



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

- Metody testowania aplikacji
- Zapewnienie bezpieczeństwa aplikacji
- Fuzzing – definicja
- Zastosowanie fuzzingu
- Podział fuzzerów
- Fuzzing a SDLC
- Kto korzysta z fuzzerów?
- Przykładowe fuzzery
- Web application fuzzing
- Podsumowanie

Testowanie aplikacji

■ Popularne testy:

- ▶ Testy jednostkowe
- ▶ Testy funkcjonalne
- ▶ Testy regresyjne
- ▶ Testy wydajnościowe
- ▶ Testy usability

■ Inna klasyfikacja:

- ▶ Whitebox, Graybox, Blackbox

■ Co z bezpieczeństwem tworzonej aplikacji?

- ▶ Bezpieczeństwo na poziomie projektowania, implementacji, testowania oraz wdrażania.

Zapewnianie bezpieczeństwa aplikacji

■ Reagowanie vs zapobieganie

■ Reagowanie:

- ▶ Firewall
- ▶ IDS/IPS
- ▶ Antywirus
- ▶ Mechanizmy uwierzytelniające
- ▶ Skanery podatności (Nessus, Nikto, etc.)
- ▶ Itd.

■ Zapobieganie:

- ▶ **Fuzzing!**
- ▶ Audyt kodu/RE

Fuzzing - definicja

- Fuzzing jest metodą testowania oprogramowania pod kątem występowania luk w bezpieczeństwie oraz nieprzewidzianych reakcji programu, za pomocą częściowo losowych (pseudolosowych) danych.
- Fuzzing najczęściej jest procesem w pełni zautomatyzowanym - „uruchom i czekaj na wyniki”.

Fuzzing – co to znaczy w praktyce?

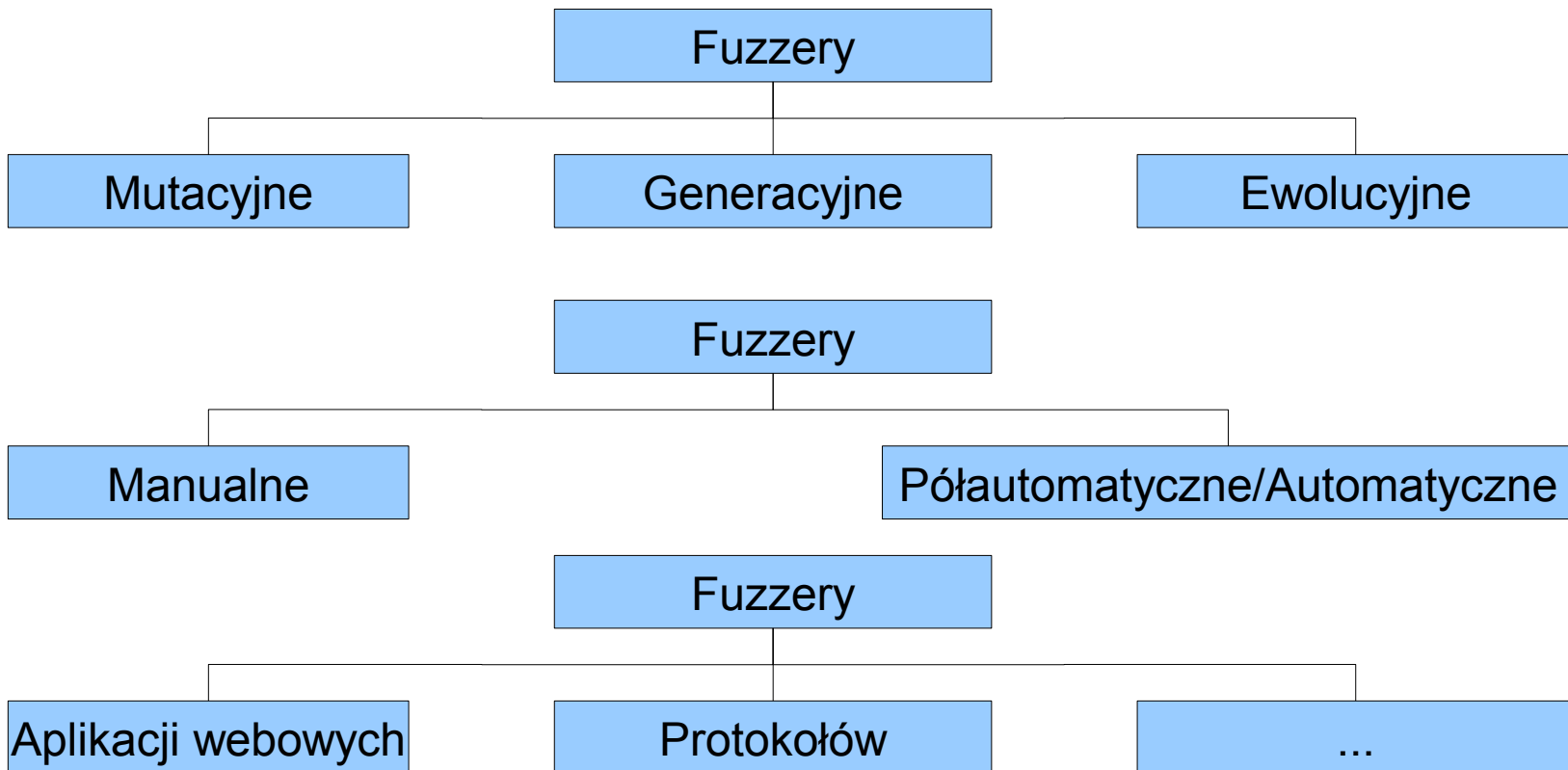
- Fuzzing == Negative testing
- Celem fuzzera jest przekazanie do testowanej aplikacji wadliwych (zbyt długi ciąg znaków, niepoprawne kodowanie, wadliwy format pliku, zła kolejność komunikatów) danych.
- Liczymy na zaakceptowanie wadliwych danych i wystąpienie nieprzewidzianej reakcji programu – DoS, wyświetlenie komunikatu o błędzie, zwiększenie zapotrzebowania na zasoby.
- Naszym celem jest „zepsucie” aplikacji!

Zastosowanie fuzzerów

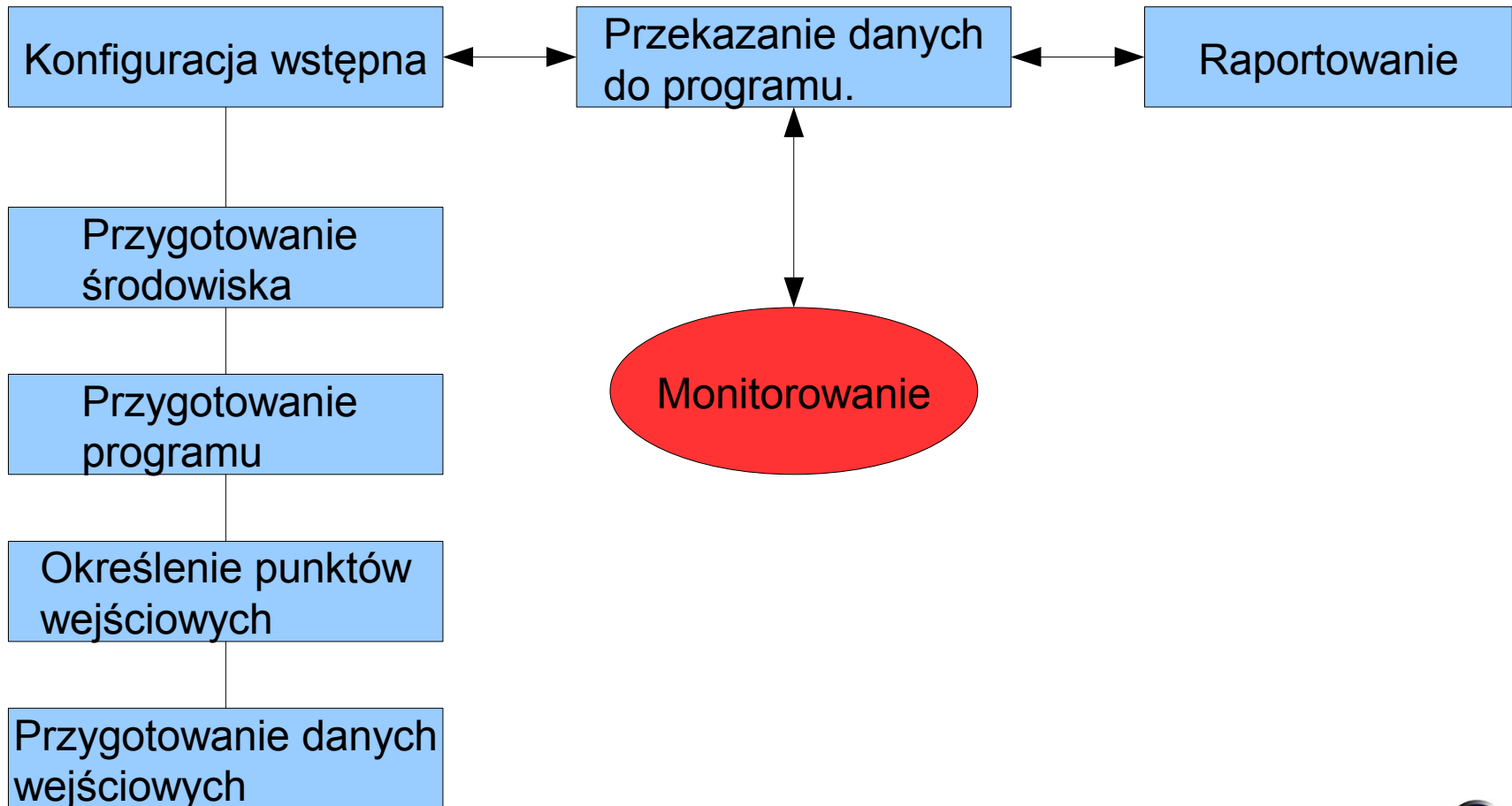
- Aplikacje działające lokalnie
- **Aplikacje webowe**
- Webservice'y
- Aplikacje sieciowe
- Kontrolki ActiveX
- Pliki
- Biblioteki
- ...

Klasyfikacja fuzzerów

- Istnieje wiele kryteriów klasyfikacji fuzzerów
- Przykłady:



Proces fuzzingu



Monitorowanie testowanej aplikacji

- Obserwacja zachowania programu
- Logi systemowe
- Debuggery (!exploitable...)
- Monitory procesów, plików, połączeń sieciowych
- Wirtualizacja (VMWare)
- Modyfikacja kodu źródłowego (dodanie punktów kontrolnych)
- Inne techniki (Valgrind, Guard Malloc)
- Techniki łączone

Process Explorer

The screenshot shows the Process Explorer application window. The title bar reads "Process Explorer - Sysinternals: www.sysinternals.com [apejron\hellsource]". The menu bar includes "File", "Options", "View", "Process", "Find", "Users", and "Help". The main area displays a tree view of processes on the left and a table of process details on the right. The table has columns for "Process", "PID", "CPU", "Description", and "Company Name". The status bar at the bottom shows "CPU Usage: 32.98%", "Commit Charge: 46.59%", "Processes: 114", and "Physical Usage: 67.65%".

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				

Process Monitor

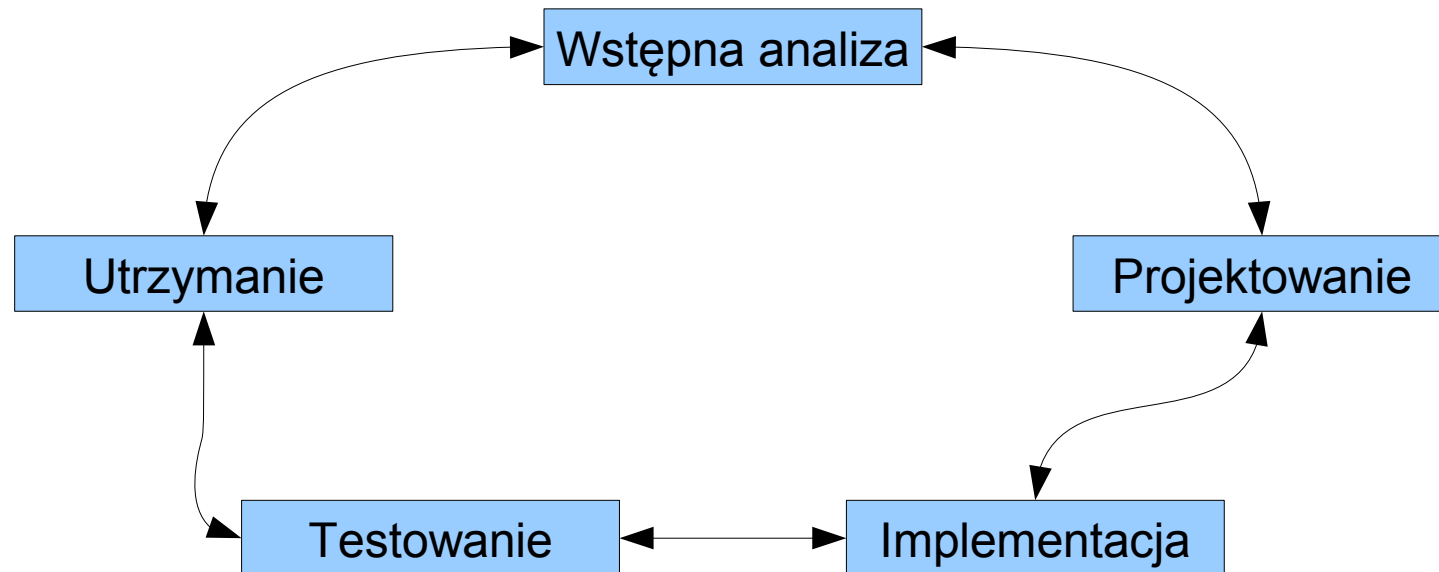
Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

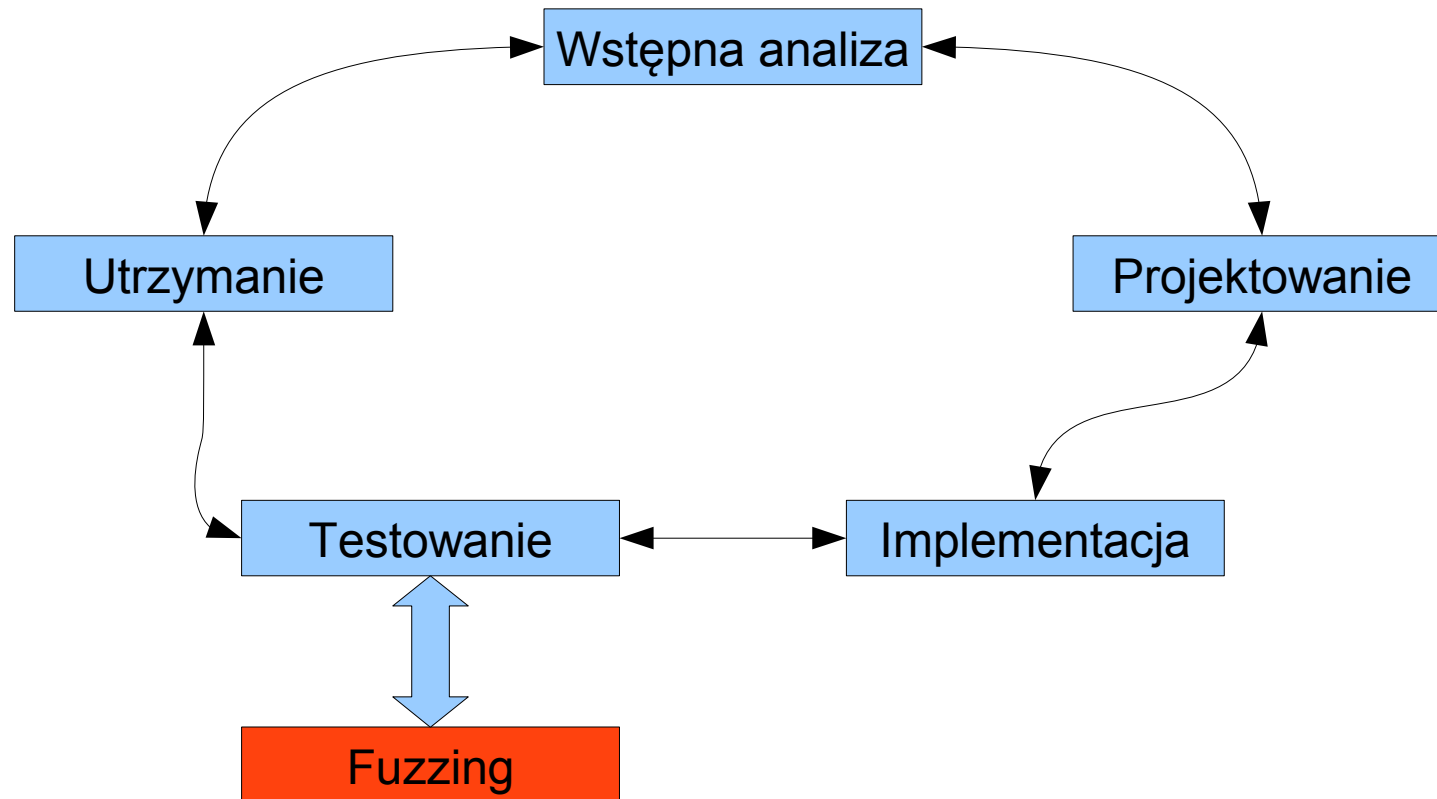
Time ...	Process Name	PID	Operation	Path	Result	Detail
22:06:...	Explorer.EXE	2656	RegQueryKey	HKCU\Software\Classes	SUCCESS	Query: HandleTags, HandleTags: 0x0
22:06:...	Explorer.EXE	2656	RegOpenKey	HKCU\Software\Classes\Applications\...	NAME NOT FOUND	Desired Access: Read
22:06:...	Explorer.EXE	2656	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 a SDLC



Fuzzing a SDLC



Fuzzing a SDLC

- Po opublikowaniu nowej wersji, aplikacja jest testowana przez przygotowane wcześniej fuzzery.
- Wyniki testów weryfikowane są przez testerów i trafiają w ręce programistów.
- W razie wystąpienia błędów programiści poprawiają oprogramowanie.
- Nowy build jeszcze raz przechodzi przez proces fuzzingu.

Kilka przykładów

■ Adobe

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

■ Błędy w 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/>

Przykładowe fuzzery

■ Frameworki:

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

■ Wyspecjalizowane fuzzery:

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

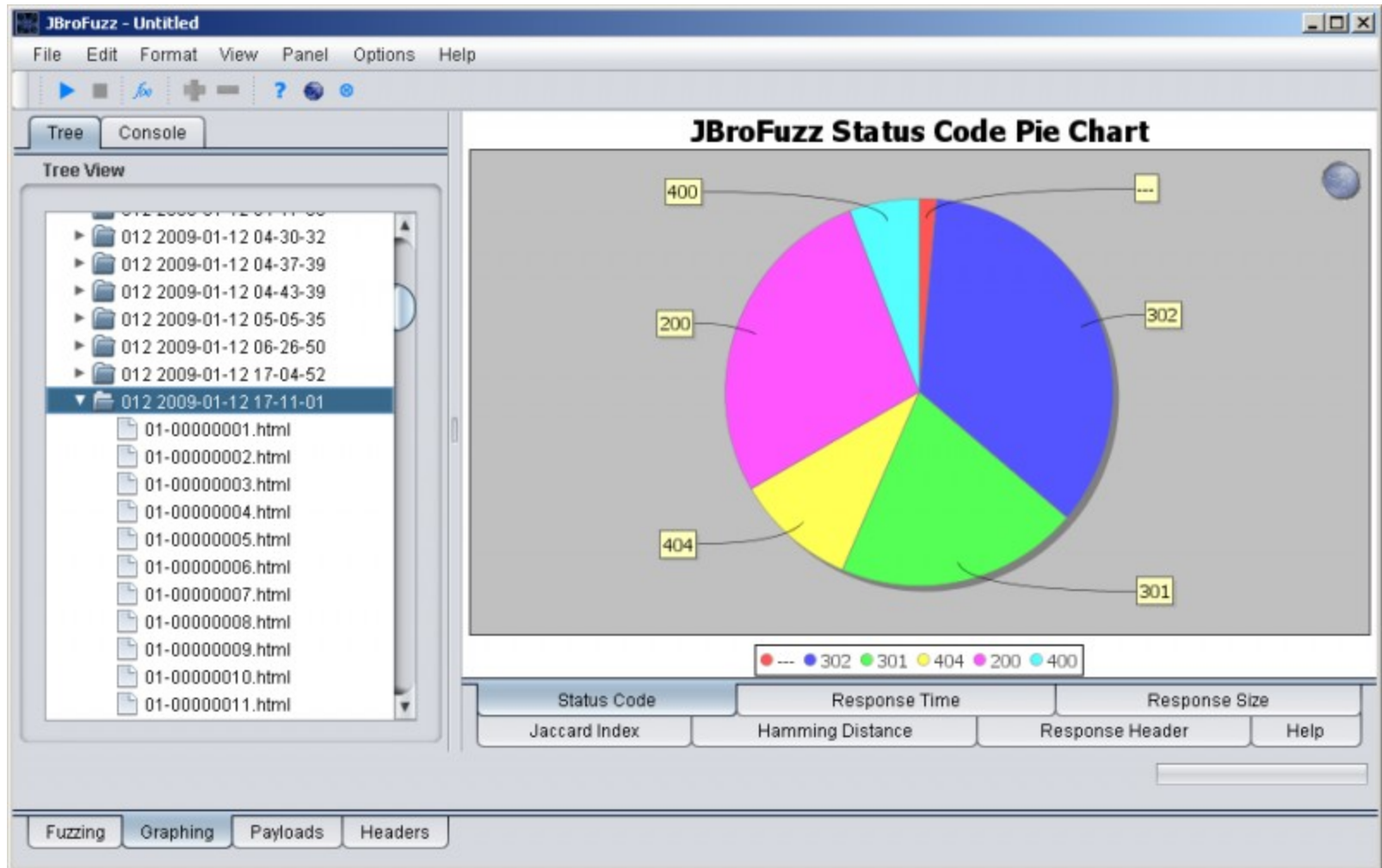
■ Autorskie rozwiązania

WebScarab Fuzzer plugin

The screenshot displays the WebScarab application window with the 'Fuzzer' tab 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 main workspace is divided into several sections:

- Method and URL:** A text box for the Method (currently 'GET') and a larger text box for the URL (currently 'http://localhost:8080/test').
- Version:** A dropdown menu showing 'HTTP/1.0'.
- Header/Value Table:** A table with two columns: 'Header' and 'Value'. It is currently empty.
- Parameters Table:** A table with columns: 'Location', 'Name', 'Type', 'Value', 'Priority', and 'Fuzz Source'. It is currently empty.
- Buttons:** On the right side of the Header/Value and Parameters tables, there are 'Add' and 'Delete' buttons.
- Request Statistics:** Two input fields for 'Total Requests' and 'Current Request', both set to '0'.
- Control Buttons:** Three buttons labeled 'Sources', 'Start', and 'Stop'.
- Request Log Table:** A table with columns: 'ID', 'Date', 'Method', 'Host', 'Path', 'Parameters', 'Status', 'Origin', and 'Tag'. It is currently empty.
- Status Bar:** A green bar at the bottom indicating 'Started' and 'Used 7.24 of 63.56MB'.

JBroFuzz



Fuzzing aplikacji webowych

■ Problemy:

- ▶ Identyfikowanie punktów wejściowych
 - Analiza komunikacji HTTP
 - Webspidering
 - Wyszukiwarki internetowe
- ▶ Generowanie danych testowych
 - Payloady zapisane w programie
 - Bruteforce
 - Generowanie danych na podstawie wzorców
- ▶ Identyfikowanie błędów

Identyfikowanie błędów

- Kody odpowiedzi HTTP
- Analiza treści strony
- Porównywanie struktury strony
- Ataki czasowe
- Wielokrotne zapytania
- Analiza danych jednoznacznie identyfikujących daną stronę
- Logi

Anty-fuzzing

- Nie można bronić się bezpośrednio przed fuzzingiem!
- Obrona ogólna:
 - ▶ Walidacja danych wejściowych
 - ▶ Stosowanie się do dobrych praktyk programistycznych
 - ▶ Dbanie o bezpieczeństwo oprogramowania przez wszystkie fazy SDLC

Podsumowanie

Zalety fuzzingu

- Pełna automatyzacja (w większości wypadków)
- Fuzzery znajdują realne podatności
- Możliwość wykrycia błędów trudnych do znalezienia poprzez manualne testy
- Możliwość szybkiego uzyskania zadowalających wyników (pierwszego błędu)

Wady fuzzingu

- Brak możliwości wykrycia błędu logicznego
- Brak możliwości wykrycia bardzo złożonych podatności (gdzie efekt końcowy jest składową sekwencji operacji)
- Trudny do sprecyzowania czas potrzebny na przeprowadzenie testów

Informacje dodatkowe

■ Prezentacje:

- ▶ PyCON 2008
- ▶ SEConference 2009

■ Strony:

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



09-10.04.2010

www.seconference.pl

Pytania

Dziękuję za uwagę!