Universidade do Minho
Escola de Engenharia

TECNOLOGIA DE SEGURANÇA

TP1 – Parte B

Threat Modelling

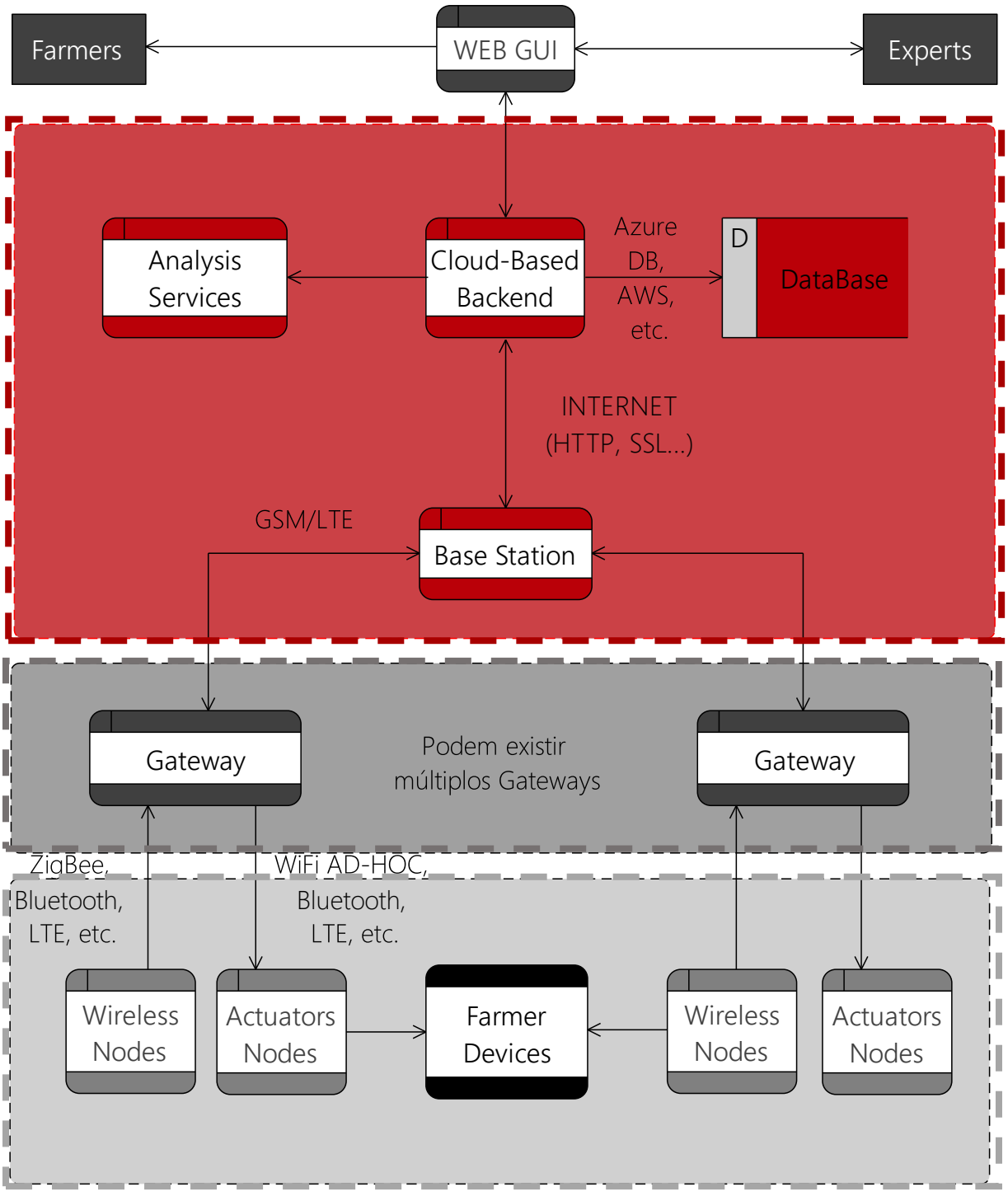
Diogo Araújo A78485; Diogo Nogueira A78957

Conteúdo

1. Contextualização do Modelo de Sistema 3
2. Modelação de Ameaças 5
2.1. Ameaças de Segurança do Sistema 6
2.1.1. Nodos de atuadores e sensores sem fios (*WSN*) 6
2.1.2. *Gateway/ Basestation* 8
2.1.3. *Backend* baseado em *Cloud* 10
2.1.4. *Dashboard/GUI* 11
2.2. Apanhado Geral das Ameaças 15
3. Conclusões e Observações Finais 16
4. Referências 17

1. Contextualização do Modelo de Sistema

O objetivo deste trabalho prático consiste na criação dum modelo de ameaça para o *Precision Agriculture System* fornecido no enunciado. Este sistema foca-se na utilização de tecnologia e princípios científicos para analisar e gerir toda uma plantação abordando a variação espacial e temporal do ambiente onde se associará com todos os aspetos de agricultura em (quase) tempo real.



Esta plataforma incide principalmente nos quatro componentes principais:

1. Nodos de atuadores e sensores sem fios (*WSN*): Este tipo de dispositivos são a base da plataforma e de toda a sua algoritmia, que através da aquisição de dados como, por exemplo, a temperatura, humidade ou luz fornecem dados para o sistema de análise. Para a conexão em rede usa-se sensores com protocolo *ZigBee* ou mesmo dispositivos *Arduino* e/ou *Raspberry* que podem comunicar por Bluetooth ou por outro protocolo *PAN* (*Personal Area Network*) a definir. Por fim, os atuadores são também dispositivos *wireless* que modificam o estado de diversos dispositivos da plantação, como por exemplo o termóstato duma estufa ou a quantidade de água no sistema de rega.

2. *Gateway/Basestation*: Com várias interfaces de comunicação rádio para transmitir e receber informação dos *WSN* e uma interface de comunicação móvel como GSM/LTE para a ligação à *Internet*, este dispositivo serve para manipular os *WSN*s ajustando-os consoante as estatísticas feitas por um *backend* numa *cloud* como será explicitado mais à frente. Estes *gateways* recebem a informação através de qualquer protocolo disponível e em tempo real sendo que agrega a mesma numa memória "passageira" até enviar para a *cloud* os sumários e agregações que obteve.

3. *Backend* baseado em *cloud*: Este sistema inclui um armazenamento na *cloud* e que uma instância de *software* pode correr para vários tipos de utilizadores. Utilizando uma solução de computação em *cloud* como *Microsoft Azure* ou *Amazon AWS*, terá também um módulo de ciência de análise e analítica que receberá as agregações de dados vindas dos *gateways* e analisará esses dados da plantação. Assim dessa forma pode enviar as novas regras e comandos a serem executados pelos *WSNs*. Também fornece APIs abertas para controlo, modificação e acesso ao serviço, tanto pelos agricultores como pelos especialistas.

4. *Dashboard/GUI:* O módulo *frontend* baseado em *web* é utilizado para a visualização através de computadores pessoais, *tablets* e *smartphones*. Tem dois modos no seu núcleo de funcionamento sendo um para os agricultores que demonstra todo o histórico de dados colecionados e o *business intelligence* por detrás das decisões a fazer e melhorar na plantação. O outro modo é para os especialistas que melhoram o base de conhecimento do sistema baseando-se no seu próprio conhecimento do ofício onde trabalham há anos.

2. Modelação de Ameaças

A modelação de ameaças é um processo de análise complexo que visa em procurar o que pode estar de errado no sistema que estamos a construir e idealizar de raiz. Ao criar um conjunto de atacantes ideais e, dessa forma, formar pareceres sobre como e onde aconteceria a ameaça ao sistema torna-se imperativo para uma construção do sistema sempre em volta da segurança do mesmo, tornando-o assim orientado sempre à manutenção da sua firmeza à ataques externos ou internos. Assim, um bom modelo de ameaças ajuda a travar e prevenir ameaças e também fornecer um produto final seguro.

Através da abstração das ameaças e colocando-as num sistema de classes definido pela Microsoft, intitulado de STRIDE, podemos assim ajudar melhor nos riscos e *standards* comuns de *exploits* usados pelos atacantes. STRIDE é um acrónimo para *Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege*.

De forma a explicar de forma sintetizada, segue-se os seguintes pontos:

- *Spoofing*: Fazer-se passar por alguém ou por algo que não o verdadeiro, violando a autenticação correta.
- *Tampering*: Modificar dados num disco, rede ou memória não persistente, violando assim a integridade do sistema.
- *Repudiation*: O ato de conseguir recusar a confirmação sobre a autoria de algo que aconteceu, tal como eliminar *logs* do sistema.
- *Information Disclosure*: Lançar informação confidencial para uma entidade que não está autorizada a ter acesso à mesma, violando assim a confidencialidade dos dados.
- *Denial of Service (DoS)*: Absorver e esgotar recursos necessários para providenciar o serviço, violando assim a disponibilidade do sistema.
- *Elevation of Privilege (EoP):* Permitir que uma entidade consiga fazer algo que que não está autorizada a fazer, violando assim os protocolos de autorização e hierarquia.

2.1. Ameaças de Segurança do Sistema

2.1.1. Nodos de atuadores e sensores sem fios (*WSM*)

Este tipo de dispositivos são a base da plataforma e de toda a sua algoritmia, que através da aquisição de dados como, por exemplo, a temperatura, humidade ou luz fornecem dados para o sistema de análise. Por tal, deve ser modelado as ameaças possíveis a acontecer de forma a não existir uma brecha de segurança logo no ponto mais baixo do sistema.

Numa plantação que utilize este sistema a ser desenvolvido existe várias trocas de dados a ocorrer nestes *WSNs*. Através de protocolos heterogéneos de rede e comunicação tal como *ZigBee*, *Bluetooth*, *Thread* ou mesmo qualquer outro protocolo que possa envolver uma rede com funcionalidade de *mesh*, ligando assim os múltiplos sensores e atuadores espalhados pela plantação.

- ***Spoofing***

Uma rede em *mesh* pode acontecer a dar uma identidade a um dispositivo externo, similar aos que estão a funcionar na rede e se introduzir na mesma, sem nunca existir uma confirmação se estou a falar com os dispositivos certos e quais são em si. Dessa forma, torna-se imperativo fornecer identidades a cada dispositivo e autenticar as suas informações antes de as receber, utilizando um protocolo como *Transport Layer Security (TLS)* ou *IPSec* na rede *mesh*, colocando assim uma criptografia para autenticar os *WSNs* do sistema.

- ***Tampering***

Numa rede de dispositivos *wireless* espalhados por uma plantação pode acontecer que um atacante parcial ou totalmente substitua o software a correr nestes atuadores e sensores conseguindo assim substituir os dados coletados nos sensores e substituir por dados falsos. Isto pode acontecer se não existir no próprio dispositivo um mecanismo que seja à prova de *tampering*, fazendo assim impossível a extração das chaves que pudessem estar a proteger o *software* e o dispositivo em si. Desde a encriptação da memória do dispositivo, à existência de certificados e chaves que

confirmem o *software* a correr e principalmente a existência do *trusted platform module (TPM)* que consegue guardar chaves e certificados em que os mesmos nunca conseguem ser lidos, mas sim utilizados para a confirmação e operação criptográfica para a proteção do dispositivo. Este *tampering* inicial iria ser catastrófico dado que por consequência poderia afetar com *Repudiation* e *Information Disclosure*, dado que era um acesso quase total também às informações confidenciais nos sensores e os seus possíveis *logs*.

- ***Information Disclosure***

Estes sensores com os seus vários protocolos de comunicação heterogéneos, tanto na comunicação entre eles em modo *mesh*, como na ligação ao *field gateway* pode ser lido através da infiltração da rede, ainda mais se a informação nela não estiver encriptada. Através do redireccionamento do tráfego para um dispositivo externo para ler os dados, pode-se obter segredos cruciais ou apenas entender como funciona a estrutura e "quem fala com quem".

Outro caso comum é caso exista a *Information Disclosure* após um *Tampering* dos dispositivos *WSN*, que o atacante ao ter acesso a material criptográfico utilizado como chaves para encriptar e desencriptar o *data flow*, pode injetar-se no caminho de comunicação entre os dispositivos e obter informação que se pensava segura dado que estava encriptada. Uma maneira de mitigar esta situação e poder existir um "*man-in-the-middle*" que não consiga obter nenhuma informação relevante é utilizar criptografia assimétrica sendo que todos os *WSNs* têm uma chave pública para encriptar os dados a captar, mas apenas a *basestation* e/ou a *cloud backend* contêm a chave privada para desencriptar os dados dos sensores. Assim poderia existir leitura do *data flow* mas nunca obtendo informações verdadeiras e cruciais.

- ***Denial of Service (DoS)***

Toda a ligação dos sensores e atuadores é feita através de protocolos associados ao *Internet of Things*, tal como *Bluetooth* em modo *mesh*, Wi-Fi e *ZigBee*. Estes dispositivos podem ser colocados irrelevantes ou incapacitados de funcionar ou comunicar caso ocorra uma interferência das radiofrequências das bandas 2.4GHz que são usadas de modo universal para as conexões efetuadas por *Bluetooth*, *Wi-Fi* e *ZigBee*. Fazendo um ataque DoS usando estas frequências faria com que os *WSNs*

não pudessem comunicar entre si quebrando assim a rede em *mesh* e perdendo dados a ser recolhidos dos sensores ou falhando nas ordens dadas aos atuadores para reconfigurar controlos na plantação.

2.1.2. *Gateway/ Basestation*

O principal objetivo destes dispositivos é organizar o tráfego de informações recebidas através dos *WSNs* e a *Internet* em si, enquanto os ajusta consoante as estatísticas feitas pelo *backend*.

Através do DTD fica percetível a ideia de como o fluxo de informação/instruções acontece. A ideia mais base que importa compreender para este estudo é que os *Wireless Nodes* "absorvem" dados essenciais para a plantação, e os *Actuators Nodes* agem de acordo com o que é necessário fazer, comandando assim os *devices* todos da plantação.

- ***Tampering***

A *Network Tampering* envolve normalmente uma variedade de truques que consistem em trazer os dados que estão a circular na rede para a máquina do atacante, onde este pode depois enviar outros dados intactos ou até mesmo modificados. A verdade é que nem sempre é preciso grande "mistério" para se conseguir extrair essa informação que está a circular na rede, devido ao facto de algumas *radio* interfaces que podem ser usadas no nosso sistema serem já propícias a isso (como por exemplo o próprio *WiFi* e *Bluetooth*). Sabe-se que no nosso sistema de agricultura, circulam dados cruciais desde os *WSNs* e os *gateways* em si, e é precisamente neste *data flow* que é viável acontecer uma violação à integridade do sistema.

A primeira e principal ameaça é a possibilidade de o atacante poder redirecionar o tráfego de dados para a sua própria máquina. Ao fazê-lo, acaba por executar o passo principal no que toca ao *tampering* através da rede, já que pode posteriormente alterar os pacotes extraídos.

Com o tráfego redirecionado diretamente para a sua máquina, o atacante pode efetuar modificações sobre os pacotes realizando aquilo a que se chama de *Packet*

Tampering. Esta possibilidade abre também portas para a extração do conteúdo bruto dos mesmos, representando uma ameaça em termos de divulgação de informação – que será explorada no tópico seguinte.

No geral, este tipo de ataque representa um grande perigo no que diz respeito ao funcionamento dos *Farmers Devices*. Ao conseguir alterar o tráfego que é transmitido do *gateway* para os *actuators nodes*, podemos estar a modificar o estado do funcionamento dos diversos dispositivos da plantação de forma errada e não baseada nas estatísticas informadas pela *cloud*. A mesma falsificação pode ocorrer no envio de *status* dos *Wireless Nodes* para os *gateways*.

- ***Information Disclosure***

Se o dispositivo estiver a executar um *software* modificado, então esse software pode potencialmente originar fugas para partes que não são autorizadas. A ideia de existir um ataque de *tampering*, pode automaticamente originar a divulgação de informações secretas. Ao redirecionar o tráfego que vai dos *actuators nodes* para os *gateways*, o atacante pode conseguir efetuar uma leitura de dados na rede e até mesmo descobrir certos segredos pelo simples facto da análise do tráfego. O mesmo pensamento pode acontecer relativamente aos dados que circulam desde o *gateway* até aos *wireless nodes*.

Algo que pode também acontecer é a descoberta das entidades que estão a manter comunicação por meio de *DNS*, o que por si só pode também representar um perigo em termos de segurança de todo o sistema.

- ***Denial of Service (DoS)***

É possível incapacitar o funcionamento ou as comunicações destes dispositivo só pelo simples facto de se interferir com as frequências rádio que existem para possibilitar a comunicação dos *gateways* com os sensores/atuadores.

Dado que os *gateways* são como um intermediário entre a informação que circula da *cloud* aos *devices* da plantação (e vice-versa), o atacante pode conseguir consumir os recursos da rede, tornando assim todo o sistema indisponível.

2.1.3. *Backend* baseado em *Cloud*

Este sistema inclui um armazenamento na *cloud* e que uma instância de software pode correr para vários tipos de utilizadores. Utilizando uma solução de computação em *cloud* como *Microsoft Azure* ou *Amazon AWS*, terá também um módulo de ciência de análise e analítica que receberá as agregações de dados vindas dos *gateways* e analisará esses dados da plantação. Assim dessa forma pode enviar as novas regras e comandos a serem executados pelos *WSNs*. Também fornece *APIs* abertas para controlo, modificação e acesso ao serviço, tanto pelos agricultores como pelos especialistas.

O modelo de segurança a fazer para esta parte é complicada por si só dado que para acontecer qualquer tipo de ameaça a esta parte do sistema tem de existir primeiro uma quebra na autenticação do utilizador/administrador do sistema que tem acesso ao *hub* da *cloud* arrendada e todos os serviços que ela possui. Isto porque um acesso físico aos servidores, torna-se impossível dado a escala de segurança e anonimato das empresas fornecedoras como a Microsoft ou Amazon.

- ***STRIDE***

Consoante o tipo de *login* utilizado para ter acesso à *hub* do *back-end*, pode acontecer um *phishing* comum, chamado de *Web Spoofing*, que finge ser a página de login do administrador e assim obter as informações inseridas pelo mesmo em espécies de formulários, tal como as suas credenciais de acesso. É possível mitigar este tipo de problemas utilizando um método de *login* por autenticação por dois fatores ou mesmo sem a utilização de palavras-passe em favor de outros métodos de autenticação com dados biométricos como impressão digital.

Caso isso não esteja implementado, o que pode acontecer é um acesso total ao núcleo do sistema e por consequência existir quase todos os tipos de ameaças à segurança do mesmo.

Começando pelo *Elevation of Privilege* que acontece automaticamente dado que o atacante conseguiu a habilidade de obter privilégios que não os teria e conseguir manipular o *hub* da *cloud* para enviar informações que são certificadas para a plantação alterando as verdadeiras ações que os atuadores deveriam estar a fazer. Com este acesso total ao sistema da *cloud* o atacante pode negar ou estragar o serviço para todos os seus utilizadores e entidades através do consumo do sistema para outras

coisas, tal como absorver a RAM ou CPU da máquina virtual/instância ou até encher a base de dados de dados irrelevantes tornando o sistema incapaz de receber e fornecer a base do conhecimento para a plantação, começando aqui um *Denial of Service*. A visualização de informações confidenciais presentes na base de dados central faz com que seja realmente catastrófico principalmente se se tiver acesso às chaves criptográficas privadas presentes no disco e/ou memória. Desta forma, pode existir um *Information Disclosure* a nível global dado que a (pouca) criptografia existente no sistema agora poderia ser revertida obtendo a base de conhecimento e até mesmo o método de autenticação dos vários intervenientes do sistema.

Com esta abertura na falha de segurança pode acontecer uma falha ao nível de *Repudiation*, dado que não só pode ser possível eliminar todos os *logs* usados para armazenar cada ação feita no sistema como também forjar a ação dizendo que outro utilizador a fez.

Por fim, é possível existir *Tampering* com facilidade dado que todos os caminhos de dados, processos e de rede vão dar à *cloud* e dessa forma existe a capacidade de alterar ficheiros presentes no sistema, levando em erro todo o sistema de *analytics* forjando assim a veracidade das decisões.

2.1.4. *Dashboard/GUI*

A *GUI* trata-se de um modelo de interface para o utilizador que permite a interação com os diversos dispositivos digitais através dos elementos gráficos do sistema. Neste caso, a *GUI* apresenta/oferece dois modos – um para os agricultores, que apenas receberão o histórico dos dados coletados e processados pelo próprio serviço de armazenamento, e outro para os especialistas, que para além de receberem as informações, podem interagir com a própria interface, melhorando de forma contínua a base de conhecimento do sistema de agricultura.

Dado que se trata de um módulo *frontend*, facilmente se entende que é algo que está diretamente exposto à vista/manipulação do utilizador. Essa ideia pode desde logo nos levar a assumir que se trata de uma componente excessivamente vulnerável a ameaças e que por isso deve ser devidamente validada/protegida. A verdade é que isso pode acontecer, mas não implica que os problemas possam efetivamente acontecer por meio do *frontend*.

O importante a reter com toda esta informação é que a *GUI* acaba por simbolizar uma espécie de "porta" para o atacante e que mesmo não sendo responsável por garantir a segurança dos dados do sistema em si, não deixam de existir contramedidas de segurança que podem proteger a *GUI* de ataques a que está indiretamente exposta.

- ***Spoofing***

Em termos de *Spoofing*, podemos assumir que este tipo de ataque pode acontecer. Temos o exemplo mais base de *WEB Spoofing*, que se trata de um ataque de segurança que permite que um atacante observe e modifique as páginas *WEB* enviadas para a máquina da vítima e com isso obtenha as informações inseridas por esta entidade em espécies de formulários. É importante ter em conta que este tipo de ataque não é impedido pelas ditas conexões seguras – o tal cadeado à esquerda do URL e que por ser um ataque facilmente implementável pode representar um problema para o sistema de agricultura.

Basta pensar na *GUI* oferecida para os *experts*, que têm o poder de atualizar/instruir a base de conhecimento do sistema. Supondo que esta *GUI* consiste num sistema de login para validar a entidade que está a entrar no sistema, se existiu *Spoofing*, estamos a dar permissão a alguém que não está autorizado para tal e com isto pode-se estar a comprometer o sistema com informações falsas (dado que existe uma ligação direta entre a *GUI* e a *backend* baseada em *cloud*).

Esta possibilidade de se fazer passar por um *expert* pode prejudicar o funcionamento dos *devices* da plantação e induzir todo o sistema de Inteligência Artificial ao erro. O mesmo pensamento pode ser feito para falsos *farmers*, que podem estar a obter informações secretas sobre as plantações, ideias, planos e etc...

Dado que este ataque não pode ser interrompido no *frontend*, é essencial que exista uma proteção no *backend*, mais especificamente na proteção dos dados que entram e que depois são validados pela *cloud* em si (como o tipo de dados e o seu formato).

- ***Information Disclosure***

No que toca à divulgação de informação, a *WEB* representa um meio propício para que se obtenham informações confidenciais ou alheias ao utilizador que está a visitar a página, violando com isso toda a confidencialidade dos dados. Se assumirmos que pode existir *Spoofing*, então podemos da mesma forma assumir que pode existir divulgação de informações secretas.

Apesar desta associação entre o *Spoofing* e *Information Disclosure* ser perfeitamente trivial, a verdade é que podem existir outras formas de entidades não autorizadas obterem informações que não lhes é suposto ser dadas.

Um exemplo muito simples é a divulgação errada de *source-code* vindo do ambiente *backend*. Este tipo de divulgação pode permitir que a entidade entenda o comportamento do sistema e consiga com isso obter informação secreta.

Não só pelo *source-code*, muitas aplicações *WEB* podem divulgar nomes/caminhos de arquivos, revelando com isso informações acerca da estrutura do sistema subjacente. Isto pode acontecer devido ao tratamento incorreto na entrada do utilizador, exceções no *backend* ou até mesmo devido a uma configuração inadequada do servidor *WEB*. O que acontece é que muitas das vezes estes tipos de informações secretas podem ser encontradas/identificadas nas respostas por parte das aplicações *WEB* tal como pelas páginas de erro.

Algo muito intrínseco nas páginas *WEB* é a *path disclosure*, que se trata da divulgação da lista de diretorias existentes no servidor *WEB* em questão. Quando não existe uma determinada página *WEB* padrão a mostrar, o servidor acaba por listar os arquivos e diretorias existentes para o *website*. Esta é uma funcionalidade que pode ser facilmente desativada, não deixando de ser importante referir a possibilidade de acontecer.

O mais importante a reter com tudo isto é que existem várias formas de uma entidade observar informações secretas através da interface *GUI* a que tem efetivamente acesso. Tudo dependerá de como a interface em si será desenvolvida – o tipo de informação que é de facto divulgada para ambas as entidades e que métodos de segurança serão implementados nas suas *GUIs*.

Muitas das vezes, uma autenticação baseada em dois fatores (*login* inicial e envio de um SMS sempre que se quer confirmar algo importante) pode ajudar em muito a que exista uma certeza na divulgação dos dados. No caso do nosso sistema, isso poderia se tornar bastante eficaz, dado que os *farmers* recebem um histórico de dados da plantação e que os *experts* podem ainda atualizar a base de conhecimento e que

dessa forma seriam necessário ambas as entidades confirmarem a sua identidade para conseguirem aceder ao sistema propriamente dito.

- ***Denial of Service (DoS)***

As aplicações *WEB* são particularmente suscetíveis a ataques do tipo *Denial of Service*, já que não é assim tão percetível a diferença entre um ataque e um tráfego comum de dados. Uma vez que os endereços IP não são úteis como "credenciais" para identificação no *website* e tendo em conta que não existe uma forma confiável de saber de onde vem uma solicitação HTTP, torna-se muito difícil filtrar o tráfego prejudicial ao servidor *WEB*, acabando assim por existir uma sobrecarga de tráfego que põe em causa toda a disponibilidade do sistema.

Vamos pensar no hipotético caso deste ser aberto a utilizadores de forma não autenticada. Se estes utilizadores conseguirem solicitar tráfego na *request message*, pode existir uma consulta excessiva à base de dados em si por cada HTTP *request* recebido, acabando por sobrecarregar o servidor *WEB* e tornando-o indisponível para futuros utilizadores legítimos. No entanto, como se espera que o sistema seja apenas aberto a *farmers* e *experts* e que ambos efetuem um *login* para eventual consulta/envio de informação, consegue-se contornar toda esta a saturação de tráfego.

Este pensamento pode não ser assim tão verdadeiro, atendendo ao facto de que o controle de utilizadores nem sempre garante que o tráfego esteja devidamente controlado. Basta existir um *Spoofing* para se tornar possível uma sobrecarga no tráfego por parte de um utilizador falso.

A indisponibilidade do sistema pode ser total, mas também pode resultar na privação dos recursos para um utilizador em específico. O exemplo mais simples desse tipo de ocorrência é a tentativa de um atacante bloquear a conta de um determinado utilizador, simplesmente pelo facto de "enganar" o sistema com *passwords* falsas até existir a suspensão da conta.

Em muitos dos casos, é difícil existir uma defesa perfeita contra estes ataques, não deixando de ser possível dificultar o êxito deles. A ideia pode passar também por limitar recursos para parte das entidades, limitando a quantidade de carga que cada uma destas pode efetuar num determinado intervalo de tempo.

2.2. Apanhado Geral das Ameaças

Threat	WSN	*Gateway/Basestation*	*Backend* baseado em *cloud*	*GUI*
Spoofing	X	-	X	X
Tampering	X	X	X	-
Repudiation	-	-	X	-
Information Disclosure	X	X	X	X
Denial of Service (DoS)	X	X	X	X
Elevation of Privilege	-	-	X	-

3. Conclusões e Observações Finais

O processo de modelação de ameaças consiste num processo que visa a descoberta daquilo que pode estar/dar errado em cada sistema. Este processo existe essencialmente por questões de segurança, tendo em conta que é sempre bom descobrir possíveis ameaças para que depois seja mais fácil de corrigi-las. Com isso, consegue-se produzir um bom modelo através de um conjunto de classes e grupos de ataque, entregando um produto final mais preciso e seguro.

No nosso *Precision Agriculture System*, essa modelação foi baseada segundo uma descrição detalhada dos seus componentes principais e as tecnologias em si envolvidas. Com a ideia de como o sistema funciona, as entidades envolvidas e os processos intervenientes, foi possível desenvolver um *DFD (Data Flow Diagram)*, que por decisão unanime do grupo, baseou-se no funcionamento geral do sistema. A ideia de construir apenas um diagrama geral do sistema deveu-se ao facto de ser o suficiente para definir as *trust boundaries* e com isso compreender as ameaças que poderiam existir ao longo deste percurso hierárquico deste a *GUI* apresentada aos *farmers* e *experts*, até aos *devices* localizados nas instalações da quinta. Com este diagrama, desenvolveu-se o modelo de ameaças STRIDE, de forma a ajudar a raciocinar e encontrar as tais ameaças do sistema levando à resposta final daquilo que pode efetivamente dar errado.

A maior dificuldade nesta segunda parte do primeiro trabalho prático foi perceber a que tipo de ameaças cada componente estava exposto. A maioria das vezes ambos ficávamos com a ideia de que todas as ameaças eram prováveis acontecer, mas na verdade, após um melhor domínio sobre o que cada componente fazia e de que forma a hierarquia do sistema podia influenciar os processos/entidades acima e abaixo de cada um destes, o grupo acabou por eliminar muitas dúvidas.

Com uma boa estratégia definida, o grupo discutiu e chegou recorreu ao jogo de cartas disponibilizado pela Microsoft, acabando por explicar a sua própria ideia do que poderia efetivamente acontecer para o sistema de agricultura em causa.

4. Referências

Application Denial of Service. (22 de Abril de 2010). Obtido de OWASP: https://www.owasp.org/index.php/Application_Denial_of_Service

Azure, M. (2018, Outubro 09). *Internet of Things (IoT) security architecture*. Retrieved from Microsoft Azure: https://docs.microsoft.com/pt-pt/azure/iot-fundamentals/iot-security-architecture

Donohue, B. (26 de Março de 2014). *GUI Vulnerabilities Expose Information Disclosure, Privilege Escalation*. Obtido de threat post: https://threatpost.com/gui-vulnerabilities-expose-information-disclosure-privilege-escalation/105039/

Tavares, P. (17 de Maio de 2014). *https://seguranca-informatica.pt/conhecendo-e-explorando-o-bug-heartbleed/#.XZoIPUZKg2w*. Obtido de Conhecendo e Explorando o HeartBleed.

Team, N. S. (19 de Junho de 2019). *Information Disclosure Issues and Attacks in Web Applications*. Obtido de netsparker: https://www.netsparker.com/blog/web-security/information-disclosure-issues-attacks/