



Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Lecture  
consigliate

# Software Reverse Engineering

Quanto mai sarà difficile andare da Binario a Sorgente?

Gianfranco Gallizia

Linux Day 2022



# Introduzione

## Disclaimer

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Lecture  
consigliate

Non sono un avvocato, né un giurista. Tutto quello che troverete in questa presentazione sono miei appunti sugli strumenti e sulle tecniche utilizzate per l'analisi di file binari che ho raccolto e sviluppato nel corso degli anni. Questa presentazione ha un puro scopo didattico ed illustrativo.

Non eseguite le tecniche descritte per fini illeciti ed in caso di dubbi consultate un professionista.



# Introduzione

Che cos'è il Software Reverse Engineering (SRE)?

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

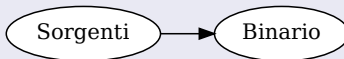
Strumenti

Analisi statica

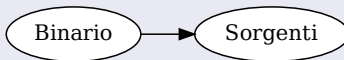
Analisi dinamica

Lecture  
consigliate

## Flusso normale



## Flusso inverso





# Introduzione

## Analisi statica e dinamica

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Letture  
consigliate

### Analisi statica

Effettuata senza eseguire il software, guardando unicamente i bit su disco.

### Analisi dinamica

Effettuata quando il software è in esecuzione guardando le attività ed il flusso di esecuzione.



# Introduzione

## Analisi statica e dinamica

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Lecture

consigliate

## Analisi statica - pro e contro

- Pro:
  - Non si esegue il binario in esame. Bypassando le protezioni a runtime ed evitando di eseguire potenziali malware.
  - L'analisi può essere fatta su una macchina di architettura diversa dalla macchina bersaglio (ad esempio: analisi di un binario ARM su un pc amd64).
- Contro:
  - Non si esegue il binario in esame. Se il binario è compresso/offuscato/criptato/modificato a runtime non lo si vedrà.



# Introduzione

## Legalità del reverse engineering

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Lecture

consigliate

## Quando è legale fare SRE nell'Unione Europea?

- Quando si ha l'esplicito consenso (meglio se scritto) del detentore dei diritti su software.
- Quando si effettua al fine di garantire l'interoperabilità del software con un altro software e/o con un altro sistema operativo/un'altra architettura. A patto che:
  - Si sia in possesso di regolare licenza d'uso per il software.
  - Non siano disponibili in altro modo le informazioni necessarie a garantire l'interoperabilità.
  - L'azione di SRE sia limitata alle sole parti necessarie a garantire l'interoperabilità.



# Introduzione

## Legalità del reverse engineering

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Letture  
consigliate



↑  
Esplicito  
consenso

↑  
Analisi  
Malware





# Strumenti del SRE

Di cosa ho bisogno?

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Lecture  
consigliate

- Conoscenza
  - Manuali, presentazioni, post su forum. Raccogliete quante più informazioni possibile.
- Pazienza
- Alcuni software
  - Editor esadecimale
  - Disassembler/Decompiler
  - System trace
  - Debugger
  - Virtual Machine/Emulatori
  - (Almeno) un linguaggio di programmazione/scripting





# Analisi statica - Editor esadecimale

## Introduzione

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Lecture

consigliate

## Editor esadecimale

- Equivalente binario degli editor di testo.
- Usa la notazione esadecimale per rappresentare i byte del file (cifre da 0-9 e poi A=10, B=11, C=12, D=13, E=14, F=15). Due cifre esadecimali possono rappresentare i numeri da 00 a FF (255).
- Possono essere molto semplici (ad es. GHex) o con funzioni aggiuntive che semplificano l'analisi di strutture dati e la ricerca di pattern.
- Solitamente divisi in tre colonne: offset, dati in esadecimale, rappresentazione ASCII.



# Analisi statica - Editor esadecimale

## Introduzione

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Lecture  
consigliate

## Editor esadecimale - Utilizzi

- Indagine preliminare:
  - È davvero un eseguibile?
  - Ci sono porzioni che contengono dati leggibili?
- Binary patching (sostituisco parte del binario per cambiarne il comportamento).



# Analisi statica - Editor esadecimale

GHex

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Lecture  
consigliate

The screenshot shows the GHex application window. The main area displays hex data for 'pokersquares.bin' in columns of 16 bytes. The ASCII column shows the corresponding characters, including spaces, punctuation, and symbols. Below the main area is a conversion panel with various input fields and checkboxes.

Signed 8 bit:	-8	Signed 32 bit:	8716969	Hexadecimal:	A9
Unsigned 8 bit:	169	Unsigned 32 bit:	8716969	Octal:	251
Signed 16 bit:	681	Signed 64 bit:	181554130240996	Binary:	10101001
Unsigned 16 bit:	681	Unsigned 64 bit:	181554130240996	Stream Length:	8 - +
Float 32 bit:	1.221508e-38	Float 64 bit:	1.606265e-296		

Show little endian decoding  Show unsigned and float as hexadecimal

Offset: 0x0

<https://wiki.gnome.org/Apps/Ghex>







# Analisi statica - Disassembler

## Introduzione

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Lecture  
consigliate

## Disassembler

- Effettua l'operazione inversa di un assembler (passare da codice macchina a codice assembly).
- Strettamente dipendente dall'architettura bersaglio.
- In alcuni casi (CPU/microcontrollori legacy ad 8 bit) il disassemblato è di fatto il codice sorgente.
- Più frequentemente è il risultato finale di numerose trasformazioni effettuate dal compilatore per ottenere il codice macchina.



# Analisi statica - Disassembler

## Esempi

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Letture  
consigliate

## Esempi

- objdump (Parte delle binutils)
- Radare2 (Suite di tools per SRE)
- Ghidra (Anche decompiler)



# Analisi statica - Disassembler

Radare2

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Lecture

consigliate

- Partito come tool per recuperare file da una partizione HFS+ si è evoluto fino a diventare un framework per il SRE.
- Ha un'interfaccia testuale scriptabile ed ha alcuni tool per compiti specifici: `rasm2` assembler/disassembler standalone, `rabin2` per l'analisi degli header binari, `radiff2` per l'analisi delle differenze tra due binari.
- Supporta architetture multiple: amd64, ARM, x86, MIPS, PowerPC, Atmel AVR, Motorola 68k, MSIL, Java bytecode, Dalvik VM bytecode, WebAssembly (WASM) e altre ancora.
- È disponibile una GUI (laito).





# Analisi statica - Disassembler

Radare2

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Lecture  
consigliate

```
skyglobe@gandalf: ~/rev_eng/sample
| pz[?] [len]                print zoom view (see pz? for help)
[0x004014c0]> pC
0x004014c0  eip:                0x004014f9  mov dword [esp], str.libgcc_s
0x004014c0  mov dword [0x406070], 0 0x00401500  call dword [sym.imp.KERNEL32.
0x004014ca  jmp sym.__tmainCRTStartup 0x00401506  sub esp, 4
0x004014cf  nop                0x00401509  test eax, eax
0x004014d0  sym.__atexit:      0x0040150b  je 0x401580
0x004014d0  sub esp, 0x1c      0x0040150d  mov ebx, eax
0x004014d3  mov eax, dword [esp + 0x20] 0x0040150f  mov dword [esp], str.libgcc_s
0x004014d7  mov dword [esp], eax 0x00401516  call dword [sym.imp.KERNEL32.
0x004014da  call sym.__onexit 0x0040151c  mov edi, dword [sym.imp.KERNE
0x004014df  test eax, eax      0x00401522  sub esp, 4
0x004014e1  sete al            0x00401525  mov dword [0x406028], eax
0x004014e4  add esp, 0x1c      0x0040152a  mov dword [esp + 4], str.__re
0x004014e7  movzx eax, al      0x00401532  mov dword [esp], ebx
0x004014ea  neg eax            0x00401535  call edi
0x004014ec  ret                0x00401537  sub esp, 8
0x004014ed  nop                0x0040153a  mov esi, eax
0x004014ee  nop                0x0040153c  mov dword [esp + 4], str.__de
0x004014ef  nop                0x00401544  mov dword [esp], ebx
0x004014f0  sym.__gcc_register_frame: 0x00401547  call edi
0x004014f0  push ebp           0x00401549  mov dword [0x403004], eax
0x004014f1  mov ebp, esp      0x0040154e  sub esp, 8
[0x004014c0]> |
```

<https://www.radare.org>



# Analisi statica - Decompiler

## Introduzione

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Lecture

consigliate

## Decompiler

- Effettua l'operazione inversa del compilatore.
- Molto più complesso di un disassembler (vi è più perdita di informazione da sorgente ad assembly che da assembly a codice macchina).
- Richiede molto più lavoro manuale da parte dell'operatore (leggasi: serve avere esperienza di programmazione).
- Solitamente il linguaggio di programmazione "ricostruito" dal binario è il C (con risultati "interessanti" nel caso in cui non sia così).



# Analisi statica - Decompiler

## Ghidra

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Lecture

consigliate

## Ghidra

- Progetto della NSA rilasciato come open source nel 2019.
- Scritto in Java e C++ (per le parti più computazionalmente intense).
- Può essere esteso con plug-in scritti in Java o in Python (grazie all'interprete Jython).
- Svolge sia funzioni di disassembler che di decompilatore (C/C++).
- Può interfacciarsi con un debugger (GDB in Linux, WinDbg in Windows).





# Analisi dinamica - System trace

## Introduzione

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Lecture

consigliate

## System trace

- La system trace (o, più precisamente, system call trace) è una tecnica di analisi dinamica che intercetta le chiamate alle API del sistema operativo effettuate da un programma.
- Non c'è bisogno di modificare il programma analizzato, basta lanciarlo tramite un wrapper che instruirà il sistema operativo a fare rapporto al wrapper.
- Il programma tracciato può cercare di individuare il tracciante e modificare il suo comportamento di conseguenza.



# Analisi dinamica - System trace

## Esempi

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Letture  
consigliate

## Esempi

- strace (Linux)
- ktrace (FreeBSD)
- Process Monitor (Windows - Sysinternals)



# Analisi dinamica - Debugger

## Introduzione

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Lecture  
consigliate

## Debugger

- Strumento usato dagli sviluppatori quando hanno bisogno di eseguire passo-passo il loro lavoro alla ricerca di problemi.
- Non è necessario avere i sorgenti di un programma per agganciare un debugger ad un processo: si riceverà un warning e l'analisi proseguirà a livello di disassemblato.



# Analisi dinamica - Debugger

## Esempi

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

**Analisi dinamica**

Lettere

consigliate

## Esempi

- gdb
- x64dbg (Windows)





# Analisi dinamica - Debugger

x64dbg

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Lecture  
consigliate

x64dbg.exe - PID: 22888 - Module: x64dbg.exe - Thread: Main Thread 10132 - x64dbg

File View Debug Tracing Plugins Favourites Options Help Jan 5 2022 (TitanEngine)

CPU Log Notes Breakpoints Memory Map Call Stack CPU Script Symbols References Handles Trace

RIP	RAX	ROV	EBP	Code	Comment	Registers	Flags
00007FF751FA2440	48:8BEC 28			CALL x64dbg.7FF751FA2950	EntryPoint	RAX 00007FF751FA2440 <x64dbg.EntryPoint>	
00007FF751FA2444	E8 07050000			JMP x64dbg.7FF751FA2950		RAX 0000000000000000	
00007FF751FA2448	48:8BEC 28			CALL x64dbg.SUB_7FF751FA2454		RAX 0000000000000000	
00007FF751FA244C	E9 02000000			CALL CC		RAX 0000000000000000	
00007FF751FA2450	CC			INT3		RAX 0000000000000000	
00007FF751FA2454	48:8BC4			MOV RAX, RSP	sub_7FF751FA2454	RAX 0000000000000000	
00007FF751FA2458	48:8B58 08			MOV QWORD PTR [RAX+0x8], RBX		RAX 0000000000000000	
00007FF751FA245C	48:8B70 10			MOV QWORD PTR [RAX+0x10], RSI		RAX 0000000000000000	
00007FF751FA2460	57			PUSH RDI		RAX 0000000000000000	
00007FF751FA2464	48:8B3C 30			SUB RSP, 0x30		RAX 0000000000000000	
00007FF751FA2468	48:8B60 F0			AND QWORD PTR [RAX-0x10], 0x0		RAX 0000000000000000	
00007FF751FA246C	48:8B50 00			AND QWORD PTR [RAX-0x10], 0x0		RAX 0000000000000000	
00007FF751FA2470	FF15 1D0D0000			CALL QWORD PTR [RAX-0x10], 0x0		RAX 0000000000000000	
00007FF751FA2474	0F87 F0			MOVZX ESI, EAX		RAX 0000000000000000	
00007FF751FA2478	65:48:8B0C25 30000000			MOV RCX, QWORD PTR [EAX+0x30]		RAX 0000000000000000	
00007FF751FA247C	48:8B51 08			MOV RDX, QWORD PTR [RCX+0x8]		RAX 0000000000000000	
00007FF751FA2480	330B			XOR EBX, EBX		RAX 0000000000000000	
00007FF751FA2484	33C0			XOR ECX, ECX		RAX 0000000000000000	
00007FF751FA2488	F048:0FB115 98320000			LOCK CMPQCBQ QWORD PTR [0x7FF751FA5728], RDX		RAX 0000000000000000	
00007FF751FA248C	74 0E			JNE x64dbg.7FF751FA24A0		RAX 0000000000000000	
00007FF751FA2490	48:3BC2			CMQ RAX, RDX		RAX 0000000000000000	
00007FF751FA2494	75 07			JNE x64dbg.7FF751FA249E		RAX 0000000000000000	
00007FF751FA2498	8B 01000000			MOV EDX, 0x1		RAX 0000000000000000	
00007FF751FA249C	EB 02			JMP x64dbg.7FF751FA24A0		RAX 0000000000000000	
00007FF751FA24A0	EB E5			JMP x64dbg.7FF751FA2485		RAX 0000000000000000	
00007FF751FA24A4	8B05 8A320000			MOV EAX, QWORD PTR [0x7FF751FA5730]		RAX 0000000000000000	
00007FF751FA24A8	82F8 01			CMQ EAX, EAX		RAX 0000000000000000	
00007FF751FA24AC	75 0A			JNE x64dbg.7FF751FA2485		RAX 0000000000000000	
00007FF751FA24B0	8048 1E			LEA ECX, QWORD PTR [RAX-0x1E]		RAX 0000000000000000	
00007FF751FA24B4	EB C2000000			CALL SHLQ <_SHLQ_0x1E>		RAX 0000000000000000	

Registers: RAX 00007FF751FA2440 <x64dbg.EntryPoint>, RBX 0000000000000000, RCX 0000000000000000 <x64dbg.EntryPoint>, RDX 00007FF751FA2440 <x64dbg.EntryPoint>, RBP 0000000000000000, RSP 0000000000000000, RSI 0000000000000000, RDI 0000000000000000, R10 0000000000000000, R11 0000000000000000, R12 0000000000000000, R13 0000000000000000, R14 0000000000000000, R15 0000000000000000

Flags: RFLGS 0000000000000024, ZF 1, OF 1, SF 0, OF 0, SE 0, DF 0

Default (x64 fastcall) | Unlocked

1: RCX 0000000000000000  
2: RDX 00007FF751FA2440 <x64dbg.EntryPoint>

fs:0000000000000000 fsp:0000000000000000  
28 ('')

.text:00007FF751FA2440 x64dbg.exe:02440 :1840 <EntryPoint>

Address	Hex	ASCII
00007FF751FA3200	00 00 00 00 00 92 02 61 00 00 00 00 02 00 00 00	....Y.da.....
00007FF751FA3204	3A 00 00 00 00 37 00 00 00 00 00 00 00 00 00 00	...07.D).....
00007FF751FA3208	00 92 61 00 00 00 00 00 00 00 00 00 14 00 00 00	.....l.....
00007FF751FA320C	0C 38 00 00 00 0C 2A 00 00 65 62 67 6E 65 6C 33 32	.8.....kernel32
00007FF751FA3210	2E 64 6C 6C 00 00 00 00 00 64 62 67 6E 65 6C 70 2E	ttl.....dbyhel.p
00007FF751FA3214	64 6C 6C 00 00 00 00 00 00 65 6E 68 68 7E 68 7A	.....NtUser!NtUser
00007FF751FA3218	82 72 69 74 8E 88 7E 70 70 00 00 00 00 00 00 00	.....NtUser!NtUser
00007FF751FA321C	63 74 50 72 6F 63 65 74 50 72 6F 63 65 73 65 73	.....NtUser!NtUser
00007FF751FA3220	65 78 63 65 75 74 6F 6F 6E 68 6F 6C 6C 6C 6C 6C	.....NtUser!NtUser
00007FF751FA3224	47 65 74 50 72 6F 63 65 74 50 72 6F 63 65 73 65	y.....GetProcess
00007FF751FA3228	73 55 73 65 72 40 6F 64 65 78 63 65 70 74 74	.....IssueHostAccept
00007FF751FA322C	69 6F 6E 50 6F 6C 69 63 79 00 00 00 45 72 72 6F	.....TmPolic..._Erfro
00007FF751FA3230	72 00 00 00 00 00 00 00 55 6E 61 62 6C 65 20 74	.....Unable t

Command: x64dbg.exe: 00007FF751FA3328 -> 00007FF751FA3328 (0x0000011 bytes)

Paused

Time Wasted Debugging: 9:01:24:07

<https://x64dbg.com>



# Analisi dinamica - Virtual Machine ed Emulatori

## Introduzione

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Lecture

consigliate

## Virtual Machine ed Emulatori

- Vi sono casi in cui non è possibile usare la stessa macchina per analizzare un programma ed eseguirlo:
  - Il programma è scritto per un'architettura diversa da quella della macchina utilizzata per l'analisi.
  - Il programma richiede hardware non presente sulla macchina utilizzata per l'analisi.
  - Non ci si fida del programma che si sta analizzando.
  - Si vuole analizzare il programma in un contesto il più pulito possibile.



# Analisi dinamica - Virtual Machine ed Emulatori

Introduzione - cont.

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Lecture  
consigliate

## Virtual Machine ed Emulatori

- Le macchine virtuali (Virtual Machine, o VM) permettono di eseguire un programma in un contesto diverso da quello del sistema operativo della macchina ospitante:
  - Diverso kernel.
  - Diversa distribuzione.
  - Diverso set di pacchetti installati.
  - Diverso sistema operativo.



# Analisi dinamica - Virtual Machine ed Emulatori

Introduzione - cont.

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Letture

consigliate

## Virtual Machine ed Emulatori

- Quando i binari da analizzare non possono essere eseguiti sulla macchina corrente si può fare ricorso agli emulatori.
- Gli emulatori simulano hardware differente da quello presente sulla macchina che li sta eseguendo.



# Analisi dinamica - Virtual Machine ed Emulatori

## QEMU

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Lecture

consigliate

## QEMU

- QEMU è un emulatore e virtualizzatore open source.
- Può operare in autonomia oppure appoggiarsi ad un Hypervisor (solitamente KVM o Xen).
- Può operare in due modalità:
  - System mode emulation, in cui emula un'intera macchina.
  - User mode emulation, in cui emula solo la CPU e consente di eseguire un processo compilato per un'architettura su di un'altra.
- Supporta il protocollo di comunicazione remota di gdb e consente quindi di fare debugging di binari compilati per architetture differenti da quella della macchina su cui gira.



# Analisi dinamica - Virtual Machine ed Emulatori

## QEMU

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

**Analisi dinamica**

Lecture

consigliate



<https://www.qemu.org/>



# Lecture consigliate

## Linguaggi di programmazione

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Lecture  
consigliate

### Linguaggi di programmazione

- Brian Kernighan, Dennis M. Ritchie "The C Programming Language" ISBN 0-13-110362-8 272 pagine
- Steve Oualline "Practical C++ Programming" ISBN 978-0-596-00419-4 550 pagine
- Randall Hyde "The Art of Assembly Language" ISBN 978-1-59327-207-4 732 pagine



# Lecture consigliate

## Sistemi operativi

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Lecture  
consigliate

## Sistemi operativi

- Brian Ward "How Linux Works" ISBN 978-1-71850-040-2  
464 pagine
- Michael Kerrisk "The Linux Programming Interface" ISBN  
978-1-59327-220-3 1552 pagine
- Charles Petzold "Programming Windows" ISBN  
978-1-57231-995-0 1479 pagine





# Lecture consigliate

## Miscellanea

Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Lecture  
consigliate

## Miscellanea

- Chris Eagle, Kara Nance "The Ghidra Book" ISBN 978-1-71850-102-7 608 pagine
- V. Anton Spraul "Think like a Programmer" ISBN 978-1-59327-424-5 256 pagine
- Brian Hook "Write Portable Code" ISBN 978-1-59327-056-8 272 pagine



Software  
Reverse  
Engineering

Gianfranco  
Gallizia

Introduzione

Disclaimer

Cos'è il SRE?

Analisi statica e  
dinamica

Legalità del SRE

Strumenti

Analisi statica

Analisi dinamica

Lecture  
consigliate

# Grazie per l'attenzione

Live Demo con domande?