FATEC Mogi Mirim

"Arthur de Azevedo"

Aluno(a):			**RA**:
Curso	Análise e Desenvolvimento de Sistemas	**Semestre: 5**	
Disciplina	Segurança da Informação		**Nota da Avaliação:**
Professor	Adriano Ricardo Ruggero		
Tarefas			Rubrica do Professor

Orientações gerais:

1 - Todas as tarefas podem ser realizadas em grupos de até dois alunos

2 - Submeta suas respostas/arquivos no endereço https://goo.gl/BmIUPJ

3 - Os arquivos devem estar em formato compactado (*.zip*), com senha, nomeados usando-se os RA's dos participantes.

4 - Envie a senha dos arquivos para o endereço de e-mail do professor.

1. (3 ½ Pontos) Desenvolva um programa em linguagem C chamado *colheita* que procura por nomes de arquivos em seu sistema e extraia palavras destes.

   O objetivo é usar o conteúdo dos arquivos para compilar uma lista de *palavras-chave* que poderiam ser usadas em um ataque de dicionário, presumindo que arquivos locais contenham informações pessoais.

   Os arquivos alvo do programa devem ser identificados por suas extensões. As extensões padrão são *.txt* e *.text*. Contudo, isto pode ser alterado por meio do parâmetro *-e*, que fará o programa aceitar várias extensões separadas por ":". Um exemplo: passando-se o parâmetro *-e txt:text:doc:asc*, informamos ao programa para que este busque por arquivos com as extensões *.txt*, *.text*, *.doc* e *.asc*.

   Seu programa deve considerar como *palavra* sequências de caracteres compostas por letras e números. Qualquer outro caracter deve ser considerado *separador de palavras* e o programa não deve considerar palavras repetidas, salvando no arquivo de saída apenas uma vez cada palavra encontrada.

   A pasta raiz usada para iniciar a busca deve ser passada via parâmetro *-d* e o arquivo de saída deve ser passado via parâmetro *-o*.

   Veja um exemplo: *colheita -e txt:text:asc -d /tmp/ -o password.txt*. Neste caso, o programa deve procurar por arquivos com as extensões *.txt*, *.text* e *.asc*, iniciar a busca a partir da pasta */tmp/* e salvar as palavras encontradas no arquivo *password.txt*.

   O programa pode fazer chamadas a comandos externos, como *find*.

   Arquivos a submeter: *colheita.c* e o arquivo de palavras-chave gerado.

2. (3 ½ Pontos) Desenvolva um programa que realize um ataque de força bruta para encontrar a senha de um arquivo protegido.

   Seu programa deve receber dois parâmetros, *-l* e *-f*. O parâmetro *-l* especifica o arquivo de dicionário e o parâmetro *-f* especifica o arquivo a ser atacado.

   O programa deve mostrar na tela a mensagem "A senha utilizada foi ****", onde "****" deve ser substituído pela senha real.

   Veja o exemplo: *bruteforce -l /tmp/list.txt -f /tmp/segredo.zip*. Neste caso, o programa deve ler as senhas do arquivo *list.txt* para tentar encontrar a senha que abre o arquivo *segredo.zip*.

   Use seu programa para descompactar o arquivo *segredo.zip* (está no SIGA!).

   Arquivo a submeter: bruteforce (em qualquer linguagem).

3. (3 Pontos) Gere um par de chaves GnuPG (pública e privada) e adicione uma *selfie* (sua) à chave pública (veja como no manual do gpg). Cada membro do grupo deve assinar a chave do outro membro.

   Crie um arquivo com seu nome, RA e curso. Assine este arquivo usando sua chave privada. Encripte seu arquivo assinado usando a chave pública do curso (arquivo SegInf2017.asc em https://goo.gl/BmIUPJ). O resultado deve estar em formato *ASCII armor*.

   Arquivos a submeter: sua chave pública (formato *ASCII armor*) e a mensagem encriptada.