BS450 Yazılım Test Mühendisliği

Fuzzing

Motivasyon

1990 yılının karanlık ve fırtınalı bir gecesinde Bart Miller (Wisc. Üniversitesi) çevirmeli bir ağ üzerinden çalışmak için oturum açtı.

Fırtınadan dolayı çok fazla gürültü bulunuyordu Bu

gürültü nedeniyle uygulamaları sürekli çöküyordu

Bu durum ona bu fikri verdi…

Motivasyon

Fikir: “**fuzz**” değerlerini ver – İşletim sistemi ve yardımcı programlara saf rastgele değerler ver, ör., **`/dev/random`**`'den gelen gürültü değerleri`

Bozulmasını izle!

Hizmetlerin %25-33’ünün çökmesine neden olur

Hatalardan bazıları, yaygın güvenlik açıklarından yararlananlarla aynıdır (özellikle arabellek aşımları)

Bart Miller et al., “An Empirical Study of the Reliability of UNIX Utilities”

Fuzzing

Yazılım geliştiricileri hata yapar
Hatalar -> buglar -> güvenlik zafiyetleri

Fuzzing, bir programa hatalı biçimlendirilmiş/rastgele girdi göndermeyi ve onu çökertmeyi veya bir güvenlik açığı bulmayı ummayı amaçlamaktadır.

Programları korumak için hataları bulmakta kullanışlıdır

Ama aynı zamanda sistemlere sızmak için hataları bulmak için de

Apple, Firefox gibi birçok tanınmış şirkette/üründe (güvenlik) sorunları bulmak için kullanılır

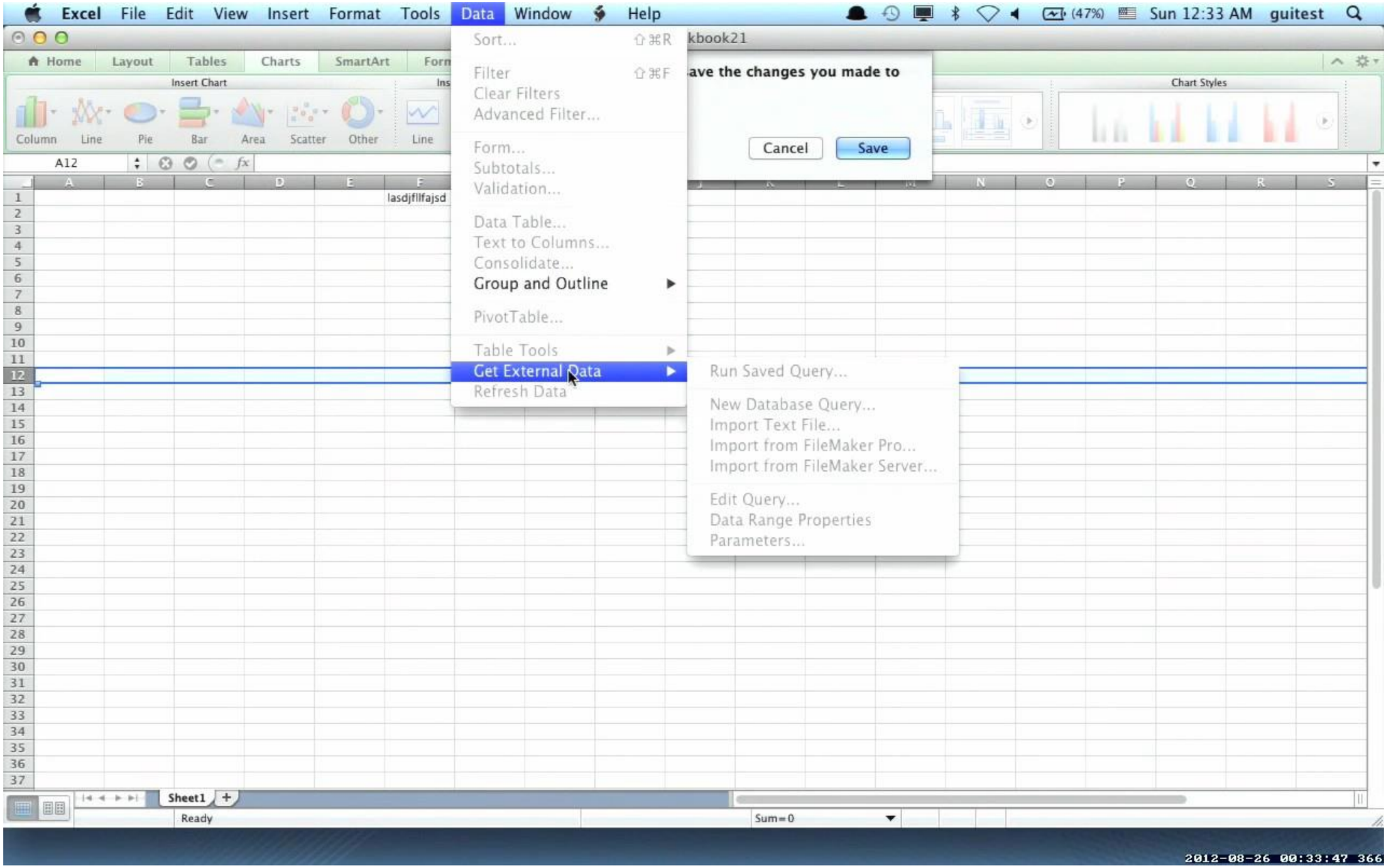
User Testing vs Fuzzing

Kötü şeylerin olmasını bekleyerek programı birden fazla **normal** girdi ile çalıştırın

Amaç: **Normal** kullanıcıların hatalarla karşılaşmasını önlemek

Kötü şeylerin olmasını bekleyerek programı birden fazla **anormal** giriş ile çalıştırın

Hedef: **Saldırganların istismar** edilebilir hatalarla karşılaşmasını önlemek


Excel File Edit View Insert Format Tools Data Window Help
Sun 12:33 AM guitest
Sort... ⇧⌘R
Filter ⇧⌘F
Clear Filters
Advanced Filter...
Form...
Subtotals...
Validation...
Data Table...
Text to Columns...
Consolidate...
Group and Outline
PivotTable...
Table Tools
Get External Data
Refresh Data
Run Saved Query...
New Database Query...
Import Text File...
Import from FileMaker Pro...
Import from FileMaker Server...
Edit Query...
Data Range Properties
Parameters...
ave the changes you made to
Cancel
Save
Home
Layout
Tables
Charts
SmartArt
Insert Chart
Column
Line
Pie
Bar
Area
Scatter
Other
Chart Styles
A12
lasdjfllfajsd
Sheet1
Ready
Sum=0
2012-08-26 00:33:47 366

Fuzzing

Otomatik olarak **rastgele** (hatalı biçimlendirilmiş) test senaryoları oluşturun
Uygulamayı hatalara karşı **izleyin**
Girdilerin doğası farklı olabilir

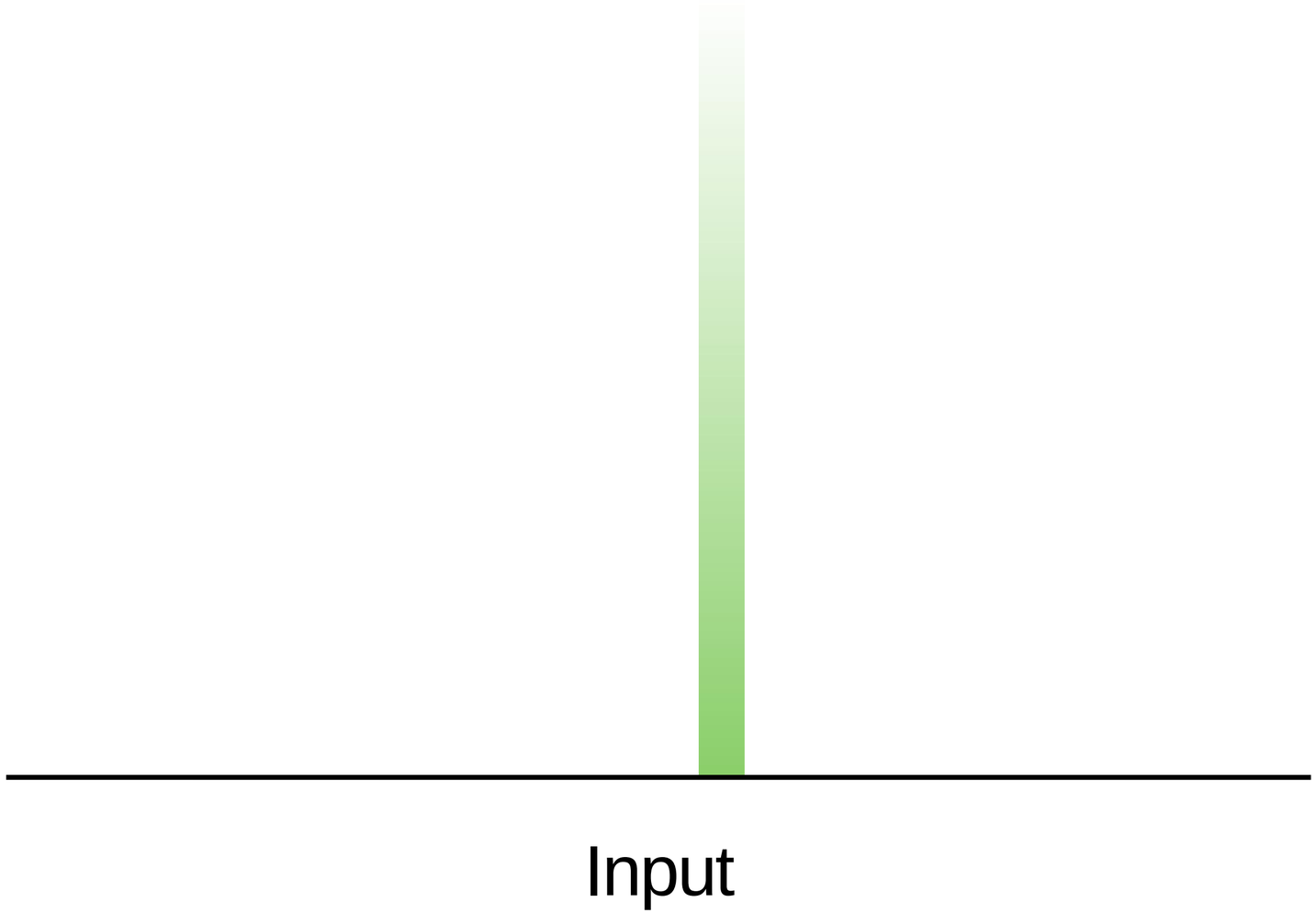
Dosyalar: .pdf, .png, .wav, .mpg, .json, etc

Network based: http, SOAP, SNMP

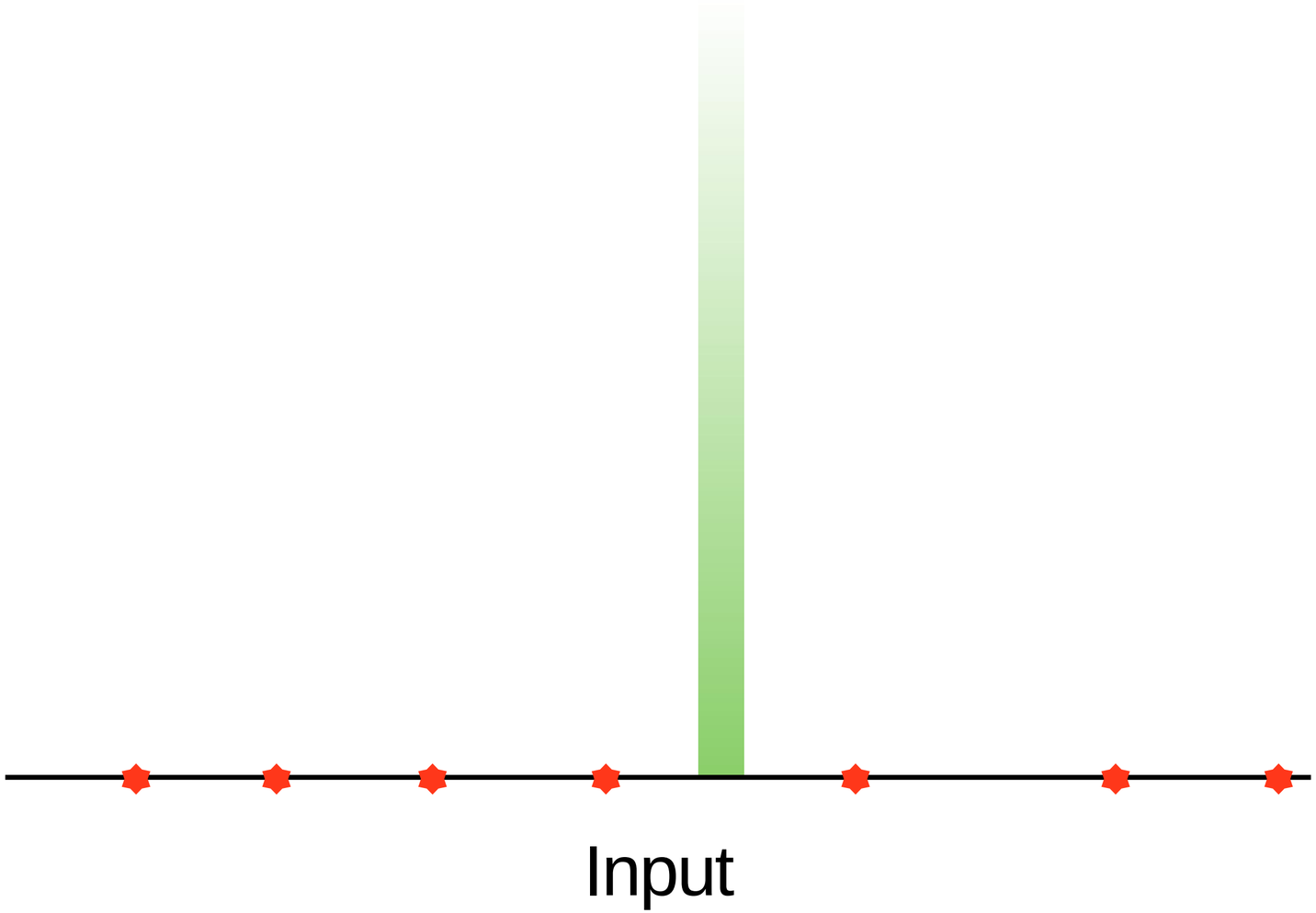
Hata Bulma Olasılığı?

Input

Hata Bulma Olasılığı?



Hata Bulma Olasılığı?



Fuzzing Nasıl Yapılır?

Çok uzun girdiler, çok kısa girdiler, boş girdiler, …

Tam sayıların minimum/maksimum değerleri, sıfır ve negatif değerler

Hataları tetiklemesi muhtemel özel değerler, karakterler veya anahtar kelimeler; örneğin boş değerler, yeni satırlar veya dosya sonu karakterleri

String ifade formatlayıcısı %s %x %n

Noktalı virgül, eğik çizgi ve ters eğik çizgi, tırnak işaretleri, yazdırılamayan karakterler,…

Uygulamaya özel anahtar kelimeler, örneğin halt, return, DROP TABLES,…

Şablon Tabanlı Fuzzing

Mutasyona dayalı olarak da bilinir

Karmaşık girdi gerektiren programlar rastgele girdiyle düzgün bir şekilde test edilemez

Ör: JavaScript yorumlayıcısının test girdileri olarak geçerli JavaScript programlarına ihtiyacı var

Rastgele girdiler yerine mevcut, geçerli girdileri **değiştirin (mutate)**

Mutasyonlar rastgele olabilir veya sezgisel yöntemler tarafından yönlendirilebilir

Kapsam Tabanlı Fuzzing

Fuzzer örneğin yeni bir dalı kapsayan geçerli, yararlı bir girdi oluşturursa bunu daha sonraki mutasyonlar için bir başlangıç noktası olarak kullanın.

White-box fuzzing olarak da bilinir

Grammar Tabanlı Test

Karmaşık **metin** girdileri nasıl oluşturulur?

Klasik uygulama: Derleyicilerin test edilmesi

Günümüzde: Web uygulamaları, yorumlayıcılar, ayrıştırıcılar, …XML, JSON vb. kullanan her şey.

Microsoft'un **SAGE** aracının (Godefroid ve diğerleri, 2008a) şirketin milyonlarca dolar değerindeki yazılım hatalarından kurtardığı bildiriliyor.

Context-free Grammars

Finite set of *terminals*

Finite set of *non-terminals*

Finite set of *rules*

Each *rule*: *Non-terminal* → List of *terminals* and *non-terminals*

Starting rule

Örnek Grammar

```
expr . expr op expr
expr . ( expr )
expr . - expr
expr . id
op . +
op . -
op . *
op . /
op . ↑
```

Türetme

Üretim yeniden yazma kuralı olarak yorumlanabilir

Sol taraftaki terminal olmayan, sağ taraftaki dize ile değiştirilir.

E → -E → -(E) → -(id)

Değiştirme serisi = türetme

Her türetme adımında iki seçenek bulunur:

- Hangi terminal olmayanın değiştirileceği
- Değiştirilen terminal olmayan için hangi alternatif kullanılacak

Grammar Tabanlı Test

Test üretme → grammar'e göre yeniden yazma

Başlangıç sembolüyle başla

Sağdaki terminal olmayan bir kuralı, soldaki terminal olmayan kuralla değiştirin

Yalnızca terminaller kalana kadar tekrarlayın

Sonuç test girdisidir

```
expr . expr op expr
expr . ( expr )
expr . - expr
expr . id
op . +
op . -
op . *
op . /
op . ↑
```

$$E \rightarrow -E \rightarrow -(E) \rightarrow -(E+E) \rightarrow -(E+id) \rightarrow -(id+id)$$

Problemler

Context-free grammars → yineleme

Yineleme →**Sonsuz sayıda** olası girdi

Sonraki kuralla değiştirilecek kural hangisi?

Grammar Annotation

Yineleme derinliğini sınırla

Tekrar sayısını sınırla

Parsa ağacı derinliğini sınırla

```
{count 3} zeros;
          zeros ::= '0';
          zeros ::= '0' zeros;
```

Kombinasyonel Problem

Ör, VoIP software testi

- Caller, VoIP server, client
- CallerOS: Windows, Mac
- ServerOS: Linux, Sun, Windows
- CalleeOS: Windows, Mac

Grammar ile Aynı Problem

Gramerleri test etmek için covering array kullanmak mümkündür

```
Call ::= CallerOS ServerOS CalleeOS;
CallerOS ::= ‘Mac’;
CallerOS ::= ‘Win’;
ServerOS ::= ‘Lin’;
ServerOS ::= ‘Sun’;
ServerOS ::= ‘Win’;
CalleeOS ::= ‘Mac’;
CalleeOS ::= ‘Win’;
```

Araçlar

LibFuzzer (https://llvm.org/docs/LibFuzzer.html),

AFL (https://github.com/google/AFL)

AFL++ (https://github.com/AFLplusplus)

Spike (https://www.kali.org/tools/spike/)

Peach (https://wiki.mozilla.org/Security/Fuzzing/Peach)

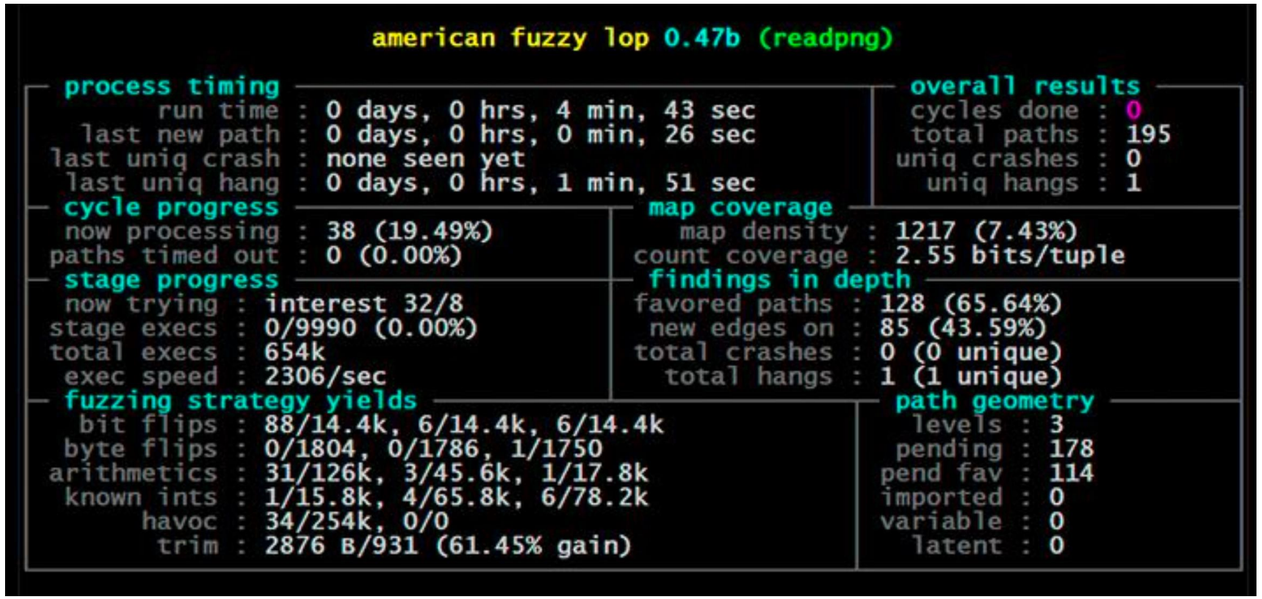
OneFuzz (https://github.com/microsoft/onefuzz)

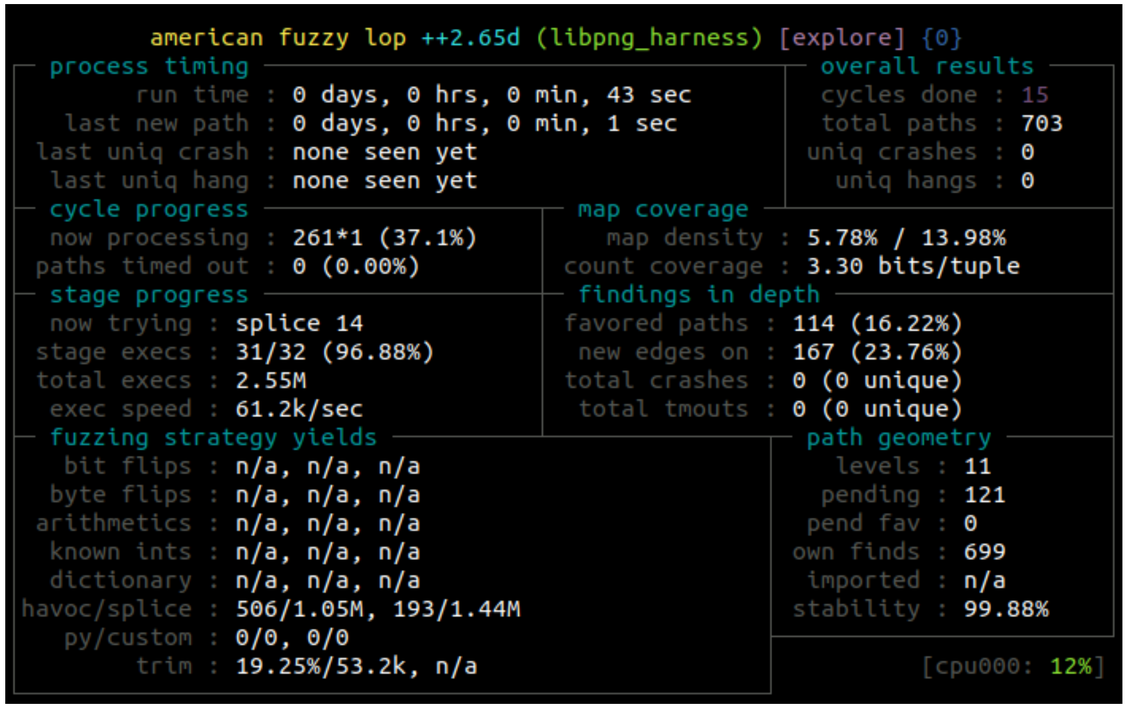
HonggFuzz (https://github.com/google/honggfuzz)

AFL (American Fuzzy Lop)++

Geri bildirime dayalı fuzzer

Dosyayı değiştir – Kapsamı artırıyor mu? Sakla; yoksa sil





Özet

Fuzzing gibi basit test teknikleri çok etkili olabilir

Uygulanması nispeten kolay
Maliyetli olabilir – ne zaman durulacak? – ve kapsam açısından değerlendirmek zor
Pek çok araç var örneğin:

Muhteşem kaynak: The Fuzzing Book (https://www.fuzzingbook.org/)