- Fuzz whatever malicious input a user can provide (file formats, program parameters, network traffic, ActiveX controls, extension protocols, drivers, APIs, etc.)
- Call fuzzing: “Automated run-time vulnerability discovery”. Don’t use only random input, feed the application what also what it might expect

Fuzzing as a process for vendors

Support:

- State (session-based) – doing A to do B

  to do C
- Prior knowledge (value locking) – know the protocols caveats, query strings and hidden fields in HTTP, custom extensions for SIP

Define:

- What is a bug?

  Different applications exhibit failures in different ways, how do you catch these if at all? Just the simple ones?

Fuzzing as a process for vendors

And…

- Be able to train the fuzzer and change the fuzzing accordingly
- Tie in with proxies (MiTM), code coverage and human auditing
- Complete fuzzing?
- “Fuzzing finds many bugs and security vulnerabilities, but it can not find all of them” == software testing can…

Fuzzing as a process for vendors

Implement…

- Both block-based and session-based
- Use both test-automation (traffic analysis – and miss a lot of bugs) as well as knowledge-based fuzzing (and the pain of documenting everything).

  Finding formal protocol descriptions??

Fuzzing as a process for vendors

QA-level fuzzing tools (with *some* recent Cisco adoptions as examples):

- Exceptions, no code!
- Simple to use
- Will finish its run in (a very) reasonable time-frame
- Distribute to QA teams

Or

- Have an infrastructure dev and QA teams can use

Fuzz before purchase

One of the surprises of selling fuzzing products at Beyond Security, is who actually wants them.

Banks, Telcos, large corporations.

Fuzz before purchase

- Being able to better decide on the security and stability of products than look at their vulnerability history
- Buying what's up to scratch
- Adding security to the list of demands from a product, and not just by a feature list

Certification

Discussion point:

- Certifying products with automated tools

- QA Security Engineer certification.. 3 years down the road?

“Test with this to work with us”

Discussion point:

- Certify your product using this or that fuzzer…

  Not exactly a certification, but when done by a large telco such as, for example, AT&T…

“Test with this to work with us”

Certification by affiliation:

- Consider ISO19977/BS9977, etc.
- When a vendor is interested in working with another vendor, it may be required to meet certain regulation or standardization (Microsoft and the “Security Development Lifecycle“ with development for Vista an example)

Open discussion

- Questions?

gadie@BeyondSecurity.com

  - Thank you!
  - Join the fuzzing mailing list