



Fuzzing

Piotr Łaskawiec
J2EE Developer/Pentester

Metrosoft (www.metrosoft.com)
piotr.laskawiec@gmail.com

OWASP

14.01.2010

Copyright © The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document
under the terms of the OWASP License.

The OWASP Foundation
<http://www.owasp.org>

Abstract

- Methods of testing the application
- Ensuring application security
- Fuzzing – definition
- Use of fuzzing
- Types of fuzzers
- Fuzzing and SDLC
- Who uses fuzzers?
- Examples of fuzzers
- Web application fuzzing
- Summary

Application testing

■ Popular tests:

- ▶ Unit testing
- ▶ Functional testing
- ▶ Regression testing
- ▶ Performance testing
- ▶ Usability tests

■ Other classification:

- ▶ Whitebox, Graybox, Blackbox

■ What about security?

- ▶ Security on level of design, implementation, testing and deployment.

Ensuring application security

■ Response vs prevention

■ Response:

- ▶ Firewall
- ▶ IDS/IPS
- ▶ Antivirus
- ▶ Authentication mechanisms
- ▶ Vulnerability scanners (Nessus, Nikto, etc.)
- ▶ Etc.

■ Prevention:

- ▶ **Fuzzing!**
- ▶ Code audit/RE

Fuzzing - definition

- Fuzzing is a method of testing software to find security holes and unexpected behavior of an application, using semirandom data.
- Fuzzing most frequently is fully automated process - „run and wait for result“.

Fuzzing – what does it mean in practice?

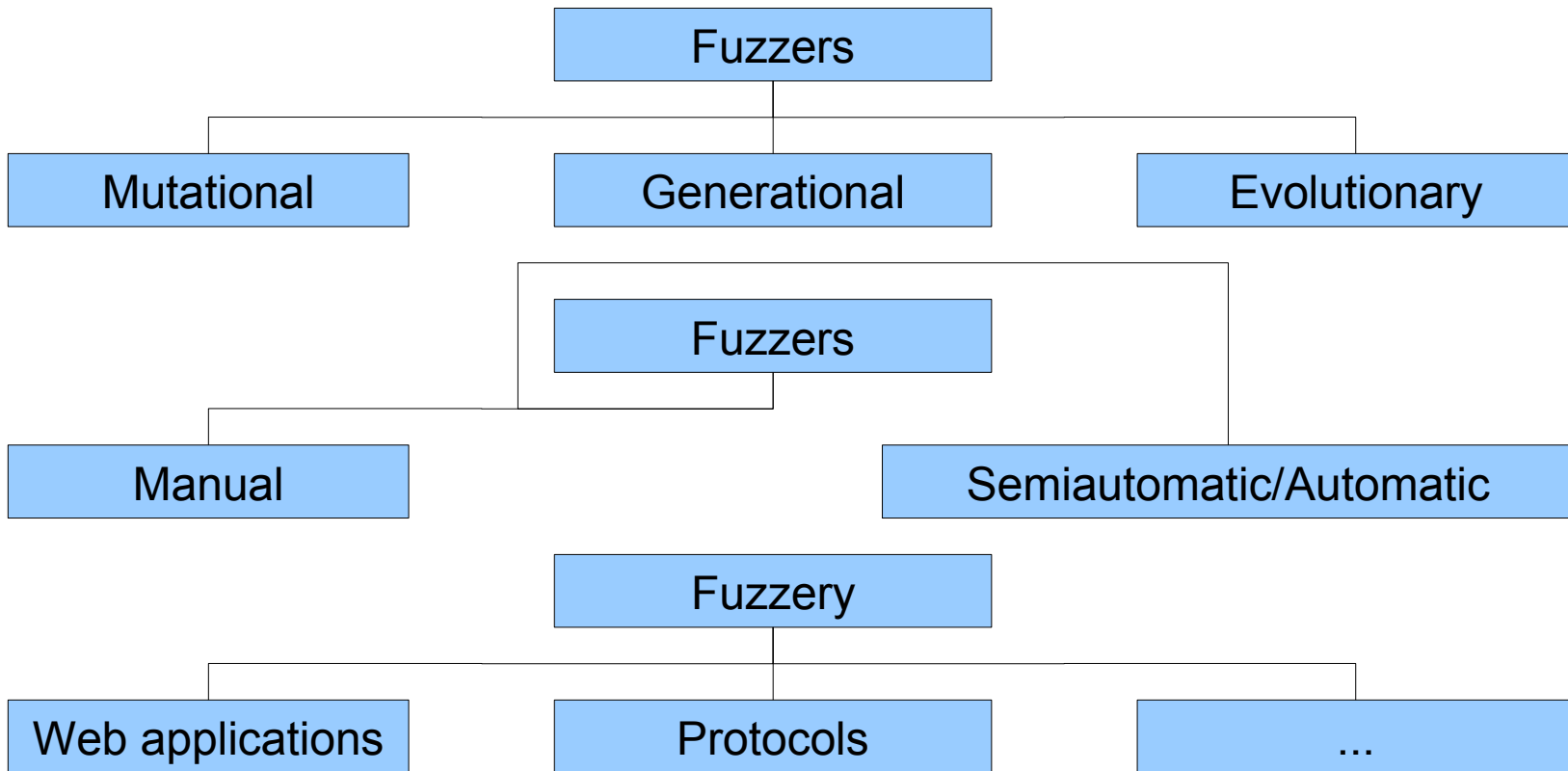
- Fuzzing == Negative testing
- The aim of the fuzzer is to send invalid data to application (too long strings, improper encoding, bad file format, bad sequence of communicates).
- We hope that the application will accept the data and an atypical reaction of the program will occur – DoS, error message, fast-growing demand for resources.
- Our target is to break the application!

Using fuzzers

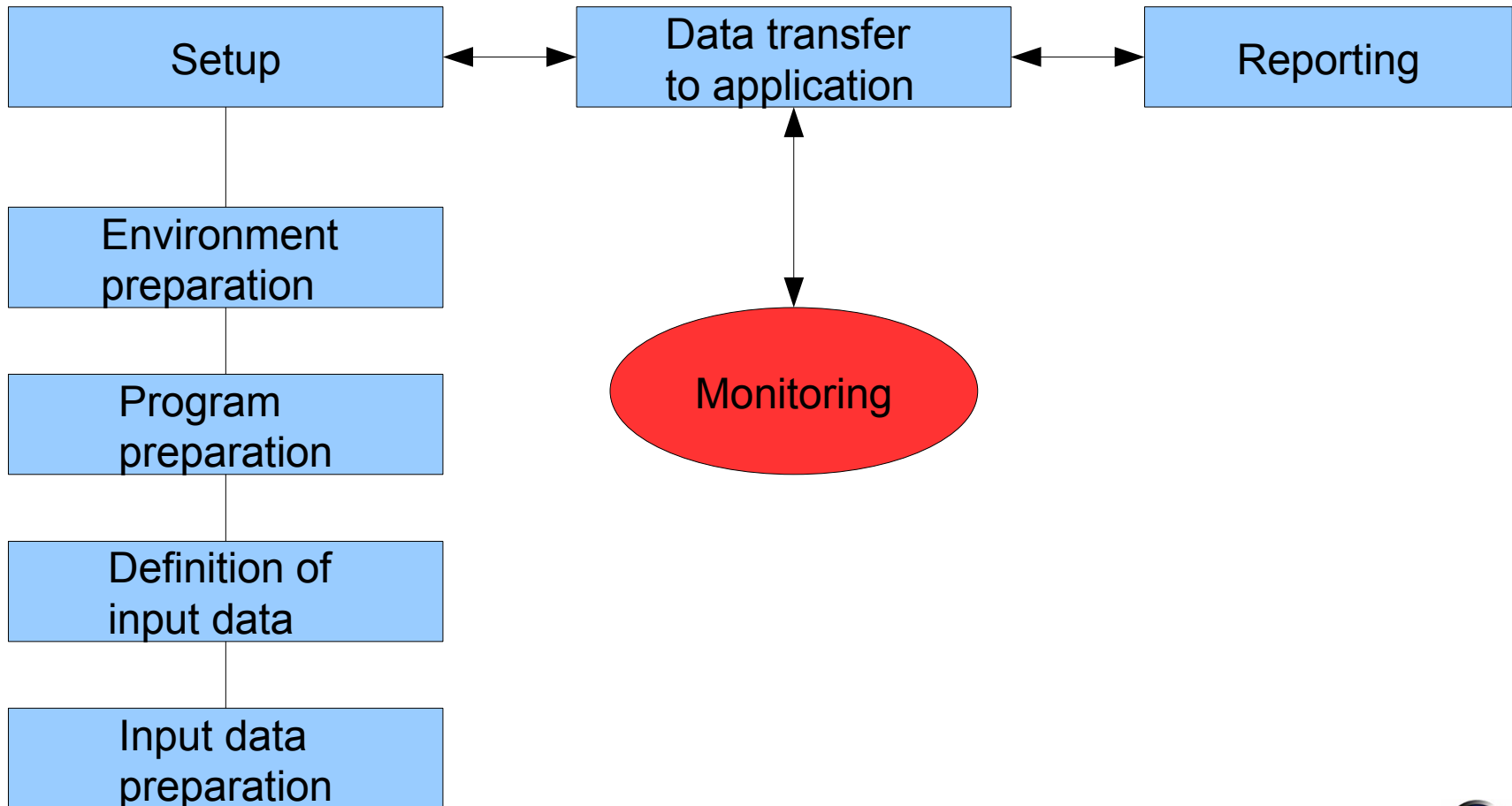
- Local applications
- **Web applications**
- WebServices
- Network applications
- ActiveX controls
- Files
- Libraries
- ...

Fuzzer's classification

- There are many criteria for classification
- Examples:



Fuzzing process



Monitoring

- Observation of program behavior
- Logs
- Debuggers (!exploitable...)
- Files, processes and network monitors
- Virtualization (VMWare)
- Source code modifications (breakpoints)
- Additional techniques (Valgrind, Guard Malloc)
- Combined techniques

Process Explorer

Process Explorer - Sysinternals: www.sysinternals.com [apejron\hellsource]

File Options View Process Find Users Help

Process	PID	CPU	Description	Company Name
System Idle Process	0	66.72		
Interrupts	n/a	0.77	Hardware Interrupts	
DPCs	n/a		Deferred Procedure Calls	
System	4	1.53		
smss.exe	312			
csrss.exe	436			
wininit.exe	504			
services.exe	552			
svchost.exe	740			
MATLAB.exe	1880			
ACEngSvr.exe	2856			
WmiPrivSE.exe	2092			
igfxsvc.exe	2920		igfxsvc Module	Intel Corporation
BTStackServer.exe	4044		Bluetooth Stack COM Server	Broadcom Corporation.
BluetoothHeadset...	3304		Bluetooth Headset Skype Pr...	Broadcom Corporation.
svchost.exe	804			
svchost.exe	880			
audiodg.exe	6128	3.83		
svchost.exe	952			
dwm.exe	2628		Menedzer okien pulpitu	Microsoft Corporation
svchost.exe	980			
taskeng.exe	2600			
BatteryLife.exe				

CPU Usage: 32.98% Commit Charge: 46.59% Processes: 114 Physical Usage: 67.65%

Process Monitor

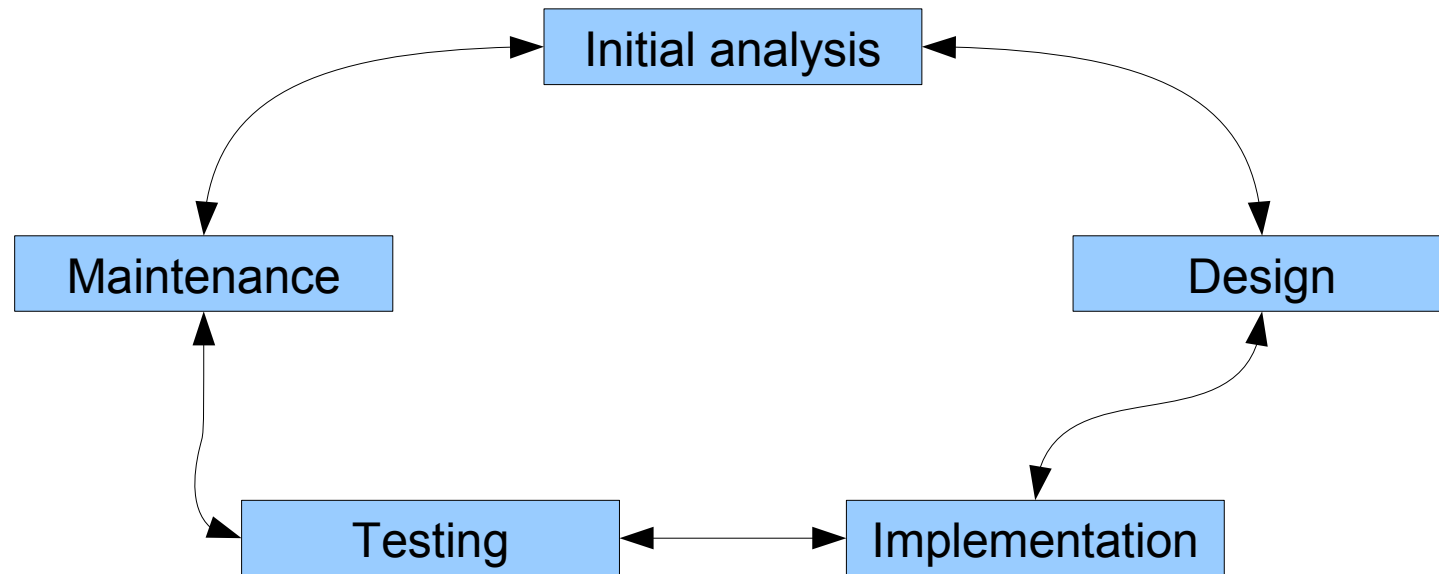
Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

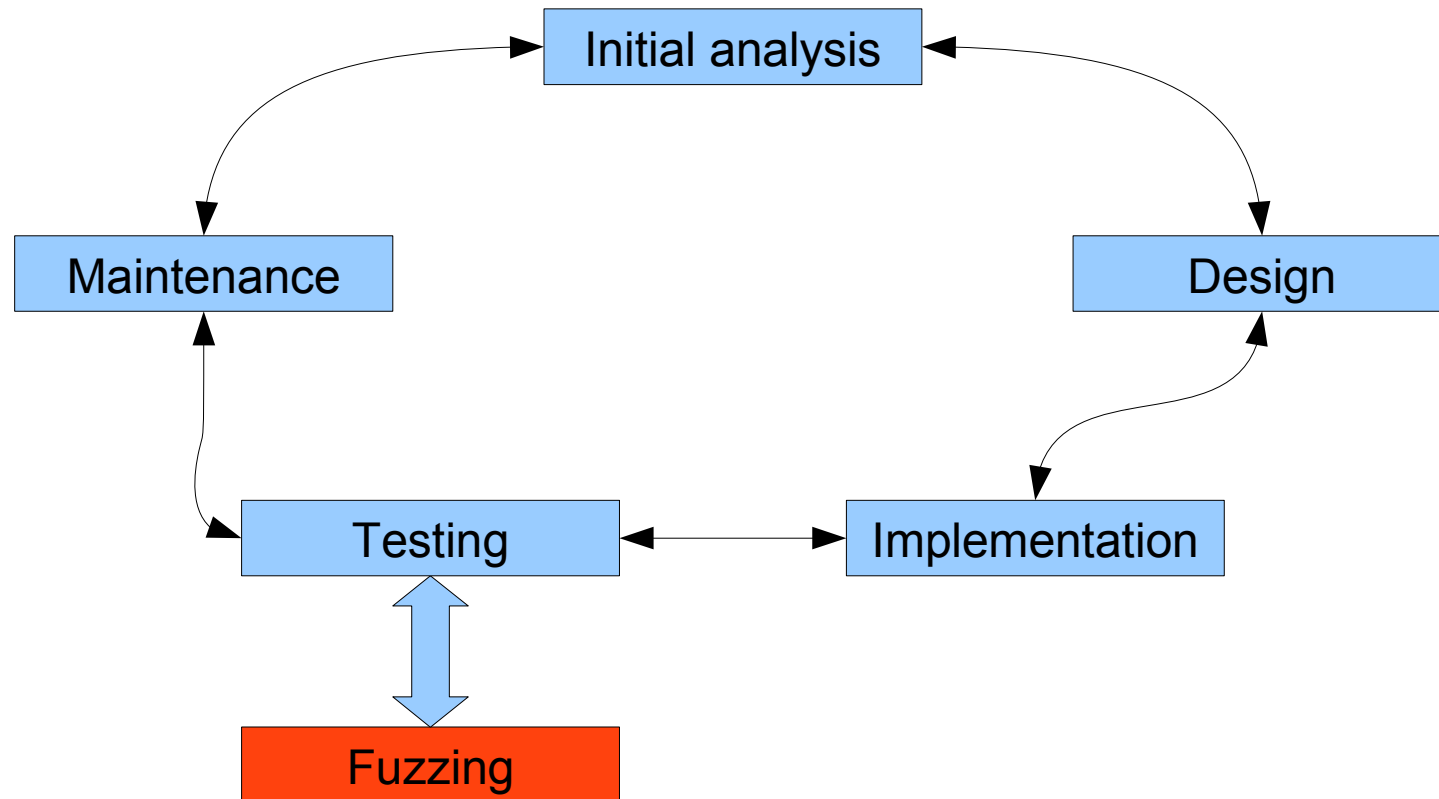
Time ...	Process Name	PID	Operation	Path	Result	Detail
22:06:...	Explorer.EXE	2656	RegQueryKey	HKCU\Software\Classes	SUCCESS	Query: HandleTags, HandleTags: 0x0
22:06:...	Explorer.EXE	2656	RegOpenKey	HKCU\Software\Classes\Applications\...	NAME NOT FOUND	Desired Access: Read
22:06:...	Explorer.EXE	2656	RegOpenKey	HKCR\Applications\Procmon64.exe	NAME NOT FOUND	Desired Access: Read
22:06:...	Explorer.EXE	2656	QueryOpen	D:\tools\analis\ProcessMonitor\Procmo...	SUCCESS	CreationTime: 2010-01-07 22:01:28, LastAccessTime: 2010-01-07 22:0...
22:06:...	Explorer.EXE	2656	ReadFile	C:\Windows\System32\timedate.cpl	SUCCESS	Offset: 191 488, Length: 4 608, I/O Flags: Non-cached, Paging I/O, S...
22:06:...	Explorer.EXE	2656	ReadFile	C:\Windows\System32\timedate.cpl	SUCCESS	Offset: 165 376, Length: 4 096, I/O Flags: Non-cached, Paging I/O, S...
22:06:...	Explorer.EXE	2656	RegQueryKey	HKCU	SUCCESS	Query: HandleTags, HandleTags: 0x0
22:06:...	Explorer.EXE	2656	RegOpenKey	HKCU\Control Panel\TimeDate\Additio...	NAME NOT FOUND	Desired Access: Read
22:06:...	Explorer.EXE	2656	RegQueryKey	HKCU	SUCCESS	Query: HandleTags, HandleTags: 0x0
22:06:...	Explorer.EXE	2656	RegOpenKey	HKCU\Control Panel\TimeDate\Additio...	NAME NOT FOUND	Desired Access: Read
22:06:...	Explorer.EXE	2656	Thread Exit		SUCCESS	Thread ID: 4684, User Time: 0.0000000, Kernel Time: 0.0000000
22:06:...	Explorer.EXE	2656	RegQueryValue	HKCU\Software\Microsoft\Windows\C...	SUCCESS	Type: REG_BINARY, Length: 72, Data: 03 00 00 00 00 00 00 00 06 0...
22:06:...	Explorer.EXE	2656	RegSetValue	HKCU\Software\Microsoft\Windows\C...	SUCCESS	Type: REG_BINARY, Length: 72, Data: 03 00 00 00 00 00 00 00 06 0...
22:06:...	Explorer.EXE	2656	RegSetValue	HKCU\Software\Microsoft\Windows\C...	SUCCESS	Type: REG_BINARY, Length: 1 612, Data: 03Type: REG_BINARY
22:06:...	Explorer.EXE	2656	RegQueryValue	HKCU\Software\Microsoft\Windows\C...	SUCCESS	Type: REG_BINARY, Length: 72, Data: 03 00Length: 72
22:06:...	Explorer.EXE	2656	RegSetValue	HKCU\Software\Microsoft\Windows\C...	SUCCESS	Type: REG_BINARY, Length: 72, Data: 03 00Data: 03 00 00 00 00 00 00 00 00
22:06:...	Explorer.EXE	2656	RegSetValue	HKCU\Software\Microsoft\Windows\C...	SUCCESS	Type: REG_BINARY, Length: 1 612, Data: 03 00 00 00 26 00 00 00 8...
22:06:...	Explorer.EXE	2656	ReadFile	C:\Windows\System32\timedate.cpl	SUCCESS	Offset: 108 032, Length: 4 096, I/O Flags: Non-cached, Paging I/O, S...
22:06:...	Explorer.EXE	2656	ReadFile	C:\Windows\System32\timedate.cpl	SUCCESS	Offset: 83 456, Length: 28 672, I/O Flags: Non-cached, Paging I/O, S...
22:06:...	Explorer.EXE	2656	Thread Exit		SUCCESS	Thread ID: 5084, User Time: 0.0000000, Kernel Time: 0.0156001
22:06:...	Explorer.EXE	2656	Thread Exit		SUCCESS	Thread ID: 8120, User Time: 0.0312002, Kernel Time: 0.0624004
22:06:...	Explorer.EXE	2656	Thread Exit		SUCCESS	Thread ID: 5516, User Time: 0.0000000, Kernel Time: 0.0000000

Showing 102 626 of 512 473 events (20%) Backed by page file

Fuzzing and SDLC



Fuzzing and SDLC



Fuzzing and SDLC

- After publication of the new version, application is tested by a previously prepared fuzzer.
- Test results are verified by testers. Next, they are sent to programmers.
- If any errors occur, programmers must fix the application.
- New build once again must pass the fuzzing process.

Examples:

■ Adobe

- ▶ http://blogs.adobe.com/asset/2009/12/fuzzing_reader_-_lessons_learned.html

■ Bugs in IIS

- ▶ <http://pentestit.com/2009/12/28/microsoft-iis-day-open/>

■ Mozilla JavaScript fuzzer

- ▶ <http://blog.mozilla.com/security/2007/08/02/javascript-fuzzer-available/>

■ Microsoft MiniFuzz

- ▶ <http://www.microsoft.com/downloads/details.aspx?FamilyID=b2307ca4-638f-4641-9946-dc0a5abe8513&displaylang=en>

■ Google Flayer

- ▶ <http://code.google.com/p/flayer/>

Examples of fuzzers:

■ Frameworks:

- ▶ Peach (<http://peachfuzzer.com/>)
- ▶ Sulley

■ Specialized fuzzers:

- ▶ JBroFuzz (OWASP)
- ▶ WSFuzzer (OWASP)
- ▶ TAOF
- ▶ Wfuzz
- ▶ Spike Proxy
- ▶ WebFuzz

■ Custom solutions

WebScarab Fuzzer plugin

The screenshot displays the WebScarab application window with the Fuzzer plugin selected. The interface includes a menu bar (File, View, Tools, Help) and a toolbar with various options: Summary, Messages, Proxy, Manual Request, Spider, Extensions, XSS/CRLF, SessionID Analysis, Scripted, Fragments, Fuzzer, Compare, and Search. The Fuzzer configuration area shows the Method set to GET and the URL set to http://localhost:8080/test. The Version is HTTP/1.0. Below this, there are sections for Headers and Parameters, each with an Add and Delete button. The Parameters section is currently empty. At the bottom, there are statistics for Total Requests (0) and Current Request (0), along with Start and Stop buttons. A table with columns ID, Date, Method, Host, Path, Parameters, Status, Origin, and Tag is also visible. The status bar at the bottom indicates 'Started' and 'Used 7.24 of 63.56MB'.

Method: GET URL: http://localhost:8080/test Version: HTTP/1.0

Header	Value
--------	-------

Parameters

Location	Name	Type	Value	Priority	Fuzz Source
----------	------	------	-------	----------	-------------

Total Requests: 0
Current Request: 0

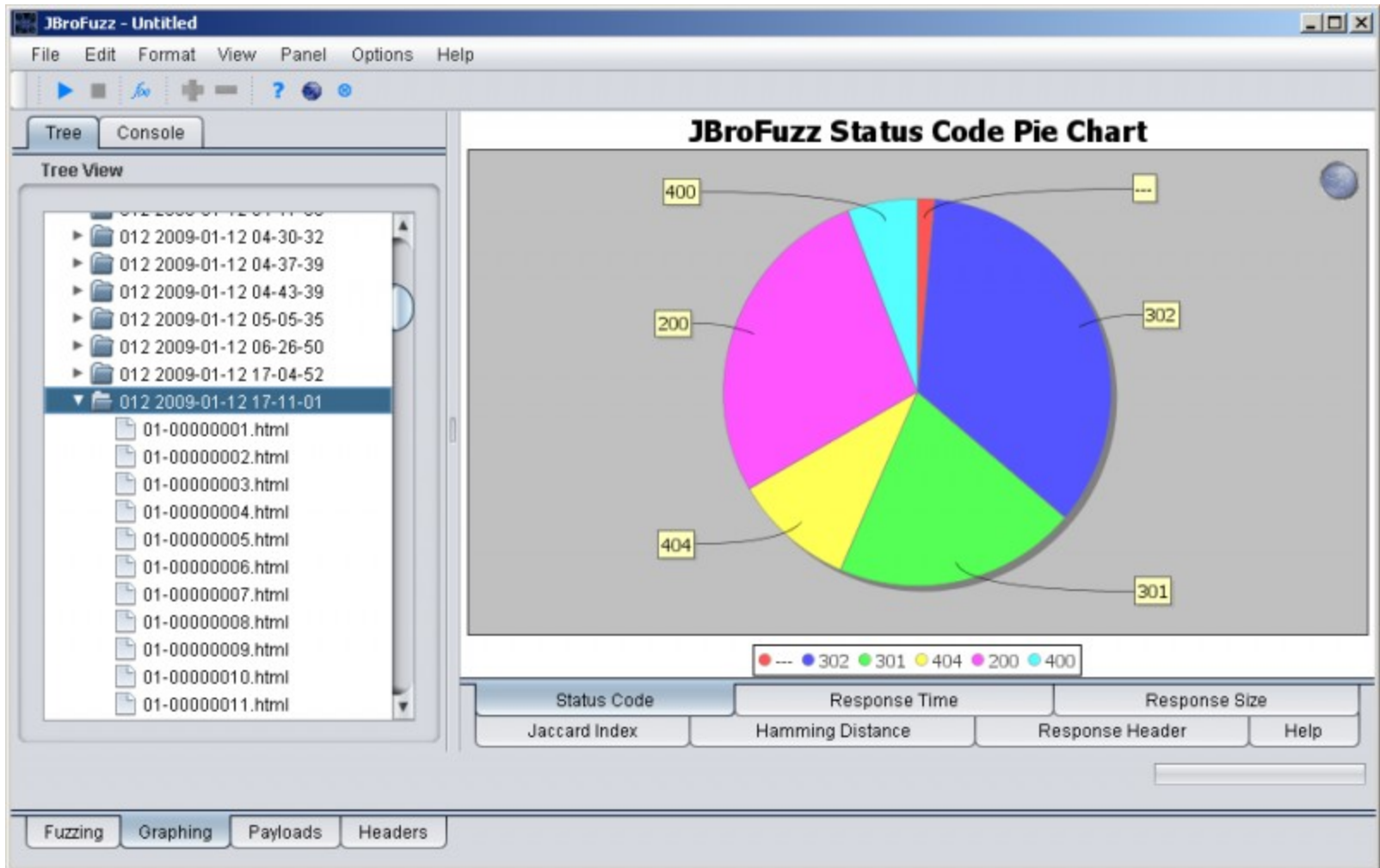
Sources Start Stop

ID	Date	Method	Host	Path	Parameters	Status	Origin	Tag
----	------	--------	------	------	------------	--------	--------	-----

Started

Used 7.24 of 63.56MB

JBroFuzz



Web application fuzzing

■ Problems:

- ▶ Identification of input points
 - HTTP communication analysis
 - Webspidering
 - Search engines
- ▶ Generation of test data
 - Payloads hardcoded in fuzzers
 - Bruteforce
 - Payloads based on patterns
- ▶ Error identification

Error identification

- HTTP response codes
- Analysis of website content
- Comparison of website internal structure
- Time attacks
- Multiple requests
- Analysis of unique data identifying website.
- Logs

Anti-fuzzing

- We can't directly defend against fuzzing!
- Generic defense:
 - ▶ Validation of input data
 - ▶ Application of good programming practices
 - ▶ Ensuring security through all phases of SDLC

Summary

Fuzzing advantages

- Full automatization (in most cases)
- Fuzzers find real vulnerabilities
- Ability to identify bugs which are hard to find by manual testing
- Ability to quickly obtain satisfactory results (first bug)

Fuzzing disadvantages

- Inability to find logical bugs
- Inability to find complex bugs
- Time required for performing test is very hard to specify

Additional information

■ Talks:

- ▶ PyCON 2008
- ▶ SEConference 2009

■ Sites:

- ▶ fuzzing.eu
- ▶ fuzzing.org
- ▶ krakowlabs.com/lof.html



09-10.04.2010

www.seconference.pl

Questions

Thanks for your attention!