Una breve introducción a la Decompilación y sus enfoques

Bratti Juan, Herrador Emanuel N., Scavuzzo Ignacio

github.com/helcsnewsxd/decompilation-report

INTRODUCCIÓN

Decompilador: Programa que dado un ejecutable (en binario) obtiene un programa en lenguaje de alto nivel (HLL) que tenga la misma función.

- *Decompilation of Binary Programs* de Cristina Cifuentes y K. John Gough
  - **dcc** (estático)
    - Objetivos, desafíos y arquitectura
- *BinRec: Dynamic Binary Lifting and Recompilation*
  - **BinRec** (dinámico)
- Herramientas actuales
  - **Ghidra** y un caso de estudio

CONTEXTO

OBJETIVOS

Correctitud sintáctica

Equivalencia semántica

APLICACIONES

Paso intermedio para la re-compilación

Mantenimiento y migración de software

Refactorización de código

DESAFIOS

Datos e instrucciones indistinguibles

Subrutinas presentes

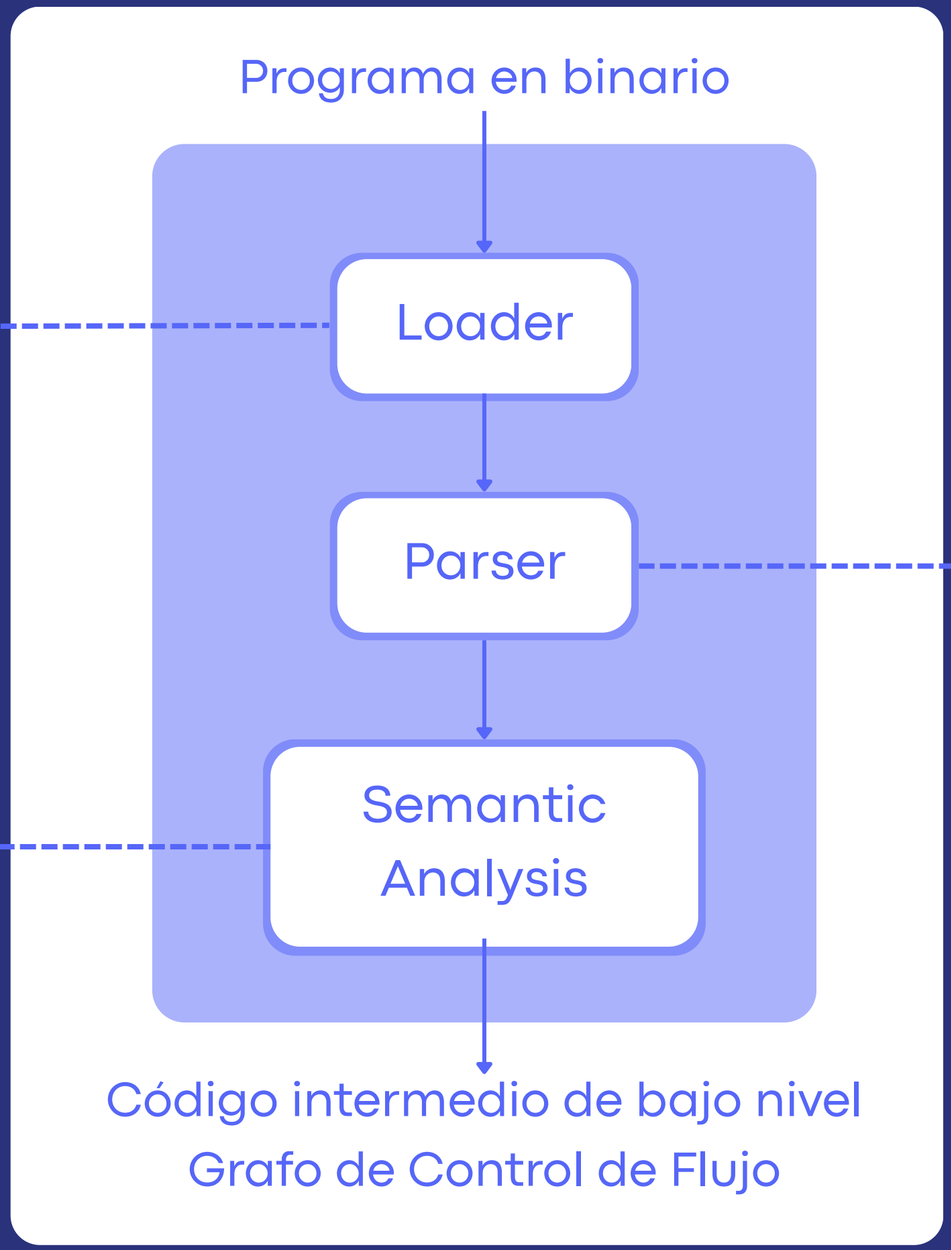
Código de propietario

ARQUITECTURA DE UN DECOMPILADOR

- **Front-end**
  - Depende de la arquitectura del procesador
  - Tareas de desensamblado y análisis del binario
- **Máquina de Decompilación Universal**
  - Núcleo del decompilador
  - Abstracción y estructuración
  - Independiente del procesador y del HLL*
- **Back-end**
  - Generación del código HLL
  - Depende del HLL

* HLL: HIgh Level Language

FRONT-END



Carga binario en memoria

Análisis de idioms*

neg dx
neg ax ⟶ neg dx: ax
sbb dx, 0

Propagación de tipos

Optimización del GCF

Conversión Secuencial

Binario → Assembler → Assembler mnemónicos

Construcción del Grafo de Control de Flujo

* patrón típico de instrucciones de bajo nivel que representa una operación semántica conocida

MÁQUINA DE DECOMPILACIÓN UNIVERSAL

Código intermedio de bajo nivel
Grafo de Control de Flujo

Análisis de Flujo de Datos

Análisis de Flujo de Control

Código intermedio de alto nivel
Grafo de Control de Flujo estructurado

C.I. bajo nivel → C.I. alto nivel

Eliminación de registros temporales

Eliminación de código con flags

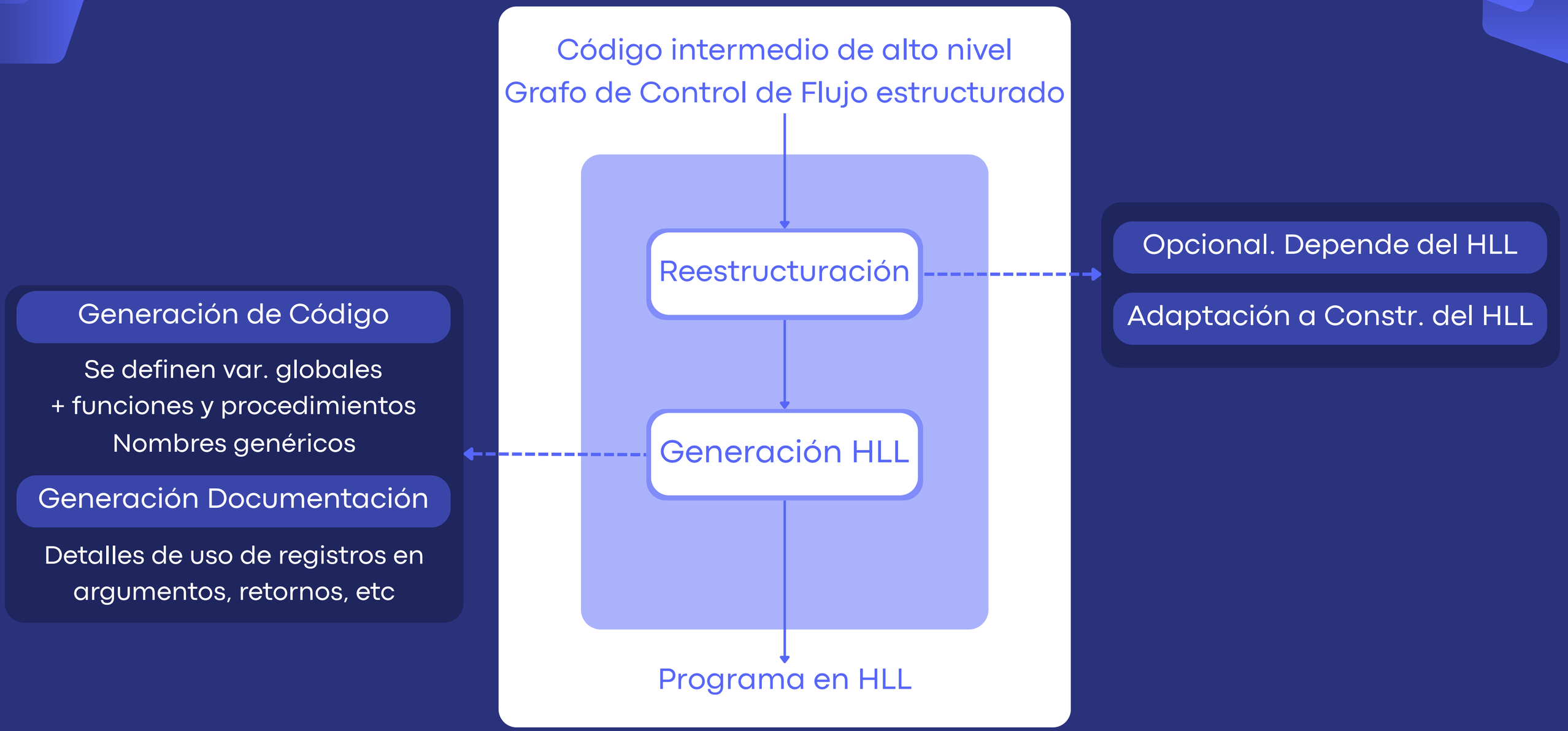
```
cmp ax, bx ; define flags SF,ZF,CF
jg labZ    ; usa flags SF,ZF
     ↓
JCOND (ax > bx)
```

Reestructuración del GCF

Conversión a estructuras de control genéricas
if ... then ... else, while(), repeat ... until()

Condiciones compuestas

BACK-END


Código intermedio de alto nivel
Grafo de Control de Flujo estructurado
Reestructuración
Generación HLL
Programa en HLL
Opcional. Depende del HLL
Adaptación a Constr. del HLL
Generación de Código
Se definen var. globales
+ funciones y procedimientos
Nombres genéricos
Generación Documentación
Detalles de uso de registros en
argumentos, retornos, etc

LIMITACIONES DEL ENFOQUE TRADICIONAL

El enfoque tradicional solamente construye el lenguaje de alto nivel a partir de construcciones intermedias como los CFG y los lenguajes HLL.

AMBIGUEDADES

No se diferencia bien entre lo que es un dato y lo que es código. Malinterpretación de bytes!

FLUJO INDIRECTO

Los decomp. estáticos no son buenos manejando saltos indirectos. Las heurísticas no son suficientes

USO DE LIBRERÍAS

Se centran en el código máquina del programa original, sin tener contexto de funciones externas.

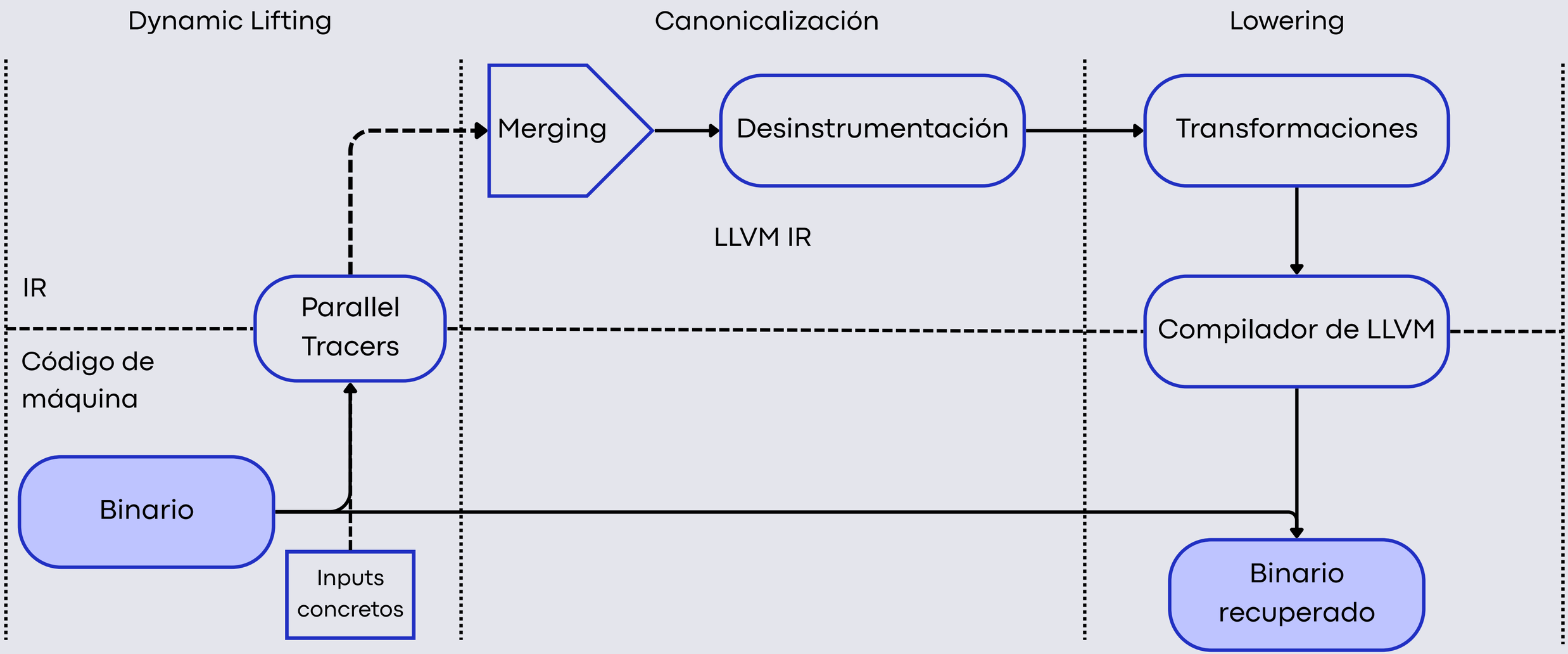
OFUSCACIÓN

Les cuesta analizar programas donde su diseño busca ofuscar la implementación

BINREC

- Binary Lifting a Intermediate Representation (IR)
- Dinámico
- Transformaciones
- Recompilación
- Libre de heurísticas
- Busca resolver las limitaciones del método estático

RECUPERACIÓN DEL BINARIO



ALGUNAS HERRAMIENTAS ACTUALES

dogbolt.org

GHIDRA

- NSA
- Estático con emulación limitada
- HLL Pseudo-C

BINARY NINJA

- Vector 35 Inc.
- Estático y emulación ligera
- HLL Pseudo-C

IDA PRO

- Hex-Rays
- Estático y dinámico
- HLL Pseudo-C

RETDEC

- Avast
- Estático
- HLL Pseudo-C

RADARE2

- Pancake
- Estático y emulación ligera
- HLL (r2dec) Pseudo-C

ANGR

- Univ. Santa Bárbara y Univ. Arizona
- Estático y emulación con SMT

EJEMPLOS DE USO

It’s demo time

01 Análisis de malware (WannaCry 2017).

02 Análisis post-mortem de exploits o backdoors en sistemas comprometidos.

03 Investigación de vulnerabilidades, auditoría y pestenting en librerías, firmware y sistemas.

04 Parcheo o compatibilidad de BIOS, controladores y videojuegos clásicos (emuladores).

¡Gracias!