U.PORTO

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Professor João Vilela

Segurança e Privacidade

Terceiro Trabalho: Mecanismos de Mitigação de Riscos

Maio de 2022

Trabalho realizado pelo Grupo S, constituído por:

Pedro Leite - 201906697

Pedro Carvalho - 201906291

Índice

1. Introdução

O COP-MODE é um projeto que recruta participantes, para instalar uma aplicação, chamada CM-AR. Após a instalação, o participante deverá executar a aplicação, esta vai coletar uma série de dados do *smartphone* do participante. A equipa do COP-MODE vai instalar as aplicações coletadas tal como o CM-NPM (gestor de privacidade) e um gestor de permissões, num novo *smartphone*, que vai ser emprestado ao participante para utilizar durante 1 semana. Sendo que o projeto não tem qualquer tipo de mecanismos de segurança, é o nosso objetivo mitigar os eventuais riscos encontrados.

2. Mitigação dos Riscos Encontrados

Os riscos propostos, para os quais vão ser procuradas maneiras de serem mitigados são: (2.1.) *eavesdropping* dos dados entre os *smartphones* e o servidor, (3.2.) vazamento de dados através do acesso não autorizado ao servidor, o acesso aos dados deve ser limitado a utilizadores permitidos e o servidor deve ser acedido remotamente e não exposto ao exterior, (3.3.) dados acedidos por terceiros e (3.4.) vazamento de informações sensíveis, nomeadamente aplicações utilizadas.

2.1. *Eavesdropping* dos Dados entres os *Smartphones* e o Servidor

Um ataque de eavesdropping ocorre quando há uma infiltração na comunicação entre um utilizador e um servidor, sendo que o infiltrado tomou vantagem das fraquezas dessa comunicação para efetuar o ataque.

Para mitigar este risco utilizamos encriptação. Mesmo se um infiltrado conseguir vazar os dados, estes não vão conseguir ser lidos pelo mesmo, sem o acesso à chave de encriptação.

Portanto tocas as comunicações devem ser feitas com HTTPS (combinação do HTTP com SSL/TSL). Este protocolo encripta: URLs do documento requisitado, o conteúdo do documento, dos formulários do navegador e dos cabeçalhos, e as cookies, através de uma chave que é diferente para cada conexão, que foi definida no início da conexão pelo protocolo *Handshake* do TLS.

Este protocolo também aplica autenticação para cada pacote recebido do servidor, através de uma nova chave de encriptação, de maneira a evitar ataques *Man-in-the-Middle*, já que o servidor iria detetar alguma alteração nos dados.

2.2. Vazamento de Dados Através do Acesso não Autorizado ao Servidor

Para limitar o acesso aos dados a utilizadores autorizados, tal como no tópico anterior, primeiro temos de estabelecer a ligação usando HTTPS (sobre SSL/TLS), isto porque podemos usar um tipo de *User Authentication Protocol* que é o PAP (*Password Authentication Protocol*). Este protocolo garante autenticação quando o cliente envia um pacote com as suas credenciais (*username* e *password*) e tenha a confirmação do servidor. No entanto, estes são enviados em c*leartext* por isso o PAP só por si é altamente inseguro mas quando a ligação é feita através de HTTPS é garantido que estes dados não são sujeitos a ataques de eavesdropping, por exemplo.

2.3. Dados Acedidos por Terceiros

Para impedir que não seja possível associar dados a um utilizador, por exemplo, um e-mail, pode ser usada uma h*ash function.* Em vez de se guardar o e-mail em concreto, guarda-se o h*ash value* gerado pela função. Quando o utilizador quiser apagar os dados, pode introduzi-los e se o h*ash value* gerado por esses dados, quando passados pela h*ash function,* que gerou o que ficou guardado, forem iguais aos dados que estão no servidor, então garante-se que aquele cliente é o proprietário dos dados e podem ser apagados. Já que duas strings iguais, que passem pela mesma *hash function* vão dar o mesmo *hash value*.

2.4. Vazamento de Informações Sensíveis

Para mitigar os danos de um possível vazamento de informações sensíveis, à semelhança do ponto 2.1., utilizamos encriptação dos dados transmitidos entre o *smartphone* e servidor.

Para a encriptação, similarmente ao tópico anterior, podemos usar uma h*ash function*. E em vez de ser guardado o nome da aplicação em *raw format*, guardamos o seu h*ash value* e desta forma conseguimos saber que no *data set,* está guardado "x" vezes a aplicação, mas não é sabido o nome real da aplicação.

Porque, como explicado anteriormente, strings iguais passadas pela mesma *hash fucntion* dão um *hash value* igual e diferente da string original.

3. Conclusão

Após apresentar propostas para mitigar todos os problemas propostos, é importante realçar que muito difícil será possível aplicar um mecanismo totalmente seguro. Vai haver sempre falhas. As medidas propostas vão ajudar os utilizadores, a sentirem-se mais seguros na utilização diária dos seus *smartphones,* já que reduzimos os riscos propostos.