Ficha 2
Tecnologia de Segurança

Ana Margarida Campos A85166
Nuno Pereira PG42846

27 de Outubro de 2020

Threat model* do sistema *Precision Agriculture System

O sistema *Precision Agriculture System* é composto por vários dispositivos que contribuem para o seu funcionamento, tais como WSN que são sensores integrados para aquisição de dados, *Gateways* que gerem os dados recebidos pelos nodos dos WSN, o *Back-end* que recebe e armazena os dados dos diferentes nodos dos *Gateways* na *cloud* e o *Front-end* que oferece dois modos: um para os agricultores e outro para especialistas. Este sistema possui algumas vulnerabilidades que podem ser exploradas.

Para inicializar o *Threat model* fizemos um *data-flow diagram*:

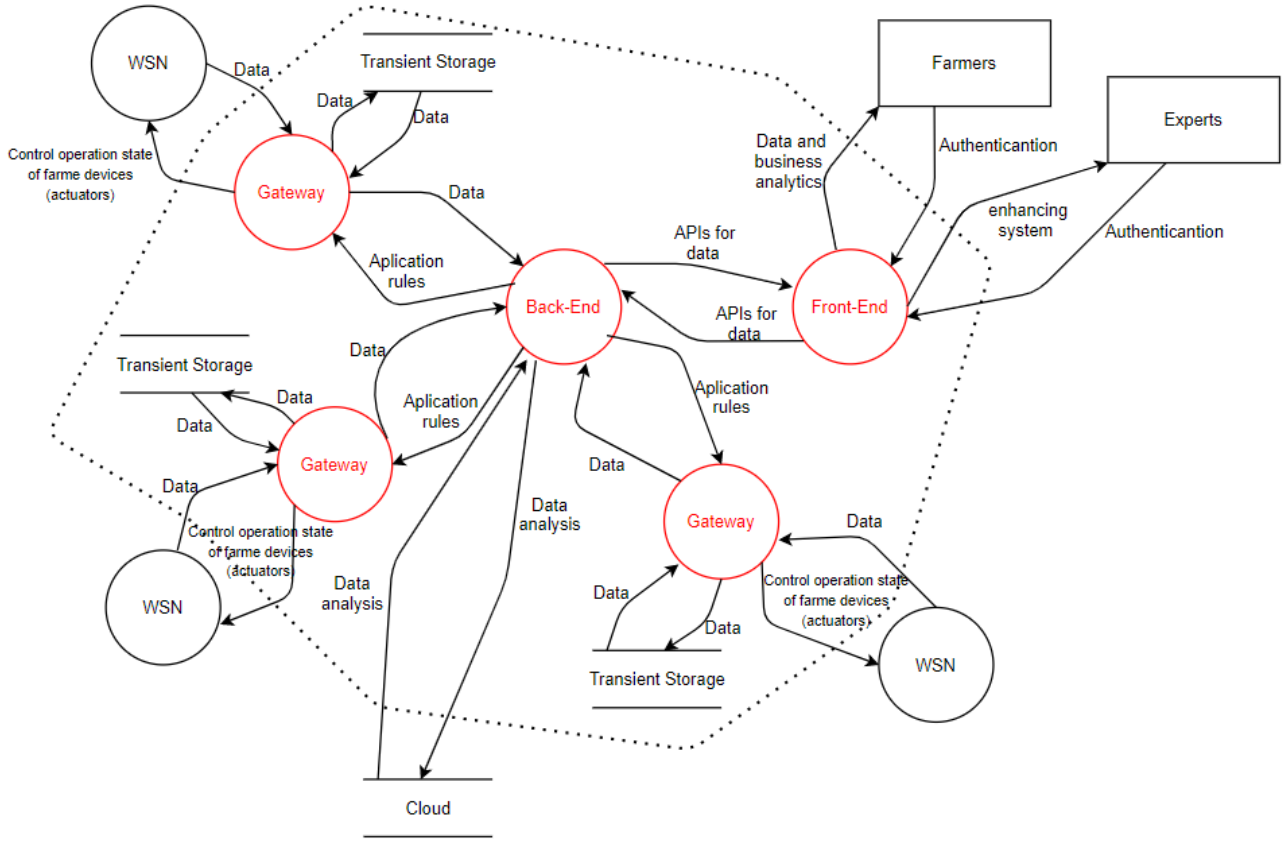

Figura 1: Data-Flow diagram

Para completar o *Threat model* fizemos a análise do modelo **STRIDE** de cada um dos componentes do sistesma de maneira a identificar algumas fraquezas e vulnerabilidades do mesmo.

Wireless sensor and actuators nodes (WSN):

Spoofing
Uma vulnerabibilidade poderá surgir se por exemplo um utilizador malicioso inventar algum tipo de equipamento que se faça passar por um sensor ou atuador e que envie informação incorreta.
Uma maneira de resolver esta vulnerabilidade seria através de uma autenticação correta ao sistema.

Tampering
Como a comunicação entre os WSN e os *gateways* é feita por interfaces sem fio (*wireless*), um atacante poderá intercetar a comunicação, ler os pacotes enviados e alterar a respetiva informação.
Uma possível solução seria cifrar os dados na sua transição de maneira a que o atacante não conseguisse retirar informações sobre os mesmos.

Information Disclosure
Um atacante poderá intercetar a comunicação entre os WSN e os *gateways* e divulgar as informações para entidades não autorizadas.
Uma das maneiras de resolver esta vulnerabilidade seria a cifragem dos dados durante a comunicação.

Basestation/gateway:

Spoofing
Pode ocorrer um *IP-Spoofing* quando o IP original é trocado pelo do atacante, permitindo ao mesmo obter informações ou enviar informações incorretas.
Uma solução seria filtrar pacotes cujo IP de origem não corresponde a nenhum dos da rede do sistema. Deste modo as insconsistências eram detetadas e o *IP-Spoofing* prevenido.

Tampering
Como a comunicação entre os WSN e os *gateways* é feita por wireless, um atacante poderá modificar dados na rede.
Uma possível solução centraria-se na cifragem dos dados durante a comunicação.

Information Disclosure
O atacante poderá intercetar a comunicação e consequentemente ter acesso a informações particulares. Pode também vir a divulgar estas informações a outras entidades.
Para evitar este tipo de vulnerabilidades é importante existir uma boa cifragem de todos os dados pertinentes.

Denial of Service
O atacante poderá bloquear as comunicações entre os *gateways* fazendo com que os pacotes sejam descartados. Isto irá indisponibilizar os dados aos utilizadores do sistema.
Para tentar contornar este problema seria importante existir redundância dos dados, ou seja, estes estarem disponibilizados em vários locais.

Elevation of Privilege
Um utilizador com intenções maliciosas pode intercetar o tráfego dos dados, fazer-se passar pelo administrador do sistema e enviar um pedido para aumentar os seu previlégios de maneira a executar ações que não está autorizado a fazer.
Para garantir que este tipo de ataques não aconteça é necessário ter mecanismos de autenticação apropriados.

Cloud:

Spoofing
Um atacante pode a partir de força bruta tentar autenticar-se com os dados de um utilizador comum.
Uma maneira de resolver esta vulnerabilidade seria através de uma atenticação correta ao sistema.

Tampering
Os dados ou ficheiros submetidos para a cloud poderão ser modificados por um atacante.
Uma forma de prevenção seria ter um *backup* do dados da cloud. Outra forma seria converter a cloud num *WORM system (Write Once Read Many).* Num sistema deste género os dados são escritos e nunca poderão ser modificados.

Repudiation
Um utilizador pode negar que submeteu ou eliminou alguns dados da cloud.
Isto seria resolvido implementando um sistema de *loggin* adequado.

Information Disclosure
Um atacante pode ter acesso à cloud, obter informações pertinentes da mesma e se pretender divulgar essa informação.
Uma solução para esta vulnerabilidade seria a cifragem de informação importante.

Denial of Service
O atacante pode fazer um *Syn flood attack* onde ele incia uma conexão mas não a finaliza. Esta conexão pode fazer com que o servidor gaste recursos à espera de uma conexão meia-aberta e como consequência o sistema fica irresponsivo.
Uma maneira de resolver seria lidar com o processo de *handshake* na cloud e reter a conexão com o servidor de destino até que o *handshake* fosse concluído.

Backend:

Tampering
O atacante pode conseguir modificar o código de forma a extrair informações ou mesmo eliminar dados importantes.
Uma solução seria ter um *backup* do código do backend.

Repudiation
Tendo acesso ao Backend os atacantes podem apagar ou truncar o ficheiro de *logs* de maneira a esconder as suas ações maliciosas.
De forma a evitar esta situação, é necessário que exista um sistema de permissões que não deixe ninguém alterar o ficheiro de *logs.*

Information Disclosure
O atacante tendo acesso ao Backend pode retirar informações sobre o código e até pode divulgar as mesmas a outras entidades.

Uma solução será usar linguagens como C# que nos protegem contra fraquezas por exemplo do tipo overflow e na generalidade fornecem uma proteção maior que outras linguagens nomeadamente o c++. Para tornar ainda mais seguro podemos usar APIs.

Denial of Service
Um atacante poderá intercetar e encerrar a comunição entre o Backend e os *Gateways* através de um *Syn flood* de modo a que os *Gateways* descartem os pacotes que os sensores enviam.
Esta vulnerabilidade pode ser evitada recorrendo a *firewalls* que consigam distinguir pacotes reais de pacotes de *flooding*.

Frontend:

Tampering
O atacante pode conseguir modificar o código de forma a extrair informações ou mesmo eliminar dados importantes.
Uma solução seria ter um *backup* do código do frontend.

Information Disclosure
O atacante tendo acesso ao Frontend pode retirar informações sobre o código e até pode divulgar as mesmas a outras entidades.
Uma possível solução, que também foi referida no Backend, será usar linguagens como C# que nos protegem contra fraquezas por exemplo do tipo overflow e na generalidade fornecem uma proteção maior que outras linguagens nomeadamente o c++. Para tornar ainda mais seguro podemos usar APIs.

Elevation of Privilege
Um atancante tendo acesso ao código do Frontend poderá modificar os dados e executar ações em modo de administrador.
Para evitar esta situação é necessário que haja mecanismos de autorização apropriados que impençam a alteração de tais dados.

Farmers and Experts:

Spoofing
Um atacante pode tentar autenticar-se com os dados de um agricultor ou especialista (através por exemplo de *fishing* ou *brute force*).
Uma maneira de resolver esta vulnerabilidade seria através de uma atenticação correta ao sistema.

Tampering
Poderá ocorrer modificação de alguns dados relacionados com a quinta, agricultores, sensores, etc.
De maneira a evitar a modificação das diversas informações deveria apenas se dar permissões de leitura aos utilizadores.

Repudiation
Qualquer ação executada pode ser negada.
Uma solução para este problema será ter um registo com a autenticação dos utilizadores ao sistema (indicando o nome do utilizador e a hora que acedeu).

Transiente storage

Tampering
Um atacante poderá modificar informações presentes nesta memória.
Uma forma de prevenção seria ter um *backup* dos dados presentes nesta memória. Outra forma seria ter um sistema de permissões.

Information Disclosure
Um atacante ao ter acesso a esta memória temporária pode recolher informações nela contidas e até partilhá-las com outras entidades.
Uma solução consiste na cifragem da informação pertinente.