Aula 8 - 9 de abril de 2018

Grupo 8

Pergunta P1.1

Segundo a informação disponível em https://informationisbeautiful.net/visualizations/million-lines-of-code/, a estimativa do número de linhas de código dos seguintes serviços é:

1. Facebook: 61 Milhões
2. Software de automóveis: 100 Milhões
3. Linux 3.1: 15 Milhões
4. Todos os serviços internet da Google: 2 Mil Milhões

Uma estimativa de bugs, tendo por base 5-50 bugs por 1000 linhas de código:

1. Facebook: 305 m-3.05 Milhões
2. Software de automóveis: 500 m-5 Milhões
3. Linux 3.1: 75 m-750 m
4. Todos os serviços internet da Google: 10 Milhões - 100 Milhões

Alguns destes bugs podem ser vulnerabilidades, mas não há uma proporção concreta, uma vez que há vários factores influenciadores, tais como: a linguagem de programação utilizada, o tipo de aplicação, ou até a experiência da equipa de desenvolvimento.

Pergunta P1.2

Exemplos de vulnerabilidades de projeto: erros na concepção da arquitectura, implementação incorrecta de protocolos criptográficos. Este tipo de vulnerabilidade pode implicar a reestruturação de todo o projecto, o que, dependendo da fase de desenvolvimento, pode representar elevados custos.

Exemplos de vulnerabilidades de codificação: utilização de software de terceiros, *buffer overflow*. Dependendo da modularidade, do bug e da documentação existente sobre o software, a correção de bugs nesta área pode ser bastante mais fácil do que nas vulnerabilidades de projecto.

Exemplo de vulnerabilidades operacionais: dificuldade na aplicação de patches, utilização de um ambiente vulnerável (e.g. sistema operativo). Quando o consumidor final utiliza o software num ambiente vulnerável, este fica também susceptível a outras vulnerabilidades. Neste caso, a correcção dos bugs passa pela aplicação de patches, cujo controlo é mais complicado de

gerir, uma vez que a sua aprovação/instalação está dependente do cliente. Se o ambiente vulnerável for do lado do servidor, a correcção é mais controlada; no entanto, dependendo da gravidade, podem ser necessárias mudanças na arquitectura, que podem ser bastante custosas.

Pergunta P1.3

Uma vulnerabilidade de dia-zero é uma vulnerabilidade desconhecida até ser revelada/descoberta pela entidade comprometida, ao contrário de uma vulnerabilidade não dia-zero, que é de conhecimento da mesma. Esta vulnerabilidade representa um maior risco para o utilizador, uma vez que o distribuidor do software comprometido não pode aplicar um *patch* ou aconselhar os utilizadores a seguirem certas práticas que podem reduzir a probabilidade de serem alvo de um ataque. Além disso, um atacante pode ficar indefinidamente a explorar esta vulnerabilidade, até que ela seja descoberta (*zero-day attacks*).