Aula TP - 04/06/2018

Estes exercícios já não têm um cariz de avaliação.

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 16 para a sua máquina local na diretoria /home/user/Aulas/Aula16 .

Exercícios

1. Validação de Input

Experiência 1.1

Analise os ficheiros InputValidation.cpp e InputValidation.java.

1. Explique os dois problemas: (i) utilização do cin/Scanner para leitura, (ii) acesso ao array
2. O que alterava?

Experiência 1.2

Analise o ficheiro WhileEx.java.

1. Qual o(s) problema(s) no input do programa?
2. Altere o programa de modo a validar todo o input e recuperar apropriadamente dos erros.

Experiência 1.3

Analise o ficheiro Input.cpp.

1. Qual o(s) problema(s) no input do programa ?
2. Altere o programa de modo a validar todo o input e recuperar apropriadamente dos erros.

Pergunta P1.1

Analise o ficheiro Input.java.

1. Qual o(s) problema(s) no input do programa ?
2. Altere o programa de modo a validar todo o input e recuperar apropriadamente dos erros.

Pergunta P1.2

Analise o programa filetype.c que imprime no ecran o tipo de ficheiro passado como argumento.

1. Existem pelo menos dois tipos de vulnerabilidades estudadas na aula teórica de "Validação de Input" que podem ser exploradas. Identifique-as.
2. Forneça o código/passos/linha de comando que permitem explorar cada uma das vulnerabilidades identificadas na linha anterior.
3. O que aconteceria se o seu programa tivesse permissões *setuid root*?

Pergunta P1.3

Analise o programa readfile.c que imprime no ecran o conteúdo do ficheiro passado como argumento, a que acrescenta o sufixo ".txt" de modo a garantir que só deixa ler ficheiros em texto.

1. Existem pelo uma vulnerabilidade estudada na aula teórica de "Validação de Input" (em conjunto com outra que já estudou) que permite que o programa imprima ficheiros que não terminam em ".txt". Explique.
2. Indique a linha de comando necessária para aceder ao ficheiro */etc/passwd*.

Experiência 1.4

Analise o ficheiro string_formato.c com o exemplo de vulnerabilidade de string de formato dado na aula, e o ficheiro string_formato2.c já sem essa vulnerabilidade.

1. Faça algumas experiências com vários valores de input tanto com o programa com vulnerabilidades como sem vulnerabilidades e tire as suas conclusões.

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.