Universidade da Beira Interior
Faculdade de Engenharia
Departamento de Informática

Segurança de Sistemas Informáticos
Guia para Aula Laboratorial 4
2º Ciclo em Engenharia Informática
2º Ciclo em Eng. Eletrotécnica e de Computadores
2º Ciclo em Matemática e Aplicações

Computer Systems Security
Guide for Laboratory Class 4
M.Sc. in Computer Science and Engineering
M.Sc. in Electrical and Computer Engineering
M.Sc. in Mathematics and Applications

Sumário
Exercícios subordinados ao tema da autenticação de entidades: autenticação fraca, razoavelmente forte e forte.

Summary
Exercises concerning the entity authentication subject: weak authentication, reasonably strong and strong authentication.

Pré-requisitos:
A maior parte das tarefas enunciadas em baixo requerem a utilização da ferramenta `OpenSSL` (`http://www.openssl.org/`) e de um compilador de ANSI C. Um ambiente linha de comandos Linux também facilita a execução de algumas dessas tarefas. Sugere-se, assim, o uso de uma distribuição comum de Linux, onde todas estas condições estarão provavelmente preenchidas ou pressupõe-se o acesso a um sistema com a possibilidade de instalar o *software* necessário.

1 Autenticação Fraca
Weak Authentication

Tarefa 1 *Task 1*

Crie uma palavra-passe com, pelo menos, 10 caracteres, que contenha pelo menos uma letra maiúscula, uma letra minúscula, um número e um caracter especial.

Password1*

Tarefa 2 *Task 2*

Registou-se num sistema *online* com um nome de utilizador e uma palavra-passe. Quando se liga pela primeira vez, o sistema mostra-lhe dois campos onde terá de colocar as informações que já registou.
Q1.: Quantos fatores de autenticação está a usar este sistema?
☒ 1 fator. ☐ 2 fatores. ☐ 3 fatores.
☐ 4 fatores. ☐ 5 fatores.

Q2.: Quais são os fatores de autenticação que está a usar neste caso?
☒ *The something you know factor.*
☐ *The something you are factor.*
☐ *The something you own factor.*
☐ *The someone you know factor.*
☐ *The someone you paid factor.*

Considere que usou uma palavra-passe com 5 letras minúsculas do alfabeto inglês (26 letras).
Q3.: Quantas palavras-passe diferentes existem neste caso?
☐ 10^5 ☒ 26^5 ☐ $26!$ ☐ e^{26} ☐ 2×5 ☐ C_5^{26}

Tarefa 3 *Task 3*

Escreva e execute um programa em ANSI C que itere uma variável `int i` do 0 até à resposta que deu na última questão da tarefa anterior (ver em baixo). **Q4.: Quanto tempo demora esse programa a executar?**
☐ O sol extingue-se antes de o programa acabar de executar.
☐ Alguns anos, mas ainda dentro da minha esperança de vida.
☐ Alguns meses.
☐ Alguns dias.
☐ Uh Oh... algumas horas!?
☐ Meros minutos!
☒ Nem um segundo!

```c
#include<stdio.h> int main(){
 int i = 0;
 for( i=0; i<11881376; i++) ;
 printf("Ended!\n\n");
}
```

Q5.: Se, em vez de 5 letras minúsculas do alfabeto inglês, pudesse também usar números e a palavra-passe tivesse 6 caracteres, quantas possíveis *passwords* haveria desta vez?

☐ 10^6 ☒ 36^6 ☐ $36!$ ☐ e^{36} ☐ 3×6 ☐ C_6^{36}

Tarefa 4 *Task 4*

Faça um gráfico em que o eixo dos xx representa o número de caracteres que uma palavra-passe pode ter e o eixo dos yy representa o número de palavras-passe que pode construir com esse número de caracteres. Se fizer esta tarefa bem feita, vai notar que **o gráfico vai mostrar aquela que é conhecida como a *exponential wall of brute force cracking*.**

Esquema de Lamport:

```
openssl dgst -sha256 -binary secret.txt >
pass1.txt

openssl dgst -sha256 -binary pass1.txt >
pass2.txt

....


openssl dgst -sha256 -binary pass3.txt >
pass4.txt
```

(Continuação da Tarefa 1)
Recorde a tarefa 1 e responda às seguintes questões relativamente à palavra-passe que construiu nessa tarefa. **Q6.: A sua palavra-passe começa com a tal letra maiúscula?**

☒ Sim, como é que adivinhou?
☐ Não.

Q7.: A sua palavra-passe contém uma palavra portuguesa ou inglesa?

☒ Sim, como é que adivinhou?
☐ Não.

Q8.: O caracter especial vem próximo do fim?

☒ Sim, como é que adivinhou?
☐ Não.

Q9.: O caracter especial aparece mais associado aos números que às letras?

☒Sim, como é que adivinhou?
☐ Não.

2 Autenticação Razoavelmente Forte

Resonably Strong Authentication

O sistema mencionado antes acabou por ser alvo de um ataque informático em que milhares de palavras-passe vazaram para a Internet. Logo de seguida, os seus autores resolveram implementar o esquema de Lamport, para *one-time passwords*. Consulte os apontamentos teóricos para relembrar esse esquema antes de evoluir para as tarefas seguintes.

Tarefa 5 *Task 5*

Considere que se havia registado no sistema com o segredo inicial (palavra-passe) `letmein`. Calcule a sequência de 4 *one-time passwords* usando o OpenSSL. Considere que a função de *hash* utilizada é o SHA256. Escreva a sequência de comandos em baixo:

```
> echo -n "letmein" > secret.txt
> openssl dgst -sha256 -binary secret.txt >
pass4.txt
```

Nota: a *one-time password* em `pass1.txt` deve terminar nos caracteres hexadecimais `5132 2cfe`. Use `> hexdump pass1.txt` para verificar se tudo está correto.

Tarefa 6 *Task 6*

Considere que já havia feito a autenticação 1 vez e o servidor está, de novo, a pedir a sua autenticação. Qual é o valor do *hash*, em hexadecimal, que terá de enviar desta vez?

☐ `e0b4 b1ed 0c57 3f1d 16e3 96f1 efae d8c8`
`b947 08f3 6c0c a15d 64d1 0be6 5132 2cfe`

☐ `d111 203b 3f13 01b3 7655 b43f 4512 966f`
`1490 cd5f 385c 8195 529f 344d 3960 126b`

☒ `32e5 3784 f124 ae4e 36a0 3355 fbac 3894`
`a9e5 ebcf 7444 13fe 7abb 1e5e 96c6 56e8`

☐ `8b1c 8ffe 1d80 7479 465c d031 ff9f c836`
`a32a c47f e4cc 94fc 8366 b3d7 b636 3230`

3 Autenticação Forte

Strong Authentication

Um sistema concorrente ao anterior quis gabar-se da segurança que oferece, e implementou o protocolo CHAP para autenticação dos seus utilizadores. Como perito na área, vieram pedir-lhe a sua opinião sobre o mesmo. As perguntas estão na próxima tarefa.

Tarefa 7 *Task 7*

Q10.: O que significa CHAP?

C hallenge Authentication Protocol H _____

A _____ P _____

Q11.: Este protocolo foi desenvolvido por quem?

☐ Linux Foundation. ☐ Microsoft Corporation.
☐ Apple Inc. ☒ Cisco Systems Inc.

Q12.: Este protocolo necessita que as comunicações sejam cifradas aquando da troca de mensagens de autenticação?

☐ Sim, necessita. ☒ Não, não precisa.

Q13.: Quantas mensagens precisam ser trocadas no âmbito do CHAP?

☐ 1 mensagem. ☐ 2 mensagens.
☒ 3 mensagens. ☐ 4 mensagens.

Q14.: Este protocolo requer que o segredo de autenticação fique guardado em ambos os intervenientes?

☐ Este protocolo não usa segredos de autenticação.
☒ Sim requer.
☐ Não, não requer.

Q15.: Por que é que este protocolo é considerado mais seguro que o esquema de Lamport (pode haver mais do que uma resposta certa)?

☐ Porque o próprio protocolo garante que cada autenticação é única.
☐ Porque o protocolo não requer que o segredo seja guardado no requerente e no autenticador.
☒ Porque o protocolo não está suscetível ao *small* n *attack*.
☒ Porque o protocolo já garante que a troca de mensagens é segura, sem ser preciso cifrar o canal onde estas são transmitidas.
☐ Porque este é um protocolo de conhecimento nulo.

Q16.: No cenário descrito nesta secção, quem é o requerente?

☐ É o sistema que faz a autenticação.
☒ É o utilizador que faz a autenticação.
☐ Neste cenário, não há requerente.

Tarefa 8 *Task 8*

Considere que um utilizador configurou o segredo de autenticação `superman` aquando do registo no sistema, e que o protocolo foi implementado com o SHA256. Como perito que é, e quando foi testar o sistema, colocou um *sniffer* de rede para poder capturar as mensagens que eram trocadas durante a fase de autenticação. Numa das tentativas, verificou que o servidor enviou a mensagem seguinte ao utilizador:

`031254826a5a68d52e`

Recorde o protocolo CHAP, consultando os apontamentos teóricos, e responda às seguintes questões.

Q17.: Qual é a composição da mensagem referida?

☐ A mensagem transporta o *hash* de `superman`.
☐ A mensagem transporta um número que é fixo e usado para todas a implementações do protocolo CHAP.
☒ A mensagem transporta a concatenação de um identificador e de um número aleatório de 64 bits.

Q18.: Que nome se dá à mensagem anterior?

☒ O desafio.
☐ O *magic number*.
☐ A GARRAAAAA!

Q19.: Que nome genérico se dá ao número aleatório que a mensagem referida no contexto desta tarefa carrega?

☐ Pitada de sal. ☒ *Nonce*.
☐ Vetor de inicialização. ☐ Colher de açucar.
☐ Segredo de autenticação. ☐ Chave de cifra.

Q20.: De acordo com a definição do protocolo CHAP, qual será a mensagem que o utilizador irá

usar para responder à solicitação do servidor?

☐ `e3b0 c442 98fc 1c14 9afb f4c8 996f b924 27ae 41e4 649b 934c a495 991b 7852 b855`

☒ `99dd 8d22 f30b 97fa 22af d3ab 3d74 f635 0a77 05d6 3b4b 251b c843 7b16 42a0 f233`

☐ `superman`

```
echo -n "031254826a5a ... > concatenated.txt

openssl dgst -sha256 concatented.txt > challhenge.txt
```

Tarefa 9 *Task 9*

Recorde, nos apontamentos das aulas teóricas, a forma de autenticação forte usando assinatura digital. **Q21.: Esta forma de autenticação requer que ambas entidades – requerente e autenticador – partilhem um segredo de autenticação?**

☐ Sim, requerem. ☒ Não, não precisam...

Considere que se estava a autenticar num sistema que usava assinatura digital como mecanismo de autenticação. O sistema já tinha a sua chave pública, correspondente ao par disponibilizado no *Moodle* juntamente com este guia laboratorial. Faça o *download* desse par de chaves para a sua máquina, e considere que era, portanto, seu. Considere que o servidor lhe havia enviado o seguinte *nonce*:
`sfdlfasomdf54a4`

Q22.: Quantas mensagens compõem o protocolo de autenticação que usa a assinatura digital?

☐ 1. ☒ 2. ☐ 3. ☐ 4.

☐ Às vezes 1, outras vezes 2, outras vezes 3. Em média, são 2,5.

Considere que o esquema de assinatura digital usado combina SHA1 com RSA. **Q23.: De acordo com a definição do protocolo de autenticação que usa a assinatura digital, qual deveria ser a sua resposta ao desafio incluído em cima (indique apenas os últimos 8 caracteres hexadecimais da resposta), para se conseguir autenticar no sistema?**

☐ `fa4d 81ef` ☐ `xfgd fa5f`

☒ `e254 6720` ☐ `1234 5678`

Q24.: Quais os comandos que utilizou para chegar à resposta anterior?

Nota: não se esqueça de ir tomando nota de todos os comandos que usou para elaborar este guia laboratorial. Este tipo de exercício é ideal para calhar numa ficha de avaliação.

Sugestão: habitue-se a usar o comando `> hexdump`, já que o teste prático irá pedir, bastantes vezes, o resultado hexadecimal de tarefas executadas com o OpenSSL.

```
echo -n ".." > nonce.txt

openssl dgst -sha1 -binary -sign pk-and-sk.pem nonce.txt > signature.txt

hexdump signature.txt
```

```
joaom@joaom:~/Documents/GitHub/ComputerSystemsSecurity-Lessons/Pratical/Lab_04$ echo -n 'sfdlfasomdf54a4' | openssl dgst -sha1 -out nonce.sha1

joaom@joaom:~/Documents/GitHub/ComputerSystemsSecurity-Lessons/Pratical/Lab_04$ openssl rsautl -sign -inkey pk-and-sk.pem -keyform PEM -in nonce.sha1 -out signature.bin

joaom@joaom:~/Documents/GitHub/ComputerSystemsSecurity-Lessons/Pratical/Lab_04$ xxd -p signature.bin | tr -d '\n' | tail -c 8
```