CET088 1 Semestre de 2017
Atividade de Laboratório L2: Desarmando uma Bomba Binária
Enviado: 07 Junho, Entrega: 17 de Junho

Leard Fernandes (`lofernandes@uesc.br`) é o responsável por este laboratório.

1 Introdução

O nefasto *Dr. Evil* implantou uma série de "bombas binárias" em algumas máquinas do laboratório. Uma bomba binária é um programa que consiste de uma sequência de fases. Em cada fase é esperado que você digite uma string particular no `stdin`. Se você digitar uma string correta. então esta fase é *desarmada* e a bomba segue para a próxima fase. Caso contrário, a bomba *explode*, imprimindo `"BOOM!!!"` e então termina. A bomba é desarmada quando cada uma das fases forem totalmente desarmadas.

Existem muitas bombas para nós desarmarmos, então estamos enviando uma determinada bomba para cada estudante desarmar. Sua missão, que você não tem escolha, é desarmar sua bomba até a data de entrega. Boa sorte, e seja bem vindo ao esquadrão antibomba!

Passo 1: Pegando sua bomba

Você receberá a sua bomba por email

Salve o arquivo `bomb.tar` num diretório protegido em que você planeja realizar a atividade. Então dê o comando: `tar -xvf bomb.tar`. Isto irá criar o diretório `./bomb` com os seguintes arquivos:

- `README`: Identifica a bomba e seu dono.
- `bomb`: O binário executável da bomba.
- `bomb.c`: O arquivo fonte com a rotina principal da bomba e uma mensagem amigável do D. Evil.

Passo 2: Desarmando sua bomba

O seu trabalho para este lab é desarmar sua bomba.

Você pode fazer a atividade em qualquer máquina. De fato, existe um rumor que Dr. Evil é realmente mau, e a bomba irá sempre explodir se você executá-la de qualquer maneira. Existem vários outros dispositivos de proteção contra intrusos na bomba.

Você pode utilizar muitas ferramentas para ajudar a desarmar a bomba. Dê uma olhada na seção **Dicas** para algumas dicas e ideias. A melhor forma é utilizar o seu depurador favorito para passar pelo binário desmontado.

Cada vez que sua bomba explodir ela irá notificar, e você irá perder 1/2 ponto (até o máximo de 20 pontos) no valor final do laboratório. Há consequências ao explodir a bomba, você deve ser cuidadoso!

As primeiras quatro fase valem 10 pontos cada. As fases 5 e 6 são um pouco mais difíceis, assim valem 15 pontos cada. Assim, o valor máximo que você poderá obter é de 70 pontos.

Embora as fases fiquem mais difíceis de desarmar a bomba, a experiência que você ganha em cada fase irá compensar o aumento da dificuldade. Entretanto, a última fase irá desafiar até os melhores alunos, assim, não espere até o último minuto para começar.

A bomba ignora entradas em branco. Se você rodar sua bomba com um argumento de linha de comando, por exemplo,

```
linux> {\em ./bomb psol.txt}
```

ela irá ler as linhas de entrada do arquivo `psol.txt` até que ele alcance o EOF (end of file), e então troca para `stdin`. Num momento de fraqueza o Dr. Evil adicionou este recurso para que você não precise redigitar as soluções de fases que você já tenha desarmado.

Para evitar detonar a bomba acidentalmente, você deverá aprender como realizar passos únicos (single-step) através do código de montagem e como inserir pontos de parada (breakpoints). Você irá precisar aprender como inspecionar ambos registradores e estado da memória. Um dos efeitos colaterais agradáveis de fazer o laboratório é que você ficará muito bom em usar um depurador. Esta é uma habilidade crucial que trará bons retornos no resto de sua carreira.

Logística

Este é um projeto individual. Todas as tarefas são eletrônicas. Esclarecimentos e correções serão postados na página do Curso/Grupo.

Tarefa

Você deverá enviar o arquivo de solução na seguinte forma `sbombn.txt`, onde n deverá ser igual ao número de sua atividade. E.g. Se você recebeu o arquivo `bomb1.tar` deverá enviar `sbomb1.txt` para o e-mail do responsável pela atividade com o título [Bomblab SWB]

Dicas *(Leia Isto!)*

Há varias forma de desarmar sua bomba. Você pode examiná-la em grande detalhe sem ao menos executar o programa e saber com grandes detalhes o que ela faz. Isto é uma técnica muito útil, mas nem sempre é fácil de se fazer. Você pode executar a sua bomba num depurador, vendo o que ocorre passo a passo, e usar essa informação para desarmá-la. Esta é provavelmente, a forma mais rápida de desarmar a bomba.

Apenas uma advertência, *não utilize força bruta!* você poderá escrever um programa que irá tentar todas as chaves possíveis para encontrar a correta. Isto não é bom por algumas razões:

- Você irá perder 1/2 ponto (até 20 pontos) a cada resposta errada que a bomba exploda.
- Cada vez que você erra uma mensagem é enviada para o servidor. Você poderá saturar a rede com estas pequenas mensagens, e assim, os administradores irão desabilitar o seu acesso ao computador.
- Vocês não sabem o tamanho das strings, bem sabem quais caracteres são. Mesmo que você assuma (incorretamente) que elas possuem 80 caracteres ou menos, e contendo apenas letras, você terá que adivinhar 26^{80} possibilidades para cada fase. Isto irá levar muito tempo para executa, e você não irá ter a resposta antes do término de entrega da atividade.

- `gdb`

  O GNU debugger, é uma ferramenta de depuração de comando de linha disponível virtualmente em toda plataforma. Você pode rastrear através de um programa linha por linha, examinar memória e registros, observar tanto o código fonte como o código de montagem (não estamos fornecendo o código-fonte para a maior parte de sua bomba), definir pontos de interrupção, definir pontos de exibição da memória (watchpoint) e escrever Scripts.

  Segue uma referência para utilizar o gdb

  `http://heather.cs.ucdavis.edu/~matloff/UnixAndC/CLanguage/Debug.html`

  - Para não explodir a bomba cada vez que você digita uma entrada errada você irá precisar aprender como configurar breakpoints.
  - Para documentação online, digite no prompt de comando "`help`" em `gdb` ou digite, "`man gdb`", ou "`info gdb`" no prompt Unix.

- `objdump -t`

  Isto irá imprimir a tabela de simbolos da bomba. A tabela de símbolos incluem os nomes de todas as funções e variáveis globais na bomba, os nomes de todas as chamadas de função da bomba, e seus endereços. Você irá aprender algo olhando estes nomes de funções.

- `objdump -d`

  Utilize para desmontar todo o código na bomba. Você poderá também apenas olhar funções individuais. Ler o código assembly pode lhe dizer como a bomba funciona.

O comando `objdump -d` lhe dá muitas informações, mas não lhe conta toda a história. Chamadas de funções do sistema são mostradas de forma encriptada. Por exemplo, a chamada para `sscanf` pode aparecer como:

```
8048c36:  e8 99 fc ff ff  call   80488d4 <_init+0x1a0>
```

Para determinar o que houve na chamada `sscanf`, você deveria desmontar com o `gdb`.

- `strings`

  Este utilitário irá mostrar as strings imprimíveis na sua bomba.

Procurando por uma ferramenta particular? Algo sobre a documentação ? Não esqueça que os comandos `apropos`, `man`, e `info` são seus amigos. Em particular, `man ascii` poder seu útil, `info gas` lhe dará mais do que você sempre quis saber sobre o GNU Assembler. Além disso, a web também pode ser um tesouro de informações. Se você ficar buggado, sinta-se à vontade para pedir ajuda ao seu instrutor.