HotFuzz

Vedoucí projektu

Mgr. Daniel Toropila <daniel.toropila@mff.cuni.cz>

Řešitelský kolektiv

Dušan Domány, Štěpán Henek, Peter Kmeť, Jan Staněk, Martin Žember

Termín dokončení

15. 8. 2010

Anotace

"Fuzz testing" nebo zkráceně "**fuzzing**" **je metoda testování programů**, při které jsou **strojově generována různá neobvyklá vstupní data**, například pseudonáhodnými mutacemi korektních dat, a je sledováno, zda některá z nich nevyvolají anomální reakci testovaného programu. Jako "**smart fuzzing**" bývá označována varianta této metody, kdy je **při vytváření neobvyklých vstupů využita znalost jejich požadované formy** (souborového formátu, síťového protokolu, syntaxe příkazů atd.), což dovoluje dosáhnout lepšího pokrytí testovaného kódu. Tato výhoda je ovšem vyvážena nutností vytvořit specifický generátor pro každý jednotlivý druh vstupních data a to může být značně náročné.

Cílem projektu je **rozšířit** platformu **Peach Fuzzing Platform** (http://peachfuzzer.com) **o nástroje dovolující snadný "smart fuzzing"** síťových aplikací využívající existujících programů jako náhrady explicitního popisu použitého komunikačního protokolu. **Komplementární síťové aplikace** (např. webový prohlížeč pro testování HTTP serveru) budou použity pro generování korektních vzorků vstupních dat, přesněji řečeno celých korektních konverzací daného protokolu. **Analyzátory síťových protokolů** ("dissectors") obsažené v programu Wireshark (http://www.wireshark.org) budou použity k rozpoznávání formátu zpracovávaných dat.

Výsledné dílo bude schopno:

1. **Spouštět a monitorovat dvě zadané síťové aplikace** (klient a server) instruované k opakovanému provádění určité sekvence či sekvencí operací. (Konkrétní mechanismy potřebné k ovládání aplikací nebudou součástí projektu.)

2. **Vystupovat vůči spuštěným aplikacím jako proxy server**, který přijímá data od klienta a předává je serveru a naopak. Podporováno bude TCP a

UDP. (Předpokládá se, že zúčastněné aplikace budou k použití proxy explicitně instruovány.)

3. **Analyzovat přenášená data** (s využitím výše zmíněných komponent Wiresharku), identifikovat zprávy příslušného síťového protokolu a rozpoznat jejich formát.

4. **Vybírat podle zvolené strategie zprávy určené k fuzzingu** a způsob jejich úpravy. Implementován bude pseudonáhodný výběr zprávy v zvoleném směru v rámci jedné konverzace a pseudonáhodný výběr způsobu, jakým bude zpráva změněna.

5. **Automaticky transformovat obsah a formát vybraných zpráv** na "data model", který používá Peach, a použít Peach k fuzzingu takové zprávy před tím, než bude předána druhé aplikaci. Implementovány budou obecné heuristiky doplňující informace o vzájemných vztazích mezi položkami zprávy (např. jedna položka může určovat délku jiné) podle jejich názvů, případně specifické transformace pro vybrané podporované protokoly.

Další požadavky

- Programovací jazyky Python a C
- Operační systém MS Windows na x86