

# Gerando traços com o Sinuca PinPoints

Existem duas principais formas de alimentar o OrCS, executando aplicações completas com o direct tracer e através de traços.

Com o direct tracer, enviamos cada instrução executada pela aplicação para ser simulada no OrCS, sem a geração de arquivos intermediários. Porém, esse tipo de simulação está sujeita a variações em cada execução, devido a diversos fatores, como, por exemplo, o mecanismo *Address space layout randomization*.

A segunda forma de alimentar o OrCS é usando traços, ou seja, arquivos contendo uma cópia do que foi executado durante uma execução normal da aplicação. Como esses arquivos armazenam apenas cópias do que foi executado, eliminam quaisquer variabilidades durante a simulação, além de serem simulados mais rapidamente pelo OrCS. Sua desvantagem é o espaço que ocupam ao serem armazenados, que pode variar de alguns megabytes a centenas de gigabytes.

Para minimizar esse problema presente nos traços e acelerar o tempo de simulação, foi desenvolvida a tecnologia [SimPoints](#), que é capaz de selecionar as partes mais significativas de uma aplicação e, a partir delas, gerar traços menores, com simulação mais rápida e, ainda assim, representando com grande fidelidade o comportamento da aplicação.

A intel implementou essa tecnologia em seu programa [PinPoints](#), que foi adaptado para gerar traços para o OrCS através do Sinuca PinPoints.

Esse manual apresenta os passos necessários para a criação de traços menores, contendo as partes mais representativas de uma aplicação, utilizando o Sinuca PinPoints presente na atual versão do OrCS.

Esse manual considera que o OrCS e o Pin + PinPlay já foram instalados e compilados, de acordo com o contido no manual **Readme OrCS tracer.pdf**.

Lembre-se de compilar o Sinuca tracer de acordo com a seção **Compilando o Sinuca Tracer** daquele manual, caso contrário, após a metade do processo de geração dos traços com o Sinuca pinpoints, você obterá um erro dizendo que um arquivo **.so** não pôde ser encontrado.

**Aviso:** Os traços com o Sinuca tracer só podem ser criados a partir de programas compilados estáticamente, ou seja, com a flag **-static**, então **não** teste no seu **/bin/ls** padrão!!!! (fonte: Zanata)

## Configuração básica

O primeiro passo da configuração é informar ao Sinuca PinPoints qual a localização da pasta “OrCS/trace\_generator/extras/pinplay/PinPoints/scripts” no seu sistema.

Para isso, dentro do arquivo **config.py** presente na pasta

**OrCS/trace\_generator/extras/pinplay/PinPoints/scripts**, altere a variável **all\_scripts\_path** (linha 115) para seu caminho completo, no meu caso, fica assim:

```
109 # Global variables
110 #
111 verbose = False
112 debug   = False
113
114 # Scripts location from trace_generator folder
115 | all_scripts_path = '/home/war/OrCS/trace_generator/extras/pinplay/PinPoints/scripts/'
116
```

Provavelmente a única alteração necessária será no seu nome de usuário, mas caso não saiba o caminho, digite o comando **pwd** dentro da pasta do **config.py**.

## Arquivo de configuração

Agora que nosso Sinuca tracer está pronto, precisamos de um arquivo de configuração para definir o que queremos que ele faça.

Para isso vamos usar como base o arquivo **config\_base.cfg** presente na pasta **OrCS/trace\_generator**, para não perdemos esse base, vamos copiá-lo para **config.cfg** antes de qualquer alteração:

```
>> cd ~/OrCS/trace_generator
```

```
>> cp config_base.cfg config.cfg
```

Agora todas as alterações serão feitas sobre o arquivo **config.cfg**

Essas são as configurações essenciais para qualquer programa:

- **sinuca\_tracer\_output**: Identificação da saída, início do nome dos arquivos de saída
- **program\_name**: Nome do programa sobre o qual o trace será feito
- **input\_name**: Um nome para a entrada (coloque qualquer coisa)
- **command**: Comando para executar a aplicação da qual serão extraídos os traços (lembre-se de colocar seus dados de entrada para que ela não fique esperando infinitamente)
- **pinplayhome**: Caminho completo até a pasta **OrCS/trace\_generator** (não coloque aspas)
  - Esse caminho pode ser descoberto digitando o comando **pwd** dentro da pasta **trace\_generator**
- **slice\_size**: Número de instruções que seus traços devem conter (Nem sempre fica igual, mas ele tenta deixar o resultado parecido)

Nota: Explicações mais detalhadas destes e de outros parâmetros estão contidas no arquivo **config\_base.cfg**

## Executando

Com as configurações definidas, o seguinte comando gera os arquivos base para seus traços:

```
>> python3 <caminho até o arquivo sinuca_pinpoints.py> --cfg <caminho até seu arquivo de configuração> --delete -lbsp
```

E esse comando obtém, dos arquivos base (que são gigantes), as partes mais significativas, de acordo com suas configurações:

```
>> python3 <caminho até o arquivo sinuca_pinpoints.py> --cfg <caminho até seu arquivo de configuração> -T
```

Exemplo:

1. **python3**  
/home/war/OrCS/trace\_generator/extras/pinplay/PinPoints/scripts/sinuca\_pinpoints.py  
--cfg /home/war/OrCS/trace\_generator/config.cfg --delete -lbsp
2. **python3**  
/home/war/OrCS/trace\_generator/extras/pinplay/PinPoints/scripts/sinuca\_pinpoints.py  
--cfg /home/war/OrCS/trace\_generator/config.cfg -T

Ao fim da execução serão gerados alguns arquivos .out.gz que são a saída do seu programa, ou seja, traços das partes mais significativas que podem ser utilizados no OrCS.

Apesar de tentador, não podemos criar vários tamanhos de traços com uma só preparação, visto que, a fase crucial para o tamanho dos traços é a primeira.