Engenharia Reversa

Introdução prática

Quem sou eu

- Meu nome é Leonardo Ventura
- 5º período de Ciência da Computação
- Membro do GRIS desde 2018.1
- Foco principal em Engenharia Reversa, mas gosto de falar besteira sobre muito mais coisa.

Introdução

- O que é engenharia reversa?
- Quais são seus objetivos?
- Como ela é aplicada?
- Em quais casos a engenharia reversa foi importante?

O que é engenharia reversa?

- Basicamente, engenharia reversa é o processo de análise para entender o funcionamento de um objeto alvo.
- Engenharia reversa não é aplicada somente em software.
- Resumidamente, engenharia reversa é descobrir como um programa funciona.

Quais são seus objetivos?

- Crackear programas pagos.
- Copiar tecnologias de adversários comerciais.
- Análise de segurança.
- Desenvolvimento de malware.
- Espionagem.
- Aprendizagem (espero que a gente se encaixe aqui).
- etc.

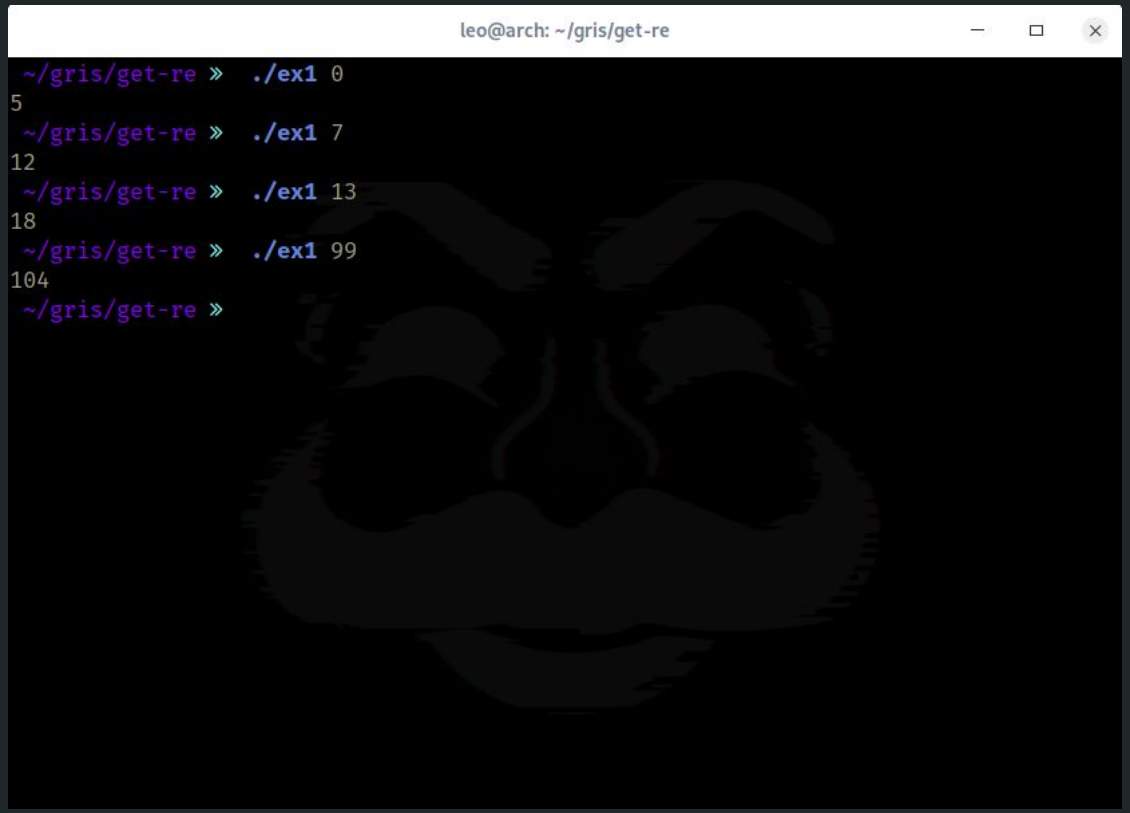
Como ela é aplicada?

- Desmontagem e remontagem.
- Testes e “chutes”.
- Em software:
  - Análise estática.
  - Análise dinâmica.
  - Análise de código fonte.

Em quais casos a engenharia reversa foi importante?

- Primeira Guerra Mundial - Roland Garros e Anthony Fokker.
- AMD - Intel.
- Spectre.
- Xbox 360.
- IPod.
- WannaCry.

Vamos praticar!



Vamos praticar!

Tentem escrever um código em C (preferencialmente) que gere a mesma saída.

Resposta

```c
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char** argv) {
    int x = atoi(argv[1]);
    printf("%d\n", x+5);

    return 0;
}
```

Continuação

- Uma das formas de engenharia reversa mais conhecidas e, provavelmente, uma das mais complexas, é a feita em cima do executável do programa.
- Para isso, utilizamos algumas ferramentas para nos ajudar - ficar entendendo 100010101010101 é maluquice.
- As ferramentas mais comuns são debuggers e descompiladores - o último é muito útil para linguagens interpretadas em Máquinas Virtuais como Java, C#.
- Vamos usar o GNU Debugger (GDB para os mais íntimos) para entender um executável sem ter acesso ao seu código fonte.
- Vale lembrar que essa é, provavelmente, a forma de engenharia reversa mais cobrada em CTFs.

Usando o GDB

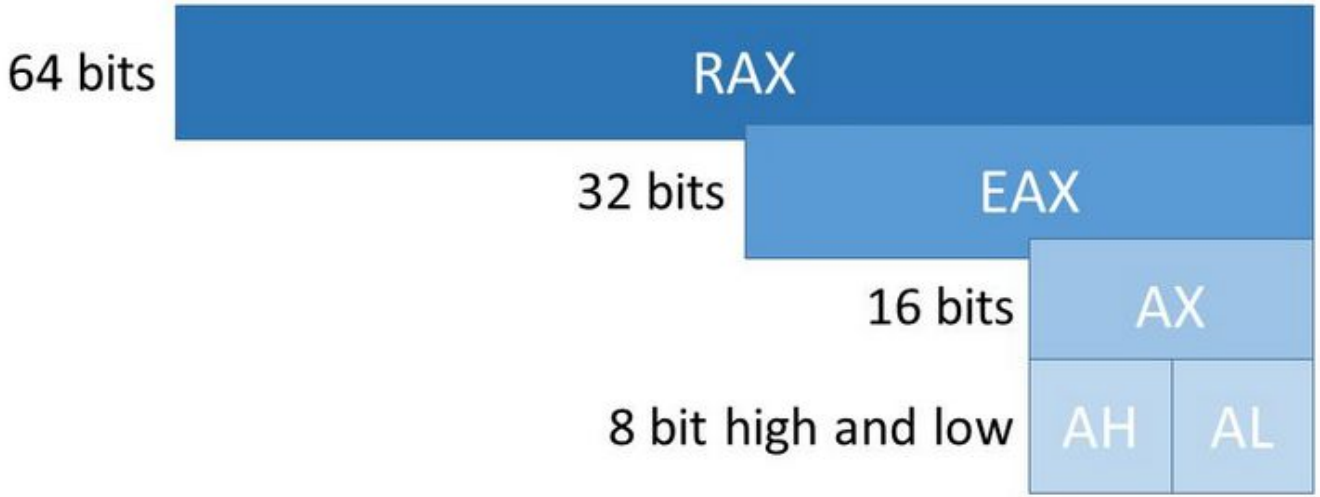
- Alguns comandos básicos estão disponíveis em: https://github.com/leo-ventura/get-re.
- Não instalem o PEDA ainda!
- Primeiro devemos entender o básico para depois utilizarmos ferramentas que facilitam nosso trabalho.

Entendendo um pouco do Assembly

- Registradores: em uma máquina Linux de 64 bits temos acesso a vários registradores (a que eu acredito que vocês estão utilizando). Entre eles temos: rax, rbx, rcx, rdx e por aí vai.
- rax, eax, ax, ah, al são parte do mesmo registrador, sendo rax para 64 bits, eax 32 bits, ax 16 bits, ah parte alta do ax, al parte baixa do ax.
- Diagraminha bonitinho no próximo slide explica melhor

Register Layout

- x86 has gone through 16, 32, and 64 bit versions
- Registers can be addressed in whole or in part



Entendendo um pouco do Assembly

- mov $10, %rax => move 10 para o registrador rax (podem imaginar o registrador como uma variável).
- test %rax, %rax => testa se %rax é 0 por meio de um &.
- cmp $50, %rax => compara %rax com 50 por meio de uma subtração, vem geralmente seguido de uma das seguintes instruções:
  - je => jump if equal: desvia se %rax possuía o valor 50.
  - jl => jump if less: desvia se %rax possuía valor menor que 50.
  - jg => jump if greater: desvia se %rax possuía valor maior que 50.
  - jle => jump if less or equal: desvia se %rax possuía valor menor ou igual a 50.

Vejam https://stackoverflow.com/questions/9617877/assembly-jg-jnle-jl-jnge-after-cmp.

Vamos praticar!

- Utilizem o executável disponível em: https://github.com/leo-ventura/get-re/blob/master/desafio
- Tentem achar a flag (vocês saberão qual é quando vê-la - pelo menos eu espero).
- Se virem alguma instrução que não sabem ao certo o que faz e não conseguir com o desafio, pesquisem ou me perguntem.
- obs: Eu resolvi o problema de uma forma, mas existem várias, qualquer uma que vocês conseguirem será válida.
- Vocês terão alguns minutos para resolver, depois resolverei aqui no projetor.
- Ah, uma dica: tem uma tabela bem conhecida que vocês vão precisar usar.

Próximos passos

- Pra quem se interessou por engenharia reversa e quer aprender mais, tem um material muito bom disponível no Youtube (em português!): https://www.youtube.com/playlist?list=PLIfZMtpPYFP6zLKInyAeWY1I85VpyshAA
- Podem falar comigo pelo Telegram caso tenham alguma dúvida.

Obrigado!