Universidade do Minho

Mestrado em Engenharia Informática

Engenharia de segurança

Ficha de exercício 9

Grupo 1

Rui Carlos Azevedo Carvalho - PG47633
Daniel Barbosa Miranda - PG47123
Ana Luísa Lira Tomé Carneiro - PG46983

Vulnerabilidade de codificação

Pergunta 1.1 - Common Weakness Enumeration (CWE)

Características

[1]

Esta fraqueza ocorre quando é possível efetuar operações de escrita em localizações fora dos limites alocados para um determinado buffer.

Como consequência, dá-se a possibilidade de ocorrer corrupção de dados, *crash* ou execução de código não autorizado.

Linguagens

[1]

- C
- C++
- Class: Assembly

Código - Exemplos

[1]

Neste primeiro caso, temos a criação de um array capaz de conter 3 elementos. De seguida são atribuídos valores em 4 posições diferentes do array, sendo a última atribuição já se encontra fora dos limites, uma vez que apenas era possível existirem 3 posições.

```
int id_sequence[3];

/* Populate the id array. */

id_sequence[0] = 123;
id_sequence[1] = 234;
id_sequence[2] = 345;
id_sequence[3] = 456;
```

Quanto a este segundo exemplo, a função `memcpy` utiliza a função `returnChunkSize` e, esta última, poderá retornar `-1`. Este valor não é verificado antes da sua chamada em `memcpy`, pelo que o tamanho passado a esta função será negativo fazendo com que seja efetuada uma cópia de uma quantidade superior de memória do que aquela existente no buffer.

```
int returnChunkSize(void *) {

/* if chunk info is valid, return the size of usable memory,

* else, return -1 to indicate an error

*/
...
}
int main() {
...
memcpy(destBuf, srcBuf, (returnChunkSize(destBuf)-1));
...
}
```

Tecnologias

[1][2]

Uma tecnologia que pode apresentar-se vulnerável a esta fraqueza trata-se do *Bluetooth*, como será visto na vulnerabilidade a seguir.

CVE-2020-0022

[1][2]

A vulnerabilidade mais recente associada a esta fraqueza é *CVE-2020-0022* e, segundo a descrição desta, na implementação de Bluetooth para telémoveis, ao calcular o tamanho de um pacote, não era incluído o *offset* do mesmo e portanto era possível para um atacante enviar mais dados que o esperado e efetuar um ataque de execução de código remoto, sem a necessidade de interação de utilizadores ou qualquer privilégio.

Por sua vez as versões `Android` afetadas por esta vulnerabilidade foram:

- Android-8.0
- Android-8.1
- Android-9
- Android-10

Pergunta P1.2

Vulnerabilidades são defeitos do sistema (software ou outro) que podem ser explorados por um atacante com o objetivo de violar a política de segurança. No

caso das vulnerabilidades de software estas podem ser classificadas nas categorias **vulnerabilidade de projeto**, **vulnerabilidade de codificação** e **vulnerabilidade operacional**.

As vulnerabilidades de projeto podem ser encontradas na fase de requisitos e na fase de desenho e dizem respeito à falta de requisitos de segurança levantados e à falta de segurança na escolha da arquitetura e ferramentas para a posterior implementação. Desta forma, possíveis vulnerabilidades que podem pertencer a esta categoria são por exemplo, definição de uma autenticação fraca na fase dos requisitos e a escolha de uma comunicação insegura através do não uso do TLS (*Transport Layer Security*). Como estas vulnerabilidades foram provocadas numa fase inicial do projeto são dificilmente corrigidas em fases posteriores, sendo que para as corrigir é necessário alterações profundas no sistema. Por exemplo, para o caso da autenticação fraca à necessidade de reformular o processo de autenticação na sua íntegra implicando mudanças nas fases seguintes se for implementado um modelo em cascata.

As vulnerabilidades de codificação são encontradas na fase de implementação e dizem respeito a vulnerabilidades que são provocados devido a problemas, erros ou *bugs* na codificação das funções e métodos. Assim, possíveis vulnerabilidades que podem pertencer a esta categoria são a não verificação do tamanho dos *buffers* que podem provocar ataques de *buffer overflow* ou a não validação dos *inputs* recebidos pelo o sistema pode provocar execução de código arbitrário e pode provocar acessos na memória. Para corrigir este tipo de vulnerabilidades é necessário acrescentar ou modificar métodos para que as vulnerabilidades sejam resolvidas. Com a utilização de metologias que visam a manutenção do código por exemplo através de um modelo em cascata, este tipo de modificações no código iam afetar menos fases, pois esta fase de implementação já é uma das fases finais do projeto.

Finalmente. as vulnerabilidades operacionais são causada pelo ambiente no qual o software é executado ou pela sua configuração, fazendo parte por isso das fases finais de instalação e execução do sistema. Vulnerabilidades que estão categorizadas como operacionais podem ser a existência de contas sem *passwords* numa base de dados ou configurações por *default* que existem no processo de instalação do sistema. Como estas vulnerabilidades são encontradas numa fase já avançada do projeto podem ser facilmente mitigadas, contudo pode haver configurações implementadas em fases anteriores que ao serem alteradas podem afetar as fases seguintes, provocando mudanças em cadeia.

Pergunta P1.3

As vulnerabilidades de um sistema podem ser encontradas por várias entidades, nomeadamente, o produtor do software, o seu utilizador, entre outros. Em alguns casos as vulnerabilidades são tornadas públicas e por isso toda a comunidade tem acesso a informação sobre a mesma.

No entanto, as vulnerabilidades de dia-zero são vulnerabilidades de um determinado sistema que apenas são conhecidas por um meio restrito de pessoas ou entidades, ou seja, não existe informação pública sobre a mesma e por isso a comunidade de segurança em geral desconhece a sua existência. Desta forma, como o próprio desenvolvedor do software não tem conhecimento destas vulnerabilidades então não consegue corrigi-la o que expõe vários sistemas a ataques de dia-zero, que são ataques bastante eficazes até

contra sistemas bem protegidos com equipas de segurança competentes, pois não sabem que esta vulnerabilidade existe e está a ser explorada. O objetivo destas comunidades restringirem a informação da vulnerabilidade a uma comunidade restrita é a de venda posterior na *dark web* ou a exploração da mesma para realizar um ataque informático. Para além disto, podem ser utilizadas por meios militares para desenvolver ataques a certos sistemas.

De facto, as outras vulnerabilidades distinguem-se das vulnerabilidades dia-zero pelo facto do seu conhecimento ser público à comunidade, sendo que, a probabilidade de estarem catalogadas é quase certa a partir do momento em que se tornam públicas.

Bibliography

[1] "CWE-787: Out-of-bounds Write" [online]. Disponível em: https://cwe.mitre.org/data/definitions/787.html [Acedido em abril de 2022].

[2] "CVE-2020-0022 Detail" [online]. Disponível em: https://nvd.nist.gov/vuln/detail/CVE-2020-0022 [Acedido em abril de 2022].