Engenharia de Segurança

Aula 8 - 09/04/2018

Vulnerabilidades de codificação

Afonso Fontes (pg35389) Bruno Carvalho (a67847) Mariana Carvalho (a67635)

16 de Abril de 2018

Universidade do Minho

Pergunta 1

1.

Para calcular o número de bugs em cada pacote de software apresentado, recorreu-se a uma fórmula simples que relaciona a quantidade estimada de bugs (que pode variar entre 5 e 50 dependendo do rigor na escrita do código) por cada 1000 linhas de código fonte. Assim, tem-se que:

- Facebook

  61 milhões de linhas de código fonte

  305 mil a 3.050 milhões de bugs

- Software de automóveis

  100 milhões de linhas de código fonte

  500 mil a 5 milhões de bugs

- Linux 3.1

  15 milhões de linhas de código fonte

  75 a 750 mil bugs

- Serviços Internet da Google

  2 biliões (2*10) de linhas de código fonte

  10 a 100 milhões de bugs

2.

É difícil estimar o número de vulnerabilidades no código fonte de cada projeto, visto ser algo que depende de muitos fatores, como a linguagem de programação escolhida (linguagens de baixo nível são mais suscetíveis a exploits), a experiência e conhecimento do programador da mesma, ou até o tipo de aplicação desenvolvida, entre outros. Após pesquisa na National Vulnerability Database, até ao momento foram identificadas (e muitas das quais já corrigidas) **68** vulnerabilidades relacionadas com o **Facebook**, **11** vulnerabilidades relacionadas com o **Linux 3.1**, e **4291** vulnerabilidades relacionadas com serviços **Google**.

Pergunta 1.2

Vulnerabilidades de projeto

Dois exemplos de vulnerabilidade de projeto são a **CWE-6: J2EE Misconfiguration: Insufficient Session-ID Length** (o J2EE usa valores de Session-ID de apenas 64 bits, o que pode facilitar a um atacante adivinhar/prever um determinado Session-ID) e **CWE-13: ASP.NET Misconfiguration: Password in Configuration File** (o ASP.NET guarda passwords num ficheiro em plaintext).

Estas vulnerabilidades podem ser particularmente problemáticas de resolver, pois tendo sido inseridas na fase de conceção do projeto, a sua resolução pode obrigar a alterações de fundo em vários componentes do projeto. Por exemplo, na CWE-6 referida, alterar a dimensão dos Session-ID para 128 bits corrigiria a vulnerabilidade, mas poderia obrigar a alterar todos os componentes do projeto que foram "pensados"de início para lidar com valores de Session-ID de 64 bits.

Vulnerabilidades de codificação

São exemplos deste tipo de vulnerabilidade a **CWE-24: Path Traversal: '../filedir'** (possibilidade de incluir a sequencia '../' num input para um filepath e assim possibilitar a um atacante aceder a diretorias às quais normalmente não deveria ter acesso) e **CWE-82: Improper Neutralization of Script in Attributes of IMG Tags in a Web Page** (execução indevida de scripts dentro de tags html IMG).

Estas vulnerabilidades podem geralmente ser corrigidas através de um patch. No entanto, o lançamento de um patch por parte de uma empresa de desenvolvimento de software tem aspectos negativos, por exemplo, os custos inerentes ao seu desenvolvimento e publicação, a possibilidade de um atacante descobrir a vulnerabilidade que o patch resolve (deixando mais expostos os utilizadores que ainda não instalaram o patch) ou a possivel introdução de novas vulnerabilidades.

Vulnerabilidades operacionais

São exemplos de vulnerabilidades operacionais a **CWE-520: .NET Misconfiguration: Use of Impersonation** (correr aplicações .NET com privilégios demasiado elevados, o que pode facilitar um ataque ou tornar um potencial ataque mais perigoso) e **CWE-554: ASP.NET Misconfigu-**

ration: Not Using Input Validation Framework (utilização de aplicações ASP.NET sem a devida validação de input).

Estas vulnerabilidades são potencialmente as mais simples de resolver e podem geralmente ser resolvidas diretamente pelo utilizador, sem necessidade de intervenção da empresa que desenvolveu o software (dado um utilizador/administrador com o devido *know-how*).

Pergunta 1.3

Uma vulnerabilidade de codificação **0-day**, é uma vulnerabilidade que embora já seja conhecida por um grupo restrito de entidades, não é ainda do conhecimento da entidade cuja vulnerabilidade afeta diretamente. Normalmente, quando uma nova vulnerabilidade é descoberta, é-lhe de imediato atribuído um **CVE**, embora este possa demorar algum tempo a ser emitido de forma pública, a entidade afetada é informada da existência de tal vulnerabilidade de forma a poder lançar um *patch* que a corrigida antes que o respetivo **CVE** se torne público. No caso de uma vulnerabilidade **0-day**, este processo não ocorre, daí a designação utilizada no jargão da segurança informática, 'Zero' remete ao número de dias que a entidade em causa teve disponível para elaborar um **patch** de correção para a respetiva vulnerabilidade.