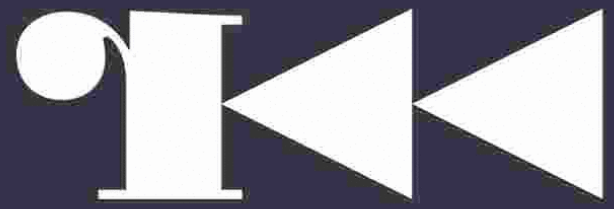
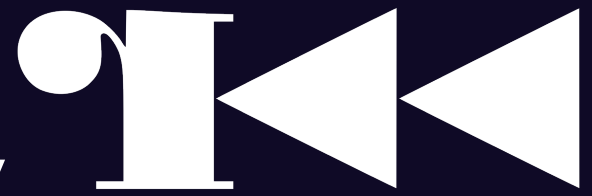


> radare2

PRIMEIROS PASSOS EM ENGENHARIA REVERSA COM A FERRAMENTA

Arquivos disponíveis no github @deniszanin

> sobre as ferramentas

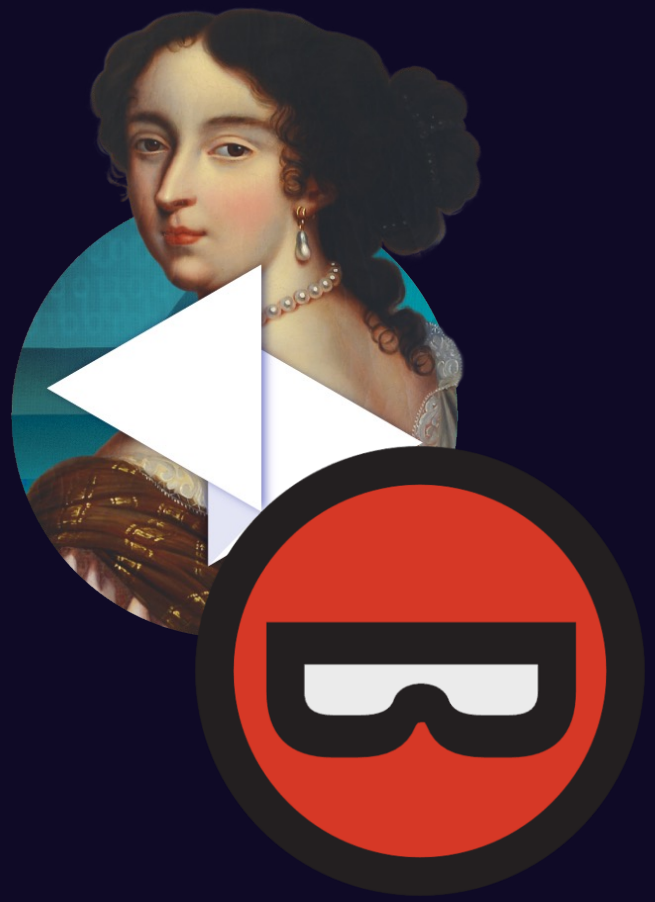


Radare2 é um conjunto de ferramentas (`rabin2`, `rasm2`, `rahash2`, `rax2`, etc.). A interface `radare2` é a principal delas. O *framework* é gratuito e de código aberto, executado pela linha de comando do sistema operacional.

Outras ferramentas existentes:

- Binary Ninja
- Ghidra
- IDA Free e IDA Pro

> outras ferramentas



IDA Free

Gratuito, código fechado e limitado.

IDA PRO

Custo inicial de US$2.000,00, versão completa (com limitações). Média de US$1.500,00 por pacote adicional.

Binary Ninja

Custo de US$299; estudantes US$74. Código fechado.

> engenharia reversa

Em linhas gerais, reverter algo pronto para seu estado original; encontrar a receita de um bolo pronto.

> instalação

A instalação pode ser feita com os binários disponíveis no repositório **radareorg/radare2**, no Github, **e/ou**, compilar o *framework* com a última versão do repositório. Para ambientes macOS é aconselhado instalar através da ferramenta brew.

EM NENHUMA HIPÓTESE UTILIZE OS *PACKAGE MANAGERS* DAS DISTRIBUIÇÕES (dnf, apt-get, etc)

_ COMPILAR LOCAL NO LINUX (SEM PRIVILÉGIOS ROOT)

```
shell> pip3 install --user keystone-engine
shell> pip3 install --user unicorn
shell> pip3 install --user capstone
shell> pip3 install --user ropper
shell> pip3 install --user meson
shell> ./sys/user.sh --install-path $HOME/.local --with-capstone5
```

> linha de comando

_ LINHA DE COMANDO INTERATIVA

```
shell> radare2 --
-- Mind the gap
[0x00000000]> comando a ser executado
```

A linha de comando é interativa, por onde o usuário executa as ações na ferramenta. No modo visual/gráfico, a linha de comando pode ser chamada com a tecla `:` (dois pontos).

> comandos iniciais

_ COMANDOS DIVERSOS

`fo`	exibir uma fortune (frase aleatória)
`cls`	limpar a tela
`eco [tema]`	exibir/ definir tema para a ferramenta

_ COMANDOS ESSENCIAIS

`?`	exibir lista com os comandos
`[comando]?`	exibir lista de ajuda para determinado comando
`q`	sair da ferramenta
`o <arquivo>`	abrir, em modo de leitura, o arquivo <arquivo>
`V`	entrar no modo visual da ferramenta
`VV`	entrar no modo gráfico da ferramenta
`s [offset]`	exibir/mover-se para determinado *[offset]* do bin. (*siga para*)

> primeiro exemplo

```
shell> r2 -w _quake3_bin_exe_
```

PRIMEIRO EXEMPLO PRÁTICO

Crackear (modificar) um arquivo binário para remover a obrigatoriedade do CD para jogar Quake 3 Arena.

_ OBJETIVOS

a. encontrar a função de verificação pelo CD.
b. remover a verificação do CD.

> ex.1 na prática

_ COMANDOS DE ANÁLISE

`aa`	realizar análise superficial do binário
`aaa`	realizar análise parcial do binário
`aaaa`	realizar análise completa do arquivo binário **(CUIDADO!)**

_ COMANDOS DE CONHECIMENTO SOBRE BINÁRIO

`ie`	exibir entrypoint do binário
`ii`	obter informações dos imports (informação de imports)
`iI`	obter informações gerais (*informação da Informação*)
`it`	obter assinaturas digitais do arquivo (*informação da assinatura*)
`iZ`	obter informação do tamanho do arquivo
`iz`	listar *strings* do binário, na seção .data (*informação stringz*)
`izz`	listar todas as *strings* do binário (*informação stringzzz*)

> ex.1 na prática

_ COMANDOS DE VISUALIZAÇÃO

`pd [número_inst] @ [endereço]`	imprimir disassembly NO(@) [endereço]
`pd-- [contexto] @ [endereço]`	imprimir disassembly NO(@) [endereço]
`pdf [função]`	imprimir disassembly da função
`axt [endereço]`	listas referências PARA [endereço]
`axf [endereço]`	listas referências DESTE [endereço]
`afn [nome]`	renomear
`afvn [novo] [antigo]`	renomear variável de [antigo] para [novo]

_ COMANDOS DE MODIFICAÇÃO

`wx [opcode] @ [endereço]`	sobrescrever [opcode] NO(@) [endereço]
`wa [instrução] @ [endereço]`	sobrescrever [instrução] NO(@) [endereço]

> modo visual

_ TECLAS DE ATALHO NO MODO VISUAL (V)

q	sair da ferramenta
?	menu de ajuda
:	acessar a linha de comando
R	alternar entre cores (temas) do radare2
p	alternar entre os modos de visualização
c	ativar/ desativar modo de cursor
i	inserir/ sobrescrever hexadecimal
A	inserir/ sobrescrever instrução *Assembly*
;	edição de comentários
e	acessar as configurações da ferramenta radare2

> segundo exemplo

```
shell> r2 -b 32 -a x86 _cdkey_dll_
```

SEGUNDO EXEMPLO PRÁTICO

Identificar o algoritmo (funções) de validação da CD-Key do *software* e criar um gerador de chaves.

_ OBJETIVOS

a. encontrar a função de verificação de CD-Key.
b. *traduzir* o funcionamento do algoritmo.

> terceiro exemplo

```
shell> r2 -A _desafioR2d_dll_
```

TERCEIRO EXEMPLO PRÁTICO

Descobrir o nome do *software* do último exemplo.

_ DESAFIO BÔNUS

a. EXTRAIR a informação com apenas dois comando no radare2.

@deniszanin

twitter, github, keybase, etc.