Threat Model for a Precision Agriculture System

Henrique Faria and Paulo Rosa A81139

Departamento de Informática, Universidade do Minho

Resumo

Este relatório pretende modelar as ameaças que um sistema, neste caso o Agriculture Precision System, poderá enfrentar.

De forma a ajudar a criação do modelo começar-se-á pela descrição de alto nivel do sistema completo. Seguido de uns modelos um pouco mais detalhados e respetivos subsistemas pertencentes ao sistema principal.

Seguidamente, analisar-se-á cada subsistema seguindo a metodologia STRIDE para identificar as ameaças mais pertinentes. Note-se que a análise de ameaças é feita tendo como foco o software principalmente.

O primeiro subsistema a ser analisado será o Front-End, seguido pelo subsistema da Farm e fazendo algumas considerações sobre o Back-end(Cloud). Juntamente com a análise, aconselhar-se-ão algumas soluções para as ameaças identificadas.

Palavras-chave: Front-End · Farm Model System · Spoofing · Tampering · Repudiation · Information Disclosure · Denial of Service · Elevation of Privilege.

1 Model System for the Precision Agriculture System

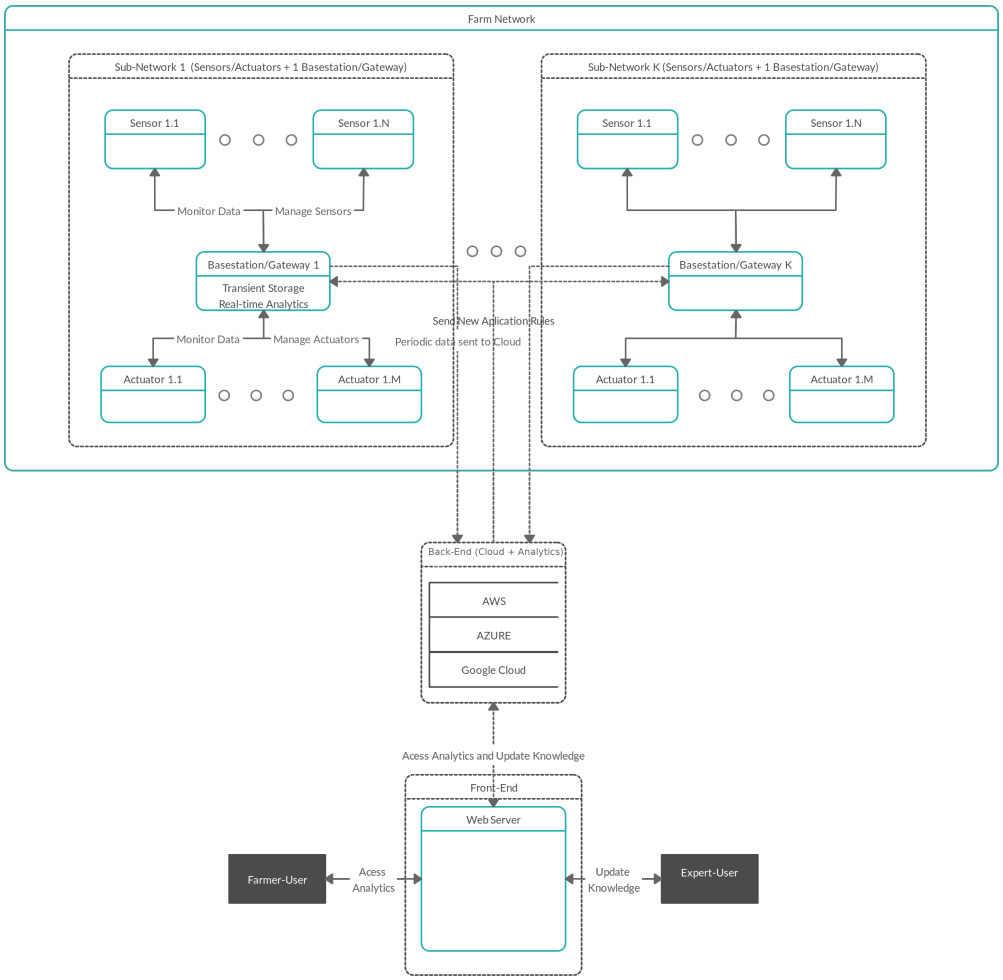

Figura 1. Model System for the Precision Agriculture System

O sistema encontra-se dividido em três boundaries principais a cloud, o sistema de agricultura e o front-end. Ainda dentro da boundary do sistema de agricultura divide-se em pequenos sub-sistemas que correspondem ao conjunto de uma basestation/gateway junto com os sensores e actuators.

A partir do modelo entende-se que os sub-sistemas referidos anteriormente não comunicam entre si, só com a cloud , de forma a atualizar as configurações das basestations/Gateways e a transmitir a informação recolhida pelos sensores

á front-end. Também o back-end, formado pelo armazenamento em cloud e o modulo de analytics, só deve comunicar com as basestations/gateways e com o servidor do front-end fornecendo uma API para comunicar com o mesmo. Finalmente, temos o front-end que para além da Cloud só comunica com dois tipos de utilizadores externos ao sistema que são os agricultores e os Experts.

É importante salientar que o back-end deste sistema é responsabilidade dos fornecedores de serviços indicados na figura, o que, por si só, já fornece algumas garantias de segurança para o próprio sistema de armazenamento. Na seção seguinte analisar-se-á mais detalhadamente cada parte deste sistema de forma a encontrar possíveis ameaças subjacentes a essas boundaries e elementos.

Nas duas seguintes imagens encontram-se generalizados os processos executados pelos dois tipos de utilizadores do nosso sistema. Apesar de se encontrar representado o acesso aos dados pelo Farmer, é esperado que o Expert também os requisite. No entanto, só o Expert é que poderá alterar os dados da base dados para melhorar o funcionamento do sistema.

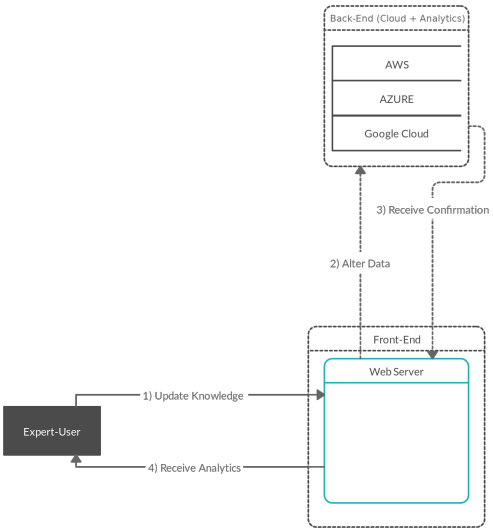

Figura 2. Uploading Data by Expert

No subsistema da quinta identificam-se 3 processos nos quais o d̈ata flowp̈ode sofrer ataques devido a vulnerabilidades. Na figura 4 podemos ver os "actuators"que controlam a temperatura/humidade das zonas da quinta. As basestations que coletam a informação recolhida pelos sensores e a Cloud para a qual enviam a informação recolhida e da qual recebem instruções.

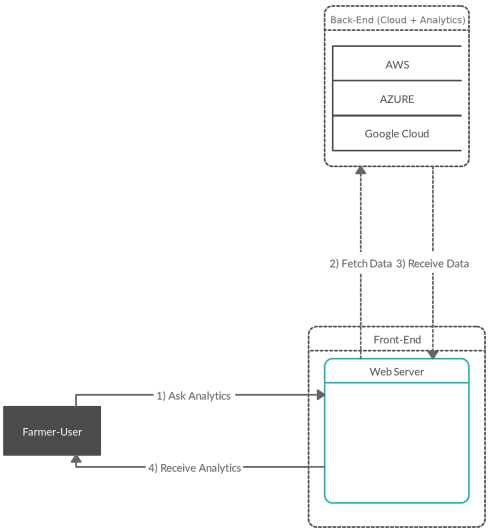

Figura 3. Acessing Data by Farmer

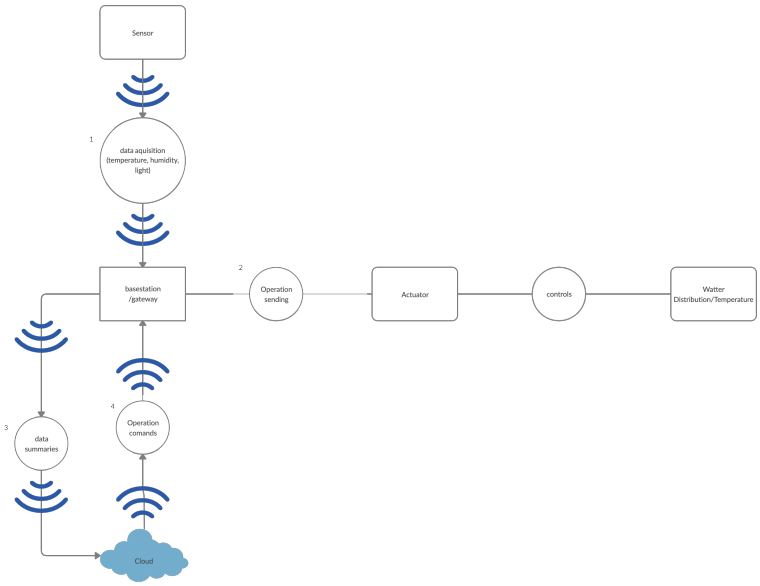

Figura 4. Model System for the Farm Subsystem

2 Finding & Addressing Threats

De forma a encontrar as ameaças referentes ao sistema ir-se-á analisar cada elemento seguindo STRIDE.Cada subseção corresponde a um dos três subsistemas definidos anteriormente.

2.1 FrontEnd

Spoofing

- Um atacante poderia conetar ao sistema em 1) e 3) em na figura 2 ou em 3 caso este nao tenha nenhum sistema de autentificação tanto para os utilizadores do Front-end como para o Front-end em si ao conetar-se ao Back-End. Para mitigar esta ameaça é necessário tanto um sistema de autentificação para os utilizadores (login) como um sistema para gerenciar as sessões do mesmo(Encrypted cookies,Session Keys). Enquanto na ligação do Front-End ao Back-End é também necessário alguma forma de Autentificação(Tickets Kerberos,IPsec). Assim, só se permitiria aos servidores Front-End autorizados aceder a API do Back-End e solicitar dados do mesmo.

- Um atacante em 1) poderia tentar fazer login na aplicaçao front-end como farmer/Expert através de tentativas sucessivas de pares user/password. De forma a mitigar o problema poder-se-ia implementar um delay quando o sistema nota muitas tentativas de login. Esta medida tem consequências visto que pode ser prejudicial ao sistema. Mesmo que o atacante não conseguisse descobrir os pares login/password, caso este processo de tentativa de adivinhação fosse realizado inumeras vezes num curto espaço de tempo, o mesmo ataque passaria a um DoS impedindo que os utilizadores do Front-End se conseguissem conectar ao sistema. Possivelmente uma Firewall ajudaria a mitigar esta ameaça.

- Um atacante poderia criar um "clone"do web-server de forma a enganar o cliente para inserir os dados de login. Assim obtendo acesso ao sistema ao nivel do utilizador. A mitigação deste ataque é possível através do uso de credenciais no Front-End que serão guardadas pelo browser do utilizador.

- Um atacante irá sempre utilizar o método com a segurança mais fraca. Consequentemente, dever-se-á obrigar sempre a utilização das ligações com protocolos que possuam encriptaçao(HTTPS/IPsec).

Tampering

- Um atacante poderia alterar dados referentes ao sensores no web server na figura 2, de forma a enganar o cliente sobre o estado dos sensores e da quinta. Utilizando um mecanismo de encriptação nos dados este ataque pode ser mitigado.

Repudiation

- Um utilizador do tipo Expert poderia em 1) na figura 3 mandar um pedido para alterar a knowledge do sistema de forma a piorá-lo e não admitir essa alteraçao. Para evitar esta situação poderia manter Logs de alterações feito pelos Experts no armazenamento na Cloud.

Information Disclosure

- Admitindo que a aplicação pode dar suporte a diferentes farms. Um utilizador ja ligado ao sistema pertencente á concorrência poderia pedir acesso a ficheiros dos concorrentes. Uma solução fácil seria implementar ACLs para restringir acesso aos ficheiros ou na lógica da aplicação WEB do Front-End.

- Outros problemas como man-in-middle, acesso ao ficheiro não encriptado,etc. Sao resolvidos com alguns do métodos referidos anteriormente.

Denial of Service

- Como referido na seção de Spoofing no primeiro ponto, poderia causar-se DoS usando o delay para prevenir tentativas de brute force das passwords.

- Um atacante depois de entrar no sistema Front-End como Expert pode tentar atualizar o sistema através de vários PCs com login na mesma conta. Provocando uma incapacidade do Servidor para que este não conseguisse satisfazer todos os pedidos. O mesmo poderia ser feito caso fosse feito login como Farmer através de varios GETs. O Profiling do tráfego da rede como anomalias e reputações de IP mitigariam o ataque.

Elevation of Privilege

- Um atacante não ligado ao sistema pode tentar utilizar a API para aceder ao Back-End.Ganhando assim os mesmos privilegios que o servidor Front-End.Uma lista com os servidores/utilizadores que podem aceder diretamente a API podia mitigar este ataque.

2.2 Farm Model System

Spoofing

Um atacante pode fazer-se passar por um sensor e transmitir dados falsos para o gateway 1), o mesmo pode acontecer se o atacante tentar fazer-se passar pelo gateway e enviar dados para a cloud 3) ou caso se faça passar pela cloud e envie dados errados ao gateway 4). Este ataques pode ser facilmente evitado usando mecanismos que permitam identificar inequivocamente cada elemento do sistema.

Tampering

Um atacante pode alterar informação dos dados recolhidos pelo gateway através dos sensores. Para evitar que este cenário se verifique basta ter ACLs que limitem de forma correta as atividades dos utilizadores.

Repudiation

Um atacante pode alterar dados armazanados no gateway e apagar o log para que não se saiba quem os alterou. Para garantir que se salvaguarda o log basta guardar os registos de acesso ao gateway numa máquina á parte na qual se mantenha uma cópia do log e só se permita escrever, como na cloud por exemplo.

Information Disclosure

Caso a informação não seja encriptada, esta pode ser lida por um concorrente que não pague o serviço dispendioso e que possa obter vantagem competitiva simplesmente ao intercetar as informações enviadas pelos sensores 1). O mesmo pode acontecer quando a informação é enviada pelo gateway á cloud processo 3) e pela cloud ao gateway 4). Esta falha pode ser colmatada fazendo uso de encriptação com chave simétrica conhecida apenas entre os dispositivos do sistema, ou usando mecanismos de chave publica.

Denial of Service

Um atacante pode tentar levar a cabo um ataque DoS nos processos 1), 3) e 4), gerando um monte de pedidos aos quais o sistema não consiga responder. No caso de tentativa de Denial of Service podemos programar os nosso equipamentos para, após um X número de tentativas, bloquear o IP (temporáriamente) de quem está a gastar recursos do sistema sem se autenticar.

Nota: No caso da Cloud, está a cargo do provedor do serviço tratar casos de tentativas de Denial of Service.

Nota: Em relação ao processo 2), como a ligação é feita usualmente por cabo não existem problemas muito serios, apenas ataques fisicos devem ser considerados, como cortar os cabos de ligação para impedir o funcionamento do sistema.

3 Considerações Finais

3.1 Front-End

A segurança necessária para assegurar o subsistema do Front-End baseia-se fortemente na encriptação da informaçao que é enviada e recebida pelos utilizadores/servidor(HTTPS,IPsec) como também pela utilização de Assinaturas e MACs para garantir a integridade. Finalmente também dever-se-á utilizar ACLs e Profiling de Tráfego para previnir acesso indevidos e DoS ,respetivamente.

3.2 Sensores

Como visto anteriormente os sensores devem ser autenticado inequivicamente e enviar a informação recolhida encriptada para garantir a veracidade e origem dos dados.

Para garantir que um atacante não adiciona um sensor com código modificado de forma a de alguma forma comprometer o sistema, os sensores a poderem comunicar com o gateway devem ser definidos á partida.

ZigBee sensors

- Quanto á identificação dos mesmos esta é feita de forma inequivoca, garantindo que não é possível um atacante replicar o id do sensor [1].
- Usa AES-128 encryption para garantir a encriptação dos dados [1].

TelosB motes

- Usa TinyOS como a biblioteca criptográfica normalmente, esta tem um algoritmo eficiente com overhead de encriptação na ordem dos microsegundos, garantindo a integridade do sistema de forma rápida e utilizando poucos recursos [2].
- Relativamente barato [2].

Arduino

- Normalmente usa AES-128 encryption para garantir a encriptação dos dados [3] [4].

  Usa X.509 que é um padrão ITU-T para infraestruturas de chaves públicas (ICP),permitindo autenticação forte [5] [6].
- Permite programar os processos de autenticação dos sensores e como estes comunicam.

Raspberry

- Usa vários mecanismos para garantir a encriptação, como RC4 encryption, entre outros. [7].
- Relativamente barato [7].

Concluindo, se o foco for ter uma encriptação o mais forte possível, devemos optar por utilizar um raspberry sensor visto este usar mecanismos de chave pública. No entanto estes pesam no requisito de transmissão de dados de forma mais rápida possível. O TelosB motes tem essa vantagem embora não possua uma encriptação tão boa como os restantes sensores.

Nota: Se se pretende uma tecnologia de encriptação que se mantenha segura sem ser necessário mudá-la, o AES-128 não é aconselhado visto que já está em fase de se tornar deprecated (estima-se que por volta de 2030) e a quinta supõe-se que deva permanecer em funcionamento por mais de 10/11 anos aquando da redação deste relatório.

3.3 Gateway

No caso da informação que flui do gateway para a cloud é preciso garantir que estamos a enviar dados encriptados visto que o provedor da cloud pode utilizar os dados que registamos para proveito próprio, ou até mesmo um terceiro que comprometa a segurança da cloud não possa tirar proveito da nossa informação. Desta forma temos de garantir que apenas o nosso gateway e a aplicação no lado do cliente tenha acesso á chave de encriptação e desencriptação. Note-se que se deve fazer uso de encriptação forte para garantir de forma mais consistente a segurança dos dados.

3.4 Cloud

O tratamento da segurança da informação na cloud (encriptada ou não) é responsabilidade do provedor do serviço, logo este relatório não englobará medidas para proteger a cloud de qualquer ataque. Apenas de proteger a informação que lhe é passada encriptando-a.
Para além disso é necessário ainda assim que tenhamos uma forma inequivoca de garantir o não repúdio por parte do gateway face a cloud e vice-versa, nesta matéria aconselha-se o uso de um certificado digital.

Referências

1. https://www.blackhat.com/docs/us-15/materials/us-15-Zillner-ZigBee-Exploited-The-Good-The-Bad-And-The-Ugly-wp.pdf
2. https://www.researchgate.net/publication/323999621_Performance_evaluation_on_TelosB_mote_of_a_secure_
3. https://www.dfrobot.com/blog-911.html
4. https://everythingesp.com/esp32-arduino-tutorial-encryption-aes128-in-ecb-mode/
5. https://www.arduino.cc/en/IoT/HomePage
6. https://docs.aws.amazon.com/pt_br/iot/latest/developerguide/x509-certs.html
7. https://ieeexplore.ieee.org/document/6782081