Aula TP - 30/04/2018

Cada grupo deve colocar a resposta às perguntas dos seguintes exercícios na área do seu grupo no Github até ao final do dia 07/Mai/2018. Por cada dia de atraso será descontado 0,15 valores à nota desse trabalho.

Note que estes exercícios devem ser feitos na máquina virtual disponibilizada. Caso já tenha a versão da máquina virtual utilizada nas últimas aulas, não precisa de fazer download da nova versão.

Instruções de *update* da máquina virtual, para quem já tem a máquina virtual utilizada nas últimas aulas:

1. Grave os ficheiros na diretoria Aula 11 para a sua máquina local na diretoria /home/user/Aulas/Aula11 .

Exercícios

1. Vulnerabilidade de inteiros

Experiência 1.1

Utilize o programa IntegerCheck.java para verificar os valores máximos e mínimos dos vários inteiros na sua máquina.

- Quantos bits tem cada um dos inteiros?

Experiência 1.2

Teste o programa IntegerError.java e verifique o que ocorre em situação de *integer overflow*.

- Teste com outros tipos de inteiro: byte, short, long
- E se precisar de utilizar inteiros maiores?

Experiência 1.3

Teste o programa IntegerCheck2.java e insira valores correctos e incorrectos.

- O que acontece quando lê um long e o atribuí a outro tipo de inteiro?
- O que acontecia se utilizasse o scan para o tipo correcto em cada um dos casos (nextByte(), nextInt(), …)?

Pergunta P1.1

Analise o programa overflow.c.

1. Qual a vulnerabilidade que existe na função *vulneravel()* e quais os efeitos da mesma?
2. Complete o *main()* de modo a demonstrar essa vulnerabilidade.
3. Ao executar dá algum erro? Qual?

Pergunta P1.2

Analise o programa underflow.c.

1. Qual a vulnerabilidade que existe na função *vulneravel()* e quais os efeitos da mesma?
2. Complete o *main()* de modo a demonstrar essa vulnerabilidade.
3. Ao executar dá algum erro? Qual?

Pergunta P1.3

Analise o programa erro_sinal.c.

1. Qual a vulnerabilidade que existe na função *vulneravel()* e quais os efeitos da mesma?
2. Complete o *main()* de modo a demonstrar essa vulnerabilidade.
3. Ao executar dá algum erro? Qual?

Projeto de desenvolvimento de software

Os alunos deverão utilizar o resto desta aula TP para continuarem o projeto de desenvolvimento de software.

O projeto 1 (Leilões online) será efetuado, em conjunto, pelos grupos 1, 6, 10, 11, 12.

O projeto 2 (Gestor de passwords com base em QrCodes) será efetuado, em conjunto, pelos grupos 2, 3, 4, 5, 7, 8, 9.

- Projeto 1 – Leilões online
    - Leilões online, com entrega de propostas em "carta fechada";
    - Pode ser uma extensão para software open source de leilões online.
- Projeto 2 – Gestor de passwords com base em QrCodes
    - Gestor de passwords, em que com base em QrCode apresentado pelo site, o telemóvel lê o QrCode e envia o user + password para desbloquear o acesso;
    - Pode ser uma extensão para software open source de gestão de passwords.

Nesta primeira fase, os dois grupos de projeto devem definir em traços gerais o projeto e as suas funcionalidades, e pensarem de que modo serão utilizado as técnicas criptográficas no projeto.

Como output desta fase, deverão ter um primeiro draft de:

- definição do projeto e suas funcionalidades,
- etapas e fluxos de comunicação / mensagens, podendo utilizar como exemplo o formato visto no segundo exemplo do voto eletrónico, na aula teórica. Esta componente deve conter um diagrama e uma parte textual de explicação do diagrama,
- identificar os passos efetuados para a concepção e desenvolvimento do projeto, de forma a seguir os princípios de "*privacy by design*" e "*data minimization*" do RGPD (Regulamento Geral de Proteção de Dados);
- identificar de que modo o software garante os direitos do titular dos dados, de acordo com o RGPD.

Estes pontos deverão fazer parte do relatório final do projeto.

SAMM (*Software Assurance Maturity Model*)

Nesta fase do projeto é-lhe pedido para, utilizando o ciclo de melhoria contínua do SAMM,

1. Avaliar a maturidade das práticas de segurança utilizadas no desenvolvimento de software deste projeto (Fase *Assess*);
2. Estabelecer o objetivo para cada uma das 12 práticas de segurança (Fase *Set the Target*), i.e., o nível de maturidade pretendido;
3. Desenvolver o plano para atingir o nível de maturidade pretendido, em quatro fases (Fase *Define the Plan*).

Para isso deverá utilizar a Toolbox (ficheiro excel) fornecida na diretoria Aula9, onde também encontrará mais informação relativa ao SAMM.

Note que:

- Para a Fase *Assess* deverá preencher a *sheet* "*Interview*";
- Para a Fase *Set the Target*, o grupo deverá discutir qual o *score* objetivo para cada uma das 12 práticas de segurança, que sirva de guia para atuar sobre as atividades mais importantes. Pode partir do princípio que a sua organização é uma startup que vai efectuar desenvolvimento de software na área de i) sistemas web seguros online (caso do projeto 1) e ii) sistemas de identificação eletrónica (caso do projeto 2). Se necessitar de outros pressupostos, indique-os na justificação à decisão tomada;
- Para a Fase *Define the Plan* deverá preencher a *sheet* "*Roadmap*", supondo que cada uma das fases tem 3 meses de duração. Tenha em conta o esforço necessário e a eventual dependência entre atividades em cada uma das fases.

A Toolbox (ficheiro excel) deverá fazer parte do relatório final do projeto, fornecendo o ficheiro excel como anexo ao relatório. Adicionalmente, no documento do relatório deverá incorporar um anexo em que indique a decisão da Fase *Set the Target* e a justifique.

Note que não há respostas certas nem erradas.