| | | |
|---|---|---|
| Universidade Federal do ABC | MCZA034-14 – **Programação Segura** | Profª **Denise Goya** |
| | Prática 02 – Buffer Overflow e String de Formatação | |

Resolva as questões abaixo. Submeta suas respostas **individualmente** no Tidia.

I) Use o compilador **gcc** no Linux e o arquivo **buffov.c** (com código-fonte em C) para responder:

1. Quais são os problemas de segurança contidos na codificação desse programa? _____
2. Compile o programa usando as opções *default* do compilador **gcc**, acionando:
   $ **gcc buffov.c -o buffov**
   Você observou alguma mensagem de erro ou aviso? _____ O arquivo executável foi gerado? _____ Que conclusão você tira a respeito? _____
3. Execute o código gerado, acionando:
   $ **./buffov**
   Teste o programa com diferentes tamanhos para a string *path*. O que ocorre quando o tamanho é menor do que 10? _____ E quando o tamanho é igual a 20? _____ E com tamanho igual a 23? _____ E com tamanho maior que 50? _____ Para quais desses tamanhos, o programa responde corretamente? _____ E para quais desses tamanhos, o programa fornece respostas incorretas ou inesperadas? _____ Para cada erro, explique o que houve e dê uma justificativa _____
4. O que é *Stack Smashing*? _____
5. O que é *Stack Guard*? _____
6. Agora, compile o programa desligando a opção "*Stack Guard*" do compilador gcc, acionando o comando abaixo. (a opção "*Stack Guard*" é ativada por *default*, como na primeira compilação acima, a partir da versão 4.3.3 do gcc):
   $ **gcc buffov.c -o bufnosp -fno-stack-protector**
   Execute o código gerado, acionando:
   $ **./bufnosp**
   Teste o programa com diferentes tamanhos para a string *path*. Que diferenças você percebe nas execuções, em relação à compilação com a opção "*Stack Guard*"? _____
   Como você explica essas diferenças? _____
7. Quais são as vantagens e desvantagens em se ativar a opção "*Stack Guard*"? _____
8. Quais são as vantagens e desvantagens em desativar a opção "*Stack Guard*"? _____
9. Cite um exemplo em que poderia ser vantajoso e com baixo risco o não uso da opção "*Stack Guard*" _____

II) Use o compilador **gcc** e o arquivo **form.c** para responder:

10. Quais são os problemas de segurança contidos na codificação desse programa? _____
11. Compile o programa usando as opções *default* do compilador **gcc**, acionando:
    $ **gcc form.c -o form**

Você observou alguma mensagem de erro ou aviso? _____ O arquivo executável foi gerado? _____ Que conclusão você tira a respeito? _____

12. Agora, compile o programa desligando a opção Wformat-security do compilador gcc, acionando o comando abaixo:

    $ **gcc form.c -o form -Wno-format-security**

    Se tudo tiver dado certo, seu programa foi compilado, sem nenhum aviso. Teste-o, acionando:

    $ **./form**

    Para que serve a opção Wformat-security e qual a importância dela? ____
13. Teste o programa, fornecendo como entrada para *msg* as strings abaixo. Indique o que foi observado e suas interpretações:

    um %d _____

    um %d dois %d _____

    um %d dois %d três %d _____

    um %s _____

    um %s dois %s _____

    um %s dois %s três %s _____

    um %s dois %s três %s quatro %s _____

    %s%s%s%s%s%s%s%s%s%s%s%s%s%s%s _____

III) No sistema operacional Ubuntu e em vários Linux é possível controlar a aleatorização dos espaços de endereçamento, com a opção sysctl -w kernel.randomize_va_space

14. O que significa essa opção? _____
15. Qual a importância de manter ativa uma das opções de aleatorização? _____