
Albeiro José Cuadrado Machado

Juan Camilo Acevedo Bedoya

Estudiantes MISO

Universidad de los Andes

02 de Mayo del 2021

Reporte de resultados: Pruebas de reconocimiento con monkey-cypress y RIPuppet

Semana 4: Pruebas automatizadas de software

VISIÓN GENERAL

En este reporte de resultados se presenta información relevante de las pruebas de reconocimiento realizadas a la aplicación bajo pruebas (ABP) Ghost en su versión 3.3.0, usando monkeys y rippers.

SISTEMA DE REGISTROS DE INCIDENCIAS

Para acceder al sistema de registros de incidencias (Issues tracker) de **Github**, acceder al siguiente enlace <https://github.com/albcm-uniandes/Issues-tracker-module4/issues> (**NO ES NECESARIO UN USUARIO O CONTRASEÑA YA QUE EL REPOSITORIO ES PUBLICO**).

MONKEY-CYPRESS

Se utiliza la herramienta monkey cypress en sus dos versiones Versión 1: "Monkey" aleatorio y Versión 2: "Smarter Monkey" aleatorio con heurística, en ambas versiones se necesitó realizar refactor de las herramientas para permitir el ingreso al panel administrativo de la ABP, adicionalmente se corrigió error con la instrucción que permite el scroll, se detalla a continuación:

Modificaciones

```
cy.visit(url).then((win)=>{  
  cy.get('[id^=login]').within(() => {  
    cy.get('.email.ember-text-field.gh-input.ember-view').type('acevedobedoya@gmail.com')  
    cy.get('.password.ember-text-field.gh-input.ember-view').type('abcd1234*+')  
    cy.get('.login.gh-btn.gh-btn-blue.gh-btn-block.gh-btn-icon.ember-view').click()  
    cy.log('Login successful')  
  })  
})
```

Instrucciones para el ingreso al ABP

`cy.scrollTo(curX, curY,{ensureScrollable: false})`

Modificación a la instrucción de scroll

Versión 1: Monkey aleatorio

Ver videos de ejecución:

https://www.dropbox.com/sh/r669mo3trt6n9hn/AACPbyYM6rcn-stgFjPj7wwEa/Monkey?dl=0&preview=monkey.js.mp4&subfolder_nav_tracking=1

Esta versión de la herramienta se caracteriza por tener eventos basados completamente en la aleatoriedad, ignorando mejoras que podrían surgir a partir de la comprensión de los mismos elementos que componen una página web.

```
PS D:\monkey-cypress> cypress run -C monkey-config.json
```

(Run Starting)

```
Cypress: 7.2.0
Browser: Electron 89 (headless)
Specs: 1 found (monkey.js)
```

```
Running: monkey.js (1 of 1)
[cypress-log-to-output] Warning: An unsupported browser family was used, output will not be logged to console: chromium
```

```
Ghost 3.3.0 under monkeys
Log started
Seed: 1234
Logged #Selector focus (hover)
Logged #Page navigation (Reload)
Logged #Key press
Logged #Selector focus (hover)
Logged #Special key press
Logged #Selector focus (hover)
Logged #Key press
Logged #Special key press
Logged #Special key press
Logged #Page navigation (forward)
Logged #Key press
Logged #Scroll event (horizontal bk)
Logged #Random click
Logged #Random click
Logged #Selector focus (hover)
Logged #Scroll event (pg down)
Logged #Scroll event (horizontal fw)
Logged #Random double click
Logged #Special key press
Logged #Special key press
Logged #Random click
Logged #Page navigation (back)
Logged #Random double click
Logged #Scroll event (horizontal bk)
Logged #Special key press
Logged #Page navigation (back)
```

Setup del proceso

```
Logged #Special key press
Logged #Special key press
Logged #Page navigation (forward)
Logged #Key press
Logged #Scroll event (horizontal bk)
Logged #Random click
Logged #Random click
Logged #Selector focus (hover)
Logged #Scroll event (pg down)
Logged #Scroll event (horizontal fw)
Logged #Random double click
Logged #Special key press
Logged #Special key press
Logged #Random click
Logged #Page navigation (back)
Logged #Random double click
Logged #Scroll event (horizontal bk)
Logged #Special key press
Logged #Page navigation (back)
Logged #Selector focus (hover)
Logged #Random click
Logged #Special key press
Logged #Key press
Logged #Scroll event (horizontal bk)
Logged #Scroll event (horizontal fw)
Logged #Page navigation (Reload)
Logged #Selector focus (hover)
Logged #Random click
Logged #Page navigation (forward)
Logged #Selector focus (hover)
Logged #Selector focus (hover)
Logged #Page navigation (back)
Logged #Random click
Logged #Page navigation (forward)
Logged #Key press
Logged #Key press
Logged #Special key press
Logged #Key press
Logged #Key press
Logged #Random click
Logged #Scroll event (horizontal bk)
Logged #Random double click
Logged #Scroll event (pg down)
Logged #Scroll event (horizontal fw)
Finished logging
✓ visits Ghost 3.3.0 and survives monkeys (31405ms)
```

Eventos disparados

```
PS D:\monkey-cypress> cypress run -C monkey-config.json
```

```
(Run Starting)
```

```
Cypress: 7.2.0
Browser: Electron 89 (headless)
Specs: 1 found (monkey.js)
```

```
Running: monkey.js (1 of 1)
[cypress-log-to-output] Warning: An unsupported browser family was used, output will not be logged to console: chromium
```

```
Ghost 3.3.0 under monkeys
Log started
Seed: 1234
Logged #Selector focus (hover)
Logged #Page navigation (Reload)
Logged #Key press
Logged #Selector focus (hover)
Logged #Special key press
Logged #Selector focus (hover)
Logged #Key press
Logged #Special key press
Logged #Special key press
Logged #Page navigation (forward)
Logged #Key press
Logged #Scroll event (horizontal bk)
Logged #Random click
Logged #Random click
Logged #Selector focus (hover)
Logged #Scroll event (pg down)
Logged #Scroll event (horizontal fw)
Logged #Random double click
Logged #Special key press
Logged #Special key press
Logged #Random click
Logged #Page navigation (back)
Logged #Random double click
Logged #Scroll event (horizontal bk)
Logged #Special key press
Logged #Page navigation (back)
```

Resumen de ejecución

```

{...} monkey-config.json > ...
1  {}
2      "projectId": "TSDL-Monkey-with-cypress",
3      "baseUrl": "http://localhost:2368/ghost/#/signin",
4      "env": {
5          "appName": "Ghost 3.3.0",
6          "events": 50,
7          "delay": 300,
8          "seed": 1234,
9          "pctClicks": 19,
10         "pctScroll": 17,
11         "pctSelectors": 16,
12         "pctKeys": 16,
13         "pctSpKeys": 16,
14         "pctPgNav": 16
15     },
16     "integrationFolder": "./cypress/integration/monkey",
17     "pluginsFile": "./cypress/plugins/index.js",
18     "pageLoadTimeout": 120000,
19     "defaultCommandTimeout": 120000,
20     "testFiles": "monkey.js",
21     "videosFolder": "./results"
22 }

```

Configuración

Versión 2: "Smarter Monkey"

Video de ejecución:

https://www.dropbox.com/sh/r669mo3trt6n9hn/AAB6fHz_HbCfRJqtGGSWgMiPa/Smart-monkey/Prueba%2002?dl=0&preview=smart-monkey.js.mp4&subfolder_nav_tracking=1

Esta segunda versión surge a causa de varias incomodidades que puede llegar a causar la versión más aleatoria, como la alta cantidad de eventos inválidos, o la baja probabilidad de que una posición aleatoria de la pantalla contenga un elemento que acepte ciertos eventos, entre otras. Los cambios que se implementan sobre esta versión corresponde, sobre todo, a la inclusión de selectores. Esto quiere decir que algunas de las operaciones ahora se realizarán en función del tipo de contenido de la aplicación web. Si bien esto puede causar que la prueba sea más sesgada, permite reducir el porcentaje de error y realizar una prueba con mayor interacción.

```
{-} smart-monkey-config.json > ...
1  {
2    "projectId": "TSDL-Smart-Monkey-with-cypress",
3    "baseUrl": "http://localhost:2368/ghost/#/signin",
4    "env": {
5      "appName": "Ghost 3.3.0",
6      "events": 50,
7      "delay": 300,
8      "seed": 1234,
9      "pctClicks": 12,
10     "pctScroll": 12,
11     "pctSelectors": 12,
12     "pctKeys": 12,
13     "pctSpKeys": 12,
14     "pctPgNav": 12,
15     "pctBwChaos": 12,
16     "pctActions": 16
17   },
18   "integrationFolder": "./cypress/integration/monkey",
19   "pluginsFile": "./cypress/plugins/index.js",
20   "pageLoadTimeout": 120000,
21   "testFiles": "smart-monkey.js",
22   "videosFolder": "./results"
23 }
```

Configuración

```
PS D:\monkey-cypress> cypress run -C smart-monkey-config.json
```

=====

(Run Starting)

Cypress:	7.2.0
Browser:	Electron 89 (headless)
Specs:	1 found (smart-monkey.js)

Running: smart-monkey.js (1 of 1)
[cypress-log-to-output] Warning: An unsupported browser family was used, output will not be logged to console: chromium

Ghost 3.3.0 under smarter monkeys
Log started
Seed: 12345
Logged #Special key press
Logged #Action: click anchor
Logged #Special key press
Logged #Special key press
Logged #Key press
Logged #Viewport change
Logged #Selector focus (hover)
Logged #Scroll event (pg down)
Logged #Special key press
Logged #Selector focus (hover)
Logged #Special key press
Logged #Random click
Logged #Chaos: Clear cookies
Logged #Action: click anchor
Logged #Action: click anchor
Logged #Selector focus (hover)
Logged #Action: click anchor
Logged #Random double click
Logged #Selector focus (hover)
Logged #Special key press
Logged #Chaos: Clear cookies

Setup del proceso

```

Logged #Random click
Logged #Page navigation (back)
Logged #Page navigation (back)
Logged #Viewport change
Logged #Page navigation (back)
Logged #Chaos: Clear cookies
Logged #Page navigation (back)
Finished logging
  ✓ visits Ghost 3.3.0 and survives smarter monkeys (27070ms)

1 passing (27s)

(Results)

Tests:      1
Passing:    1
Failing:    0
Pending:    0
Skipped:    0
Screenshots: 0
Video:      true
Duration:   27 seconds
Spec Ran:   smart-monkey.js

(Video)
- Started processing: Compressing to 32 CRF
- Finished processing: D:\monkey-cypress\results\smart-monkey.js.mp4 (13 seconds)

Compression progress: 100%

=====

(Run Finished)

Spec                                Tests  Passing  Failing  Pending  Skipped
✓ smart-monkey.js                   00:27    1        1        -        -        -
✓ All specs passed!                 00:27    1        1        -        -        -

```

Resumen del proceso

Modificaciones

- Comentamos el evento reload de la herramienta ya que presenta excepciones
- Agregamos awaits necesarios
- Cambios en la semilla

RIPUPPET

El proyecto RIPuppet fue desarrollado con el lenguaje de programación JavaScript. En particular, se hizo uso de la librería Playwright para ejecutar las pruebas de extremo a extremo (E2E) sobre un sitio web. Esta librería ofrece funcionalidades suficientes para interactuar con el cuerpo DOM de una página web desplegada en una instancia de alguno de los navegadores Firefox, Chrome o WebKit.

Además, permite una fácil integración con otras instrucciones y librerías de JavaScript, dado que no interfiere con el manejo del *event loop*. En particular, se incluyó la librería Faker para generar datos aleatorios que serían utilizados como parte de los campos de entrada y se incluyó la librería FileSystem (o fs) para escribir los archivos temporales y aquellos que finalmente harán parte de los reportes.

La forma en que RIPuppet desarrolla la prueba consiste en una exploración recursiva de los estados de la aplicación web guiada por la interacción con los elementos de la página web que ofrecen funcionalidad al usuario.

Un estado de la aplicación corresponde a una vista que se activa para el usuario según las interacciones que haya tenido con ella. Esto no solo se traduce en las distintas rutas o URL de la aplicación web, sino a cualquier cambio que normalmente induciría un usuario y se ve reflejado en el DOM.

El primer paso de la herramienta consiste en la lectura de los parámetros de configuración. Luego de esto, para cada uno de los navegadores, se genera un contexto de ejecución y unos directorios que almacenarán los resultados de la prueba.

Una vez todo está listo, se llama la ejecución recursiva. Cuando esta finaliza, se genera el reporte correspondiente.

Modificaciones

```
{-} config.json > ...
1  {
2    "url": "http://localhost:2368/ghost/#/signin",
3    "headless": true,
4    "depthLevels": 5,
5    "inputValues": true,
6    "values": {"ember8": "acevedobedoya@gmail.com",
7              "ember10": "abcd1234*+"
8    },
9    "browsers": ["chromium", "webkit", "firefox"]
10 }
```

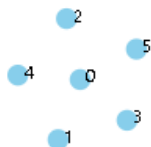
Configuración

Resultados

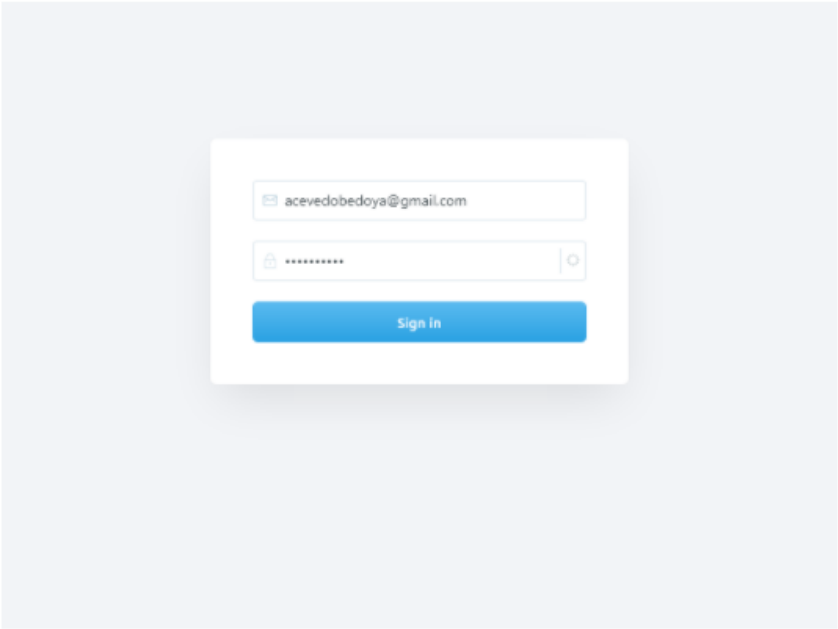
Screenshots y webpages visitadas por el ripper

https://www.dropbox.com/sh/r669mo3trt6n9hn/AAARNXFHABb-4Wk6gSGODN3ma/RIPPuppet?dl=0&subfolder_nav_tracking=1

Visited webpages



State #1



Element type
State
URL
http://localhost:2368/ghost#/signin
Errors found
0

Errors

CONCLUSIONES

- Podemos destacar como **pros** de utilizar monkey testing y específicamente al utilizar la herramienta **monkey-cypress**
 - Son económicas, al ser aleatorios los eventos producidos, solo necesitamos disponer de una máquina con recursos suficiente para levantar las herramientas, para conseguir resultados de las pruebas de reconocimiento
 - Su rápida ejecución es otra de sus ventajas, estas herramientas usan librerías especializadas que conocen el comportamiento del navegador permitiendo aprovechar eficientemente los recursos de estos mismos y saltar eventos dispendiosos de la GUI
 - Al seguir el principio de los monos dactilográficos, nos brindan información acerca de los casos atípicos que un humano podría omitir o evitar ser influenciado por los sesgos
 - Agnósticos al negocio, es decir, no existe necesidad de conocer la naturaleza del ABP, los monkeys realizarán su trabajo independientemente de eso.
 - Pueden ser utilizados como pruebas de desempeño, al tener una rápida ejecución, posibilidad de paralelismo y
- Podemos destacar como **contras** de utilizar rippers y específicamente al utilizar la herramienta **monkey-cypress**:
 - Al ser procesos de fuerza bruta, se les dificulta pasar verificaciones como CAPTCHA o al ingresar múltiples veces operaciones fallida, puede ser bloqueada la sesión del monkey
 - En ocasiones el monkey puede quedarse en la misma ventana por mucho tiempo, entrar en loops o quedarse detenido en ventanas que contienen entradas dinámicas de datos
 - El índice de eventos inválidos es muy alta, debido a la pobre heurística que contiene la herramienta
 - La herramienta necesita refactor en muchas de sus instrucciones, por ejemplo al hacer scroll o pasar una ventana de logueo
- Podemos destacar como **pros** de utilizar rippers y específicamente al utilizar la herramienta **RIPuppet**
 - Nos brinda lo que no nos brindan los monkeys, trazabilidad de los estados visitados por medio de árboles o grafos
 - Brinca una heurística eficiente a los procesos, al tener memoria del proceso
 - Es más fácil para la herramienta encontrar los componentes vs monkey-cypress
- Podemos destacar como **contras** de utilizar monkey testing y específicamente al utilizar la herramienta **RIPuppet**

- Se necesita constantemente extender y actualizar la herramienta en muchas de sus instrucciones, es decir implica un costo de mantenimiento.
 - No es infalible ante escenarios que requieren pre condiciones, como por ejemplo al ingresar a editar un post necesita previamente crearlo
- Se encontraron pocas incidencias por el proceso de reconocimiento debido al poco tiempo de ejecución, la idea principal de estas herramientas aleatorias es preparar escenarios de larga duración, el cual se escapa del alcance de esta prueba de conceptos
- Es importante acompañar estas herramientas con mecanismos de trazabilidad, logueo y alerta de incidencias.
- Como conclusión final, las herramientas aquí mostradas, nos demuestran que este tipo de pruebas son de gran utilidad, fácil de automatizar y económicas, por lo cual deben ser consideradas para ampliar la cobertura de calidad de los sistemas bajo pruebas