Fundamentos de Segurança Informática

CTF para a semana #5 (com início a 15/11)

Matéria relacionada: usurpação de controlo por buffer overflow

Objetivo: Explorar o funcionamento da stack e buffer-overflows

Contexto

Nesta semana são propostos dois desafios com níveis de dificuldade diferentes, sendo ambos mais simples do que o problema abordado no tutorial de buffer overflows do SEED Labs da semana #5.

Os desafios terão como objetivo a exploração de vulnerabilidades de buffer overflow. Para facilitar a exploração destas vulnerabilidades, recomenda-se a utilização da VM do seed labs. **AVISO: criar um snapshot antes de instalar qualquer utilitário ou alterar configurações, para ser possível repor o estado da máquina para os restantes tutoriais e CTFs, se necessário.**

Recomenda-se também a instalação da biblioteca de python `pwntools`, que facilita significativamente a interação com executáveis e serviços remotos a partir de um script python.

Instalação da biblioteca `pwntools`.

```
echo 'export PATH="$HOME/.local/bin:$PATH"' >> .bashrc
python3 -m pip install --upgrade pwntools
```

Em ambos os desafios, programa que é fornecido pela plataforma CTF é exatamente igual ao que está a correr na porta do servidor indicada. O objetivo é começar por explorar a vulnerabilidade localmente e, numa segunda fase, interagir com o servidor e ativar a vulnerabilidade remotamente para obter a flag.

Semana 5 - Desafio 1

A plataforma CTF fornece um ficheiro ZIP com os seguintes ficheiros: um executável (`program`), o código fonte (`main.c`) e um script em python (`exploit-example.py`). Além disso, é indicado que o serviço se encontra à escuta na porta `4003` do host `ctf-fsi.fe.up.pt`. Para contactar este serviço pode utilizar-se o programa `netcat` da shell seguinte forma `nc ctf-fsi.fe.up.pt 4003`, ou executar o script em python que é fornecido (`$ python3 exploit-example.py`) de forma a interagir com o serviço.

A flag encontra-se num ficheiro chamado `flag.txt`, no working directory. O objetivo é de alguma forma ler este ficheiro tomando controlo das funcionalidades do programa que se encontra a correr.

- Para qualquer desafio desta categoria, deves começar por analisar com que proteções é que o programa foi compilado. Para fazer isso deves correr o seguinte comando `$ checksec program` (pode ser necessário instalar o pacote `checksec` com `sudo apt update`, seguido de `sudo apt install checksec`). O output deverá ser equivalente ao seguinte:

```
$ checksec program
  Arch:     i386-32-little
  RELRO:    No RELRO
  Stack:    No canary found
  NX:       NX disabled
  PIE:      No PIE (0x8048000)
  RWX:      Has RWX segments
```

Ao analisar podemo-nos aperceber que a arquitetura do ficheiro é x86 (Arch), não existe um cannary a proteger o return address (Stack), a stack tem permisssão de execução (NX), e as posições do binário não estão randomizadas (PIE), por fim existem regiões de memória com permissões de leitura, escrita e execução (RWX), neste caso referindo-se à stack.

O RELRO é uma mitigação adicional contra ataques ROP que torna read-only os endereços de algumas funções definidos no arranque do executável; não é relevante para este contexto; mais informação aqui.

- De seguida, deves analisar o código source do desafio e tentar responder às seguintes questões:
  - Existe algum ficheiro que é aberto e lido pelo programa?
  - Existe alguma forma de controlar o ficheiro que é aberto?
  - Existe algum buffer-overflow? Se sim, o que é que podes fazer?
- Depois de teres respondido a estas questões, deves ter percebido que consegues trocar o nome do ficheiro que será lido. A ideia é, assim, por em prática o ataque e substituir `mem.txt` por `flag.txt`. A flag que obtiveres deverá ser submetida no desafio *"Semana 5 - Desafio 1"*. Poderás testar localmente o programa primeiro, ou tentar diretamente com o servidor. Deverás usar o `exploit-example.py` como ajuda na exploração da vulnerabilidade. Isto dará jeito em desafios mais complicados.

Enunciado CTF

Num concurso CTF normal, os concorrentes não receberiam a informação passo a passo acima. Um enunciado possível para este problema seria um teaser com algumas pistas, como o seguinte.

Show me what you got!! A flag encontra-se no working directory, no ficheiro flag.txt

Semana 5 - Desafio 2

Novamente, é fornecido um ficheiro ZIP com um executável (`program`) e o código fonte (`main.c`). Podes e deves adaptar o script *python* que te foi dado anteriormente para interagir com este novo desafio. O Desafio 2 é uma versão ligeiramente mais elaborada do Desafio 1 e está disponível na porta 4000 (`nc ctf-fsi.fe.up.pt 4000`).

Tarefas

- Como anteriormente deves começar por correr o `checksec`. Deves reparar que não houve alterações nas permissões do executável.
- Este desafio foi adaptado do anterior. Deves analisar o código fonte de forma a entender as alterações que foram feitas e responder às seguinte questões?
  - Que alterações foram feitas?
  - Mitigam na totalidade o problema?
  - É possivel ultrapassar a mitigação usando uma técnica similar à que foi utilizada anteriormente?
- Deves explorar a vulnerabilidade, a flag encontra-se novamente no ficheiro `flag.txt` no working directory. A flag deve ser submetida no desafio *"Semana 5 - Desafio 2"*.

Enunciado CTF

Um possível enunciado num CTF real para este desafio seria o seguinte.

O nosso serviço tinha uma vulnerabilidade, mas com as nossas mitigações, acho que os atacantes já não a conseguem explorar!!! A flag encontra-se no working directory, no ficheiro flag.txt

Notas para o futuro

Repara que, no caso de desafios de pwn (vulnerabilidades envolvendo binários), os enunciados são geralmente muito mais vagos, já que apenas um programa compilado é entregue ao jogador.

Normalmente não é dado o código fonte e é suposto que ele retire toda a informação que precisa do binário, descompilando e analisando *assembly*.

O script fornecido no Desafio 1 será útil em desafios de pwn que lançaremos em semanas seguintes, pelo que é importante compreendê-lo bem e saber utilizá-lo e adaptá-lo.

Última alteração: sexta, 12 de novembro de 2021 às 21:33

◄ Tarefas para a semana #5 (com início em 15/11)

Ir para...

Tarefas para a semana #6 (com início a 22/11) ►

Portal de e-learning
Ajuda Moodle
Inovação Pedagógica

Nome de utilizador: Marcelo Henriques Couto (Sair)
FEUP-L.EIC021-2021/2022-1S

Português (pt)
Deutsch (de)
English (en)
Français (fr)
Português (pt)

Obter a Aplicação móvel