Demonstrating Shiva v0.13 Alpha

By Ryan O’Neill

State of the art ELF patching solution

- A practical approach to patching native software in Linux (AArch64 support)
  - Natural patch development in C code
  - ELF symbol and relocation driven patching operations
- An ELF runtime transformation technology
  - Loads and links relocatable ELF patches at runtime
  - Introduces new linking concepts for program transformation

Shiva design goals for phase-2

- Flexible and modular micropatching system
  - Specialized ELF interpreter “/lib/shiva”
  - Program transformation at runtime with relocatable objects and ELF ABI extensions (i.e. transformations, linker chaining, etc.)
- Versatile patching capabilities
  - Relink any part of a programs code or data symbolically
  - Splice operations: Insert symbolically rich C code anywhere
  - Transformation operations with C macros
- The ability to install many patches successively

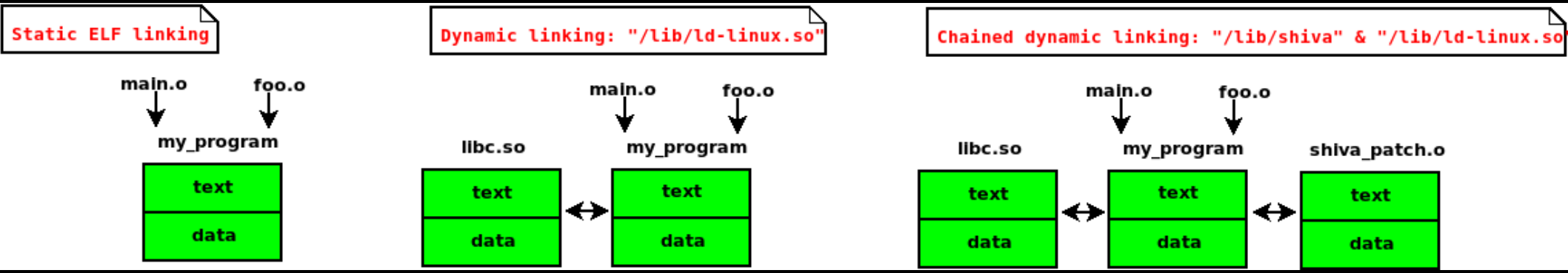
ELF ABI compatibility

- Shiva fits into the ELF ABI toolchain
  - Compiler: gcc/clang
  - ELF Linker: “/bin/ld”
  - ELF Dynamic linker: “/lib64/ld-linux.so”
  - Shiva (Custom dynamic linker): “/lib/shiva”

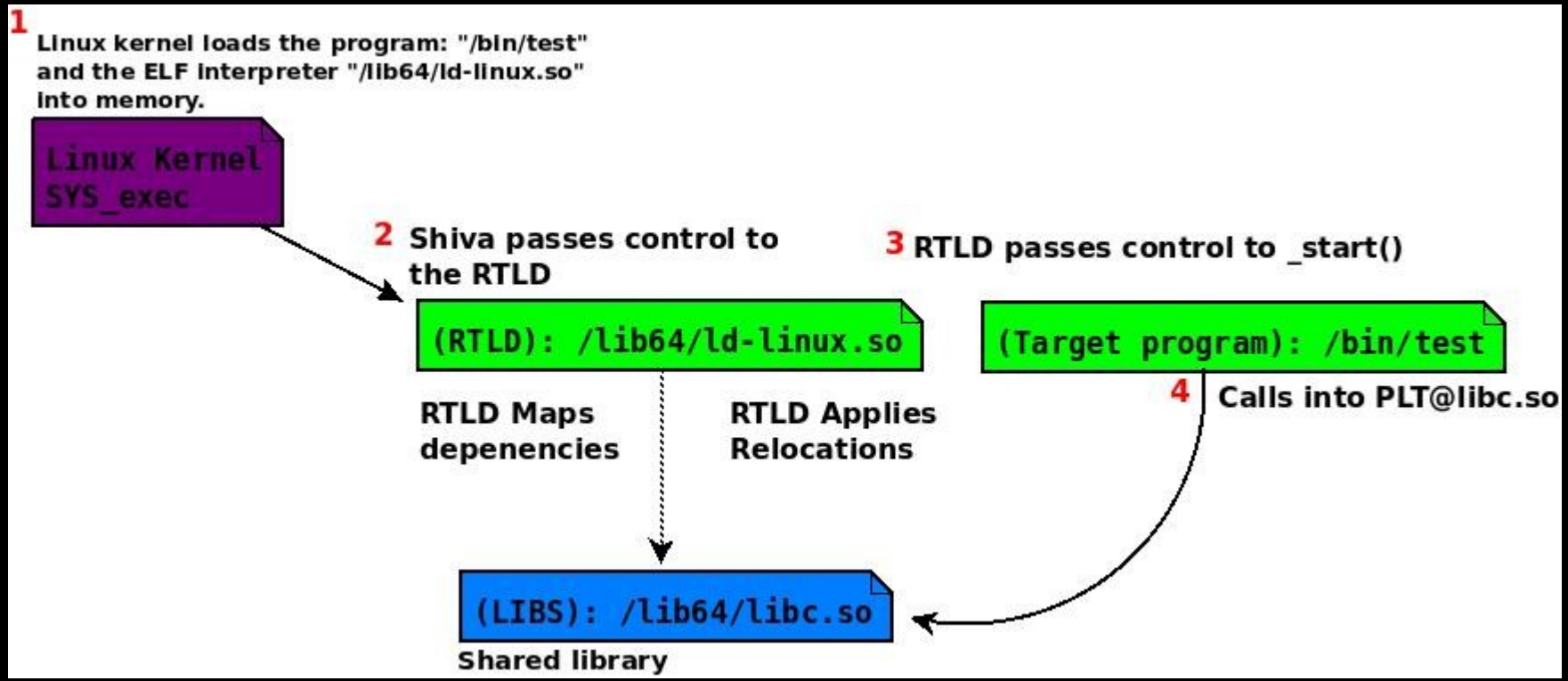
ELF ABI extensions

- Shiva extrapolates on existing ELF ABI
  - ELF Transforms
  - Chained dynamic linkers (aka ELF interpreter chaining)
  - Cross ELF Relocations
  - Delayed ELF relocations
  - LDSO ELF meta-data programming
  - Shiva patch prelinking
  - On-the-fly generation of code relocations

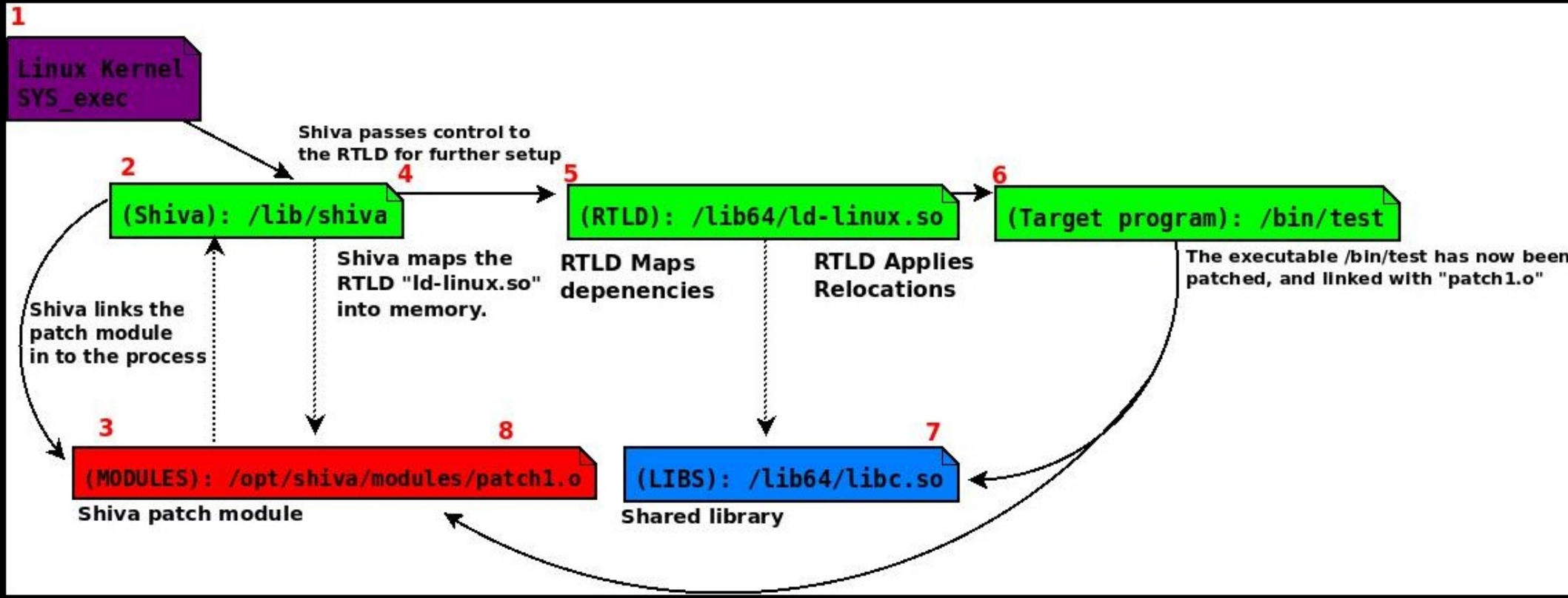
Shiva ELF linking workflow



Standard ELF dynamic linking workflow



ELF “Linker chaining” diagram



NASA Requirements

- Support for Linux AArch64 ELF
  - cFS (Control Flight Software): ELF PIE binary
  - Ported Shiva from Linux x86_64 to AArch64 and greatly enhanced it through AMP phase-2
- Small modular ELF patches
- X86_64 support is on the way in current phase-3

Versatile patching capabilities

- Aims to accommodate many types of patching challenges
- Patches are written in C code
  - Symbolic interposition for natural expression of re-writing code and data
  - Linking of live variables: registers and stack variables via ***Shiva Transform Macros.***
  - ***Function splicing***: Splice relocatable C code into any function with rich symbolic expression

Phase-3 enhancements. P1

- Shiva PreLinker v2.0 finished
  - Generates and stores control flow data in .shiva.branch and .shiva.xref sections
  - Previously Shiva had to generate control flow data at runtime
  - 3000% increase in performance on ELF programs with large .texts.

Phase-3 enhancements. P2

- Enhancements to Shiva's ***ELF Transformations***
  - Function splicing has more nuanced usability now
    - Can now inject rich symbolic code in between two contiguous addresses

Function splicing example: Fixing a strcpy vulnerability

```
#define MAX_BUF_LEN 16

void copy_string(char *src)
{
        char buf[MAX_BUF_LEN];

    - strcpy(buf, src);
}
```

```
#define MAX_BUF_LEN 16

void copy_string(char *src)
{
        char buf[MAX_BUF_LEN];

    + strncpy(buf, src, MAX_BUF_LEN);
    + buf[MAX_BUF_LEN - 1] = '\0';
}
```

```
00000000000007b4 <copy_string>:
 7b4:   stp     x29, x30, [sp, #-48]!
 7b8:   mov     x29, sp
-7bc:   str     x0, [x29, #24]
-7c0:   add     x0, x29, #0x20
-7c4:   ldr     x1, [x29, #24]
-7c8:   bl      690 <strcpy@plt>
 7cc:   nop
 7d0:   ldp     x29, x30, [sp], #48
 7d4:   ret
```

```
00000000000007b4 <copy_string>:
 7b4:   stp     x29, x30, [sp, #-48]!
 7b8:   mov     x29, sp
+7bc:   mov     x9, x29
+7c0:   add     x9, x9, #0x20
+7c4:   mov     x1,x9
+7c8:   mov     x3, x1
+7cc:   mov     x2, #0xf
+7d0:   mov     x1, x0
+7d4:   mov     x0, x3
+7d8:   bl      8d4 <strncpy@patch.plt>
 7dc:   nop
 7e0:   ldp     x29, x30, [sp], #48
 7e4:   ret
```

Phase-3 enhancements. P3

- Port AMP version of Shiva to x86_64
  - For NASA’s most relevant challenge problems
  - Coming soon….

Stay tuned for a 10 minute demo

- 1. Function splicing to fix a strcpy vulnerability
- 2. Splice new C code into any part of the binary

For more detailed information about Shiva

- Contact:
  - elfmaster@arcana-research.io
- Official Shiva site:
  - https://arcana-research.io/shiva
- Shiva github
  - https://github.com/advanced-microcode-patching/shiva
- Shiva user manual
  - https://github.com/advanced-microcode-patching/shiva_user_manual