Escola de Engenharia
Universidade do Minho

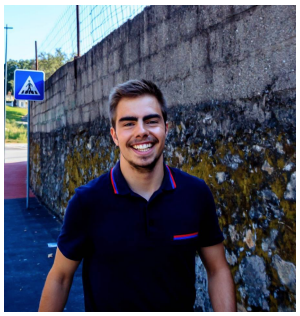
DEPARTAMENTO DE ENGENHARIA INFORMÁTICA
Mestrado em Engenharia Informática
Engenharia de Segurança

Modelagem Tática de Ameaças

Grupo 1 e Grupo 3

Adriana Meireles
A82582

Carla Cruz
A80564



Ricardo Pereira
A73577

Tiago Ramires
PG41101

Braga, 23 de Março de 2020

1. Introdução

A área da engenharia de segurança consegue ser extremamente vasta, abrangendo inúmeras áreas de outros campos da ciência. Nos dias que correm é quase impossível não estar perto de um dispositivo que necessite de um determinado tipo de proteção, seja um sensor biométrico, um *smartphone*, um router, entre outros. Se atentarmos a outros casos, apercebemo-nos que a engenharia de segurança é também aplicada em dispositivos médicos, dispositivos bancários, sendo por isso uma área extremamente requisitada para conceber e assegurar a proteção dos mais diversos sistemas que podemos encontrar no nosso dia a dia.

Contudo, apesar de existir um ramo da engenharia orientado à segurança de sistemas, é muito frequente vermos a ocorrência de falhas extremamente básicas, muitas das vezes detetadas após ataques bem sucedidos, que implicam custos extremamente avultados para as organizações que deles foram alvo. Assim, a principal questão impõe-se: prevenir ou remediar?

Crê-se que a grande maioria das organizações já saiba o que é remediar, pois os ataques são frequentes e muitas vezes não necessitam de grande conhecimento da área para serem perpetuados, ou seja, atualmente os problemas de segurança surgem já após a implementação da aplicação estar feita e a sua resolução passa por colocar um ou mais técnicos informáticos a mitigar o problema. Já a prevenção, ainda é algo estranho às organizações, principalmente às empresariais, visto que existe sempre muito trabalho a ser feito, e fazer um esboço das medidas de segurança a implementar parece sempre ser uma perda de tempo. É este tipo de problemas que são abordados neste relatório, bem como as soluções para os resolver. A divulgação da área e das ferramentas existentes ou que poderão vir a existir são também aspetos importantes a salientar.

2. Desenvolvimento

2.1 Porquê fazer Modelação de Ameaças?

Primeiramente, antes de saber o porquê da modelação de ameaças ser feita, é necessário perceber de que é que se trata quando nos referimos a modelação de ameaças. Assim, modelação de ameaças é uma atividade principal e a prática fundamental no processo de construção de tecnologia confiável. O objetivo é identificar os ataques que um sistema deve resistir e as defesas que levarão o sistema ao estado defensivo desejado. Estes ataques expõem e exploram possíveis pontos fracos que afetarão o sistema modelado de maneira negativa.

Assim, a modelação de ameaças é importante ser realizada pois deve identificar e eliminar problemas de design: identificar pontos fracos de segurança ou chegar a um conjunto de necessidades de segurança que devem ser construídas com base nos problemas de segurança que precisam ser resolvidos.

Uma vez identificados, os requisitos de segurança, quando implementados, identificarão ameaças prováveis e as consequências prováveis de um ataque bem-sucedido e isto é o método de investigação para identificar um conjunto apropriado de defesas.

Enquanto alguns métodos de modelação de ameaças se concentram na identificação de ameaças e problemas de segurança, outros métodos também executam uma avaliação dos riscos resultantes, classificando as consequências e a probabilidade de ameaças. Esses métodos também são chamados de Análise ou Avaliação de Ameaças e Riscos. Esta classificação pode ser usada para priorizar defesas.

2.2 Quando fazer Modelação de Ameaças

Idealmente, a modelação de ameaças é aplicada assim que uma arquitetura é estabelecida. Não importa o quão tarde é desenvolvida a modelação de ameaças, é sempre essencial entender os pontos fracos nas defesas de um projeto.

Assim sendo, a modelação de ameaças deve começar quando as principais estruturas, as principais componentes ou funções de uma arquitetura, são conhecidas. Antes deste ponto, pode ser desperdiçado muito tempo pois poderá ter de ser tudo reformulado à medida que a estrutura começa a mudar. Começar muito tempo mais tarde pode significar que mudanças estruturais significativas ou adicionais necessárias para segurança só serão descobertas após o esgotamento dos prazos e dos recursos alocados para o desenvolvimento. Requisitos e restrições amplas ajudam a definir uma arquitetura, e esta torna-se mais específica à medida que as coisas se definem melhor e a segurança desta deve estar presente desde o início.

Contudo, é importante ainda referir que embora as equipas sejam incentivadas a executar a modelação de ameaças durante processo de definição estrutural, isto pode não ser possível de realizar, mas ainda asssim a modelação de ameaças é um exercício útil, independentemente de quão próximo o sistema esteja da implementação ou de quanto tempo ele esteja em uso.

2.3 Atualização da Modelação de Ameaças

Nem sempre é possível saber quando devemos realizar a atualização do modelo. Portanto, de seguida iremos apresentar uma lista parcial de revisão que poderão indicar a necessidade de atualização, nomeadamente quando:

1. Se realizam alterações que afetam o processamento, a manipulação ou a classificação de dados pelo seu software. Por exemplo, alterações no conteúdo confidencial, formatação do fluxos de dados, alterações relacionadas a algoritmos criptográficos, chaves e gerenciamento de chaves, etc.

2. Se adiciona uma nova subcomponente, repositório de dados, mesmo que esta alteração pareça pequena e não esteja diretamente relacionada à segurança.

3. Se existem alterações adicionais nos controlos de segurança e funcionalidades:

    - **Autenticação** - adição ou alteração de um método ou mecanismo de autenticação
    - **Autorização** - mudança nas relações de confiança entre quaisquer componentes no sistema (alteração dos níveis do usuário, alteração das permissões de acesso a dados) e ainda adição ou alteração de um método ou mecanismo de autorização
    - **Registo, monotorização e alerta** - adição ou alteração da monitorização da aplicação, análise de negócio, auditoria e conformidade de requisitos
    - **Criptografia** - adição ou alteração da funcionalidade criptográfica: algoritmos de hash, salt, algoritmos de encriptação/ desencriptação, configuração SSL / TLS, gerenciamento de chaves, etc.

4. Se existir a introdução ou alteração dos canais de comunicação entre as subcomponentes, backend, etc. Um novo fluxo de dados pode precisar de ser autenticado, autorizado e protegido "em trânsito".

Como regra geral, alterar ou adicionar dados produzidos externamente ou utilizados internamente - por exemplo, armazenamentos de dados, mensagens de erro descritivas, arquivos temporários etc. - devem ser tomados em conta pois podem levar à reconsideração do modelo de ameaça aplicado. Importante referir que é impossível criar uma lista de verificação abrangente para tudo que invalide um modelo de ameaça, portanto, deve usar-se a lista acima como referência e ainda realizar os ajustes necessários para ir ao encontro das necessidades do nosso desenvolvimento de software.

2.4 Como fazer Modelação de Ameaças

O processo de modelação de ameaças geralmente envolve algumas, distintas mas relacionadas, sub-atividades:

- Uma descrição inicial, possivelmente incompleta, da estrutura, casos de uso, casos de abuso e recursos aos quais o sistema está sujeito ou limitado. Isso geralmente é representado como um diagrama que descreve o sistema e mapeia (alguns dos) os possíveis pontos de ataque fora do sistema. Este pode ser feito em diferentes níveis de formalidade, desde documentos de especificação até desenhos na parte de trás de um envelope - mas a descrição deve representar com precisão o sistema que está a ser modelado.

- A identificação de um conjunto de possíveis ameaças que seriam relevantes para o sistema em análise, como elas iriam ser apresentadas aos vários cenários possíveis e o que poderia ser feito para mitigá-las.

Existem algumas maneiras populares de expressar esta descrição e identificação, nomeadamente: representações da arquitetura, diagramas de fluxo de dados, bibliotecas de possíveis ameaças, listas de objetivos de segurança, vetores de ataque e possíveis atenuações, diagramas de sequência, entre outros.

O formato específico é menos importante do que a sua utilidade para os modeladores. Se este tiver todos os detalhes importantes sem ser sobrecarregado com informações sem relevância, e as ameaças sejam relevantes e bem definidas, iremos "concluir"que funciona. Em geral, uma representação conterá mais do que um dos pontos apresentados ou até mesmo a maioria deles. Ainda assim, na maioria das modelações de ameaças, um ou mais dos elementos incluídos estarão incompletos. Estas modelações estão sujeitas a revisões à medida que existem mais informações disponíveis.

Existem muitas maneiras possíveis de criar a modelação de ameaças. Um processo válido é aquele que é capaz de identificar ameaças em potencial. Iremos apresentar abaixo algumas diretrizes para identificar ou desenvolver uma metodologia que funcione para casos em específico.

Como observado anteriormente, a modelação de ameaças identifica as ameaças às quais um sistema deve resistir e as defesas que levarão o sistema ao estado defensivo desejado. Este estado defensivo desejado é descrito como um conjunto de requisitos de segurança para o sistema a ser implementado. Este conjunto de requisitos de segurança leva a requisitos de teste de segurança que definem o alcance do teste de segurança do sistema. Se os requisitos de teste de segurança não forem definidos, o teste de segurança será realizado às cegas, aumentando o custo do esforço e reduzindo a abrangência do teste.

Tradicionalmente, os objetivos do sistema são definidos em duas categorias de requisitos:

- **Requisitos funcionais** - definem o comportamento ou função específica de um sistema.
- **Requisitos não funcionais** - também conhecidos como requisitos de qualidade, podem ser utilizados para decidir uma operação de um sistema, em vez de comportamentos específicos. Estes cobrem várias áreas, nomeadamente: Segurança e Privacidade, Acessibilidade, Capacidade de Resposta e Escalabilidade.

Cada requisito de segurança é composto por 3 partes:

- **O problema** - Tentamos perceber qual é a possível fraqueza a ser resolvida. Alguns profissionais mapeiam as descobertas para uma base de dados Common Weakness Enumeration (CWE) quando apropriado.
- **O control** - Esta é a tarefa que precisa de ser feita ou a operação que precisa de ser executada. Normalmente é escrito numa linguagem independente da tecnologia e foca-se nos critérios de aceitação sem especificar de que forma podem ser alcançados.
- **Diretrizes de implementação** - Apesar de opcional, este explica de que forma é que o ponto em cima pode ser alcançado.

Assim, modelação de ameaças é uma boa oportunidade para identificar restrições que devem ser aplicadas à implementação e para levantar possíveis sinais de alerta que devem ser considerados na implementação e no teste.

2.5 Falha na Modelação de Ameaças

Quando estamos a construir um modelo de ameaças torna-se imperativo perceber como é que este pode falhar bem como é que pode ser bem sucedido.

De acordo com "SAFECode membersJim DelGrosso of Cigital and Brook Schoenfield of Intel addressed", existem 6 mitos que podem ser considerados como falhas na modelação de ameaças:

- O facto de se fazer testes de intrusão com ferramentas e pessoas então já não é necessária a modelação de ameaças;
- O facto de o sistema já estar construído e implementado então já não é necessária a modelação de ameaças;
- O facto de se ter criado um modelo de ameaças quando o sistema foi construído então já não é necessário fazê-lo novamente;
- O facto da modelação de ameaças ser um processo complicado;
- O facto de não possuir profissionais na área então não podem fazer modelação de ameaças;
- O facto de se estar a fazer modelação de ameaças nos momentos certos então já não é necessário realizar testes de intrusão nem revisões no código;

As falhas apresentadas anteriormente são consideradas como "falhas de mentalidade". Por outro lado, também existem falhas práticas causadas pelo incumprimento de uma metodologia adequada:

- falha em controlar o alcance da análise. É necessário controlar o que realmente será analisado;
- foco em áreas que já são bem conhecidas. Por exemplo, peritos na criptografia pesquisarem intensivamente nas nuances da mesma quando o uso desta é desnecessário;
- Não definir o que é o "Sucesso";

As principais armadilhas podem ocorrer quando as partes interessadas com conhecimentos cruciais não são incluídas no processo ou ainda quando a equipa falha em acordar uma atualização ao sistema bem como na falta de comunicação por parte dos membros da equipa. Contudo, podemos acrescentar que este último fator é o que contribui mais para falhas na modelação de ameaças pois, mais tarde, são identificadas ameaças que não foram reconhecidas durante o design do modelo.Deste modo, torna-se importante incluir os principais interessados e garantir que os formatos adotados permitam uma comunicação clara e precisa dos detalhes.

Outra armadilha na modelação de ameaças são as superfícies de ataque que podem ter sido derivadas de falhas na comunicação.Por exemplo,as expectativas sobre interfaces bem como interações com sistemas externos que não foram bem explicadas nem compreendidas durante a modelação de ameaças. Também a inclusão de funcionalidades fornecidas por terceiros podem não ser conhecidas. Portanto, caso estas falhas sejam descobertas, é necessário uma revisão do modelo de ameaça.

Em suma, usar a modelação de ameaças para focar exclusivamente na identificação de ameaças é um erro. Deste modo, é necessário identificar ameaças específicas,documentar a topologia, identificar falhas no controlo do ambiente e localizar violações de política que não são ameaças específicas à segurança.

2.6 Construção de uma Boa Equipa

Ao liderar o processo de modelação de ameaças para além de tudo o que foi referido nos pontos anteriores é necessário ter uma equipa de qualidade o que inclui engenheiros bem como pessoas responsáveis pela documentação , entre outros.

Podem ser considerados os seguintes especialistas na parte técnica:

- **Arquitetos de solução**:possuem conhecimentos aprofundados da estrutura total de todo o sistema;
- **Arquitetos**:pessoas que são especialistas na estrutura para partes definidas do sistema;
- pessoas com experiência em rede, sistemas operativos,processos de implementação,cloud, garantia de qualidade,design de software;
- **implementadores**: podem ter informações para melhorar a abrangência e aplicabilidade do modelo.

A modelação de ameaças é para ser feito em equipa. Portanto, reduzir os membros da equipa pode parecer vantajoso no início mas a longo prazo é mau pois prejudica o resultado do modelo de ameaças devido ao pequeno número de membros envolvidos no processo.É recomendado criar um rascunho que é preenchido à medida que o processo evolui e que o conhecimento é adquirido. Isto permite construir a perceção numa equipa e levar a um maior conhecimento dentro da mesma conforme o processo é desenvolvido.

Para além de toda a parte técnica, é fundamental ter pessoas com diferentes abordagens/-talentos. Por exemplo, ter alguém que seja capaz de imitar o processo realizado por um atacante bem como ter alguém que desvende a sequência de eventos e transformações de dados à medida que avança no processo. Estes dois, apesar de oferecerem visões distintas, podem complementar-se e descobrir o maior número de ameaças a serem avaliadas.

2.6.1 Selecionar Bons Modeladores

Os envolvidos na modelação de ameaças devem possuir licenciatura em engenharia e arquitetura de software que terá de involver arquitetura empresarial e um conhecimento básico de segurança de software. Não é necessário mestrado visto que o conhecimento e as habilidades podem ser obtidos com a experiência.

2.7 Extensão da Modelação de Ameaças

Ao analisar-se a extensão na modelação de Ameaças deve ter-se em consideração 3 questões:

1. **Quão profundo vão os modeladores na estrutura de um sistema?**

   Um sistema deve averiguar quando a modelação de ameaças depende do tamanho e complexidade do sistema, bem como ambiente de execução, ambiente específico em que o software será executado, linguagem que o software é escrito, entre outros.

2. **Quais os pré-requisitos para o começo da modelação de ameaças?**

   Quanto aos pré-requisitos existem alguma discórdia. Alguns especialistas sugerem listar todos os ativos associados a todos os ataques possíveis. Enquanto que outros dizem que

as listas iniciais necessitam de incluir todos os vetores a partir dos quais uma ameaça se possa materializar, bem como todos os pontos no sistema em que as mitigações podem ser implementadas.No entanto, o principal problema será escolher,dessas listas,os cenários de ataque cuja perda seria irrelevante ou não interessaria a possíveis atacantes. Perante recursos limitados, a priorização de quais controles/defesas implementar será importante.

3. **O que permite a finalização da modelação de ameaças?**

   Após selecionados os pré-requisitos e escolhida a metodologia existem algumas formas de saber que o trabalho está completo. Uma delas é quando todos os ataques possíveis, que poderiam afetar a organização ou o produto, sejam expostos. Outra é quando as defesas possíveis para protegerem as superfícies de ataque tiverem sido estabelecidas.

Qualquer que seja o sistema em estudo,é no início que deve ser definido um limite para a sua análise de modo a que seja delimitado um ponto de saída que seja considerado como a conclusão da modelação de ameaças.

2.8 Metodologia

Na modelação de ameaças existem várias metodologias aceites pela indústria tais como:

- **Processo de Modelação de ameaças da Microsoft:**segue uma abordagem passo a passo que se foca na identificação de ativos e na arquitetura, decomposição da aplicação,identificação e documentação das ameaças e classificação das mesmas de acordo com a sua severidade.

- **P.A.S.T.A (Process for Attack Simulation and Threat Analysis):** É um processo de sete etapas. Para além de alinhar os objetivos de negócios aos requisitos técnicos, também leva em conta os requisitos de conformidade, a análise do impacto nos negócios e uma abordagem dinâmica à gestão, enumeração e pontuação de ameaças.

- **Trike:**metodologia que segue um modelo de requisitos que garante que cada nível de risco atribuído a cada ativo é classificado como aceitável pelas partes interessadas do sistema.

- **ATASM(Architecture, Threats, Attack Surfaces, and Mitigations):** abordagem que destaca a importância de compreender a estrutura de um sistema(arquitetura).A arquitetura é dividida nas suas componentes lógicas e funcionais(decomposição) de modo a descobrir os possíveis ataques de superfície(entradas e saídas do sistema). A decomposição também é usada para definir os pontos onde a defesa será construída, sendo as mitigações colocadas em limites defensíveis.

- **Biblioteca de ameaças/ Abordagem de lista:**segue um conjunto pré-definido de ameaças comuns no qual a equipa tentará reconhecer as instâncias delas no produto procurando por gatilhos.

- **Modelação de Ameaças Rápida:**variações mais leves de outras metodologias assim como abordagens adicionais que usam classificações rápidas e outras maneiras de obter um resultado semelhante em menos tempo.

Para além das metodologias referidas anteriormente, existem outras para enumeração e descoberta de ameaças tais como: Microsoft's STRIDE,Top X Threats(OWASP's Top 10). Para

o ranking das descobertas existem opções como:CVSS (Common Vulnerability Scoring System),Open Group™ Factor Analysis of Information Risk (FAIR), CWSS (Common Weakness Scoring System) e CWRAF (Common Weakness Risk Analysis Framework).O CVSS tem como alvo *vulnerabilities*, enquanto o FAIR e o CWSS têm como alvo *weakeness*(estes conceitos são explicados mais à frente).

Cada organização irá adotar um conjunto de diferentes práticas que sirva às suas necessidades. Portanto, em vez de escolher uma abordagem específica em detrimento de outras deverá consultar as recomendações baseadas nas lições aprendidas pelos membros da organização SAFECode.

2.9 Terminologia

Como é sabido, no mundo empresarial, as pessoas que estabelecem e que acordam os negócios, nem sempre são as melhores para o fazer, uma vez que na grande maioria das vezes não compreendem sequer a essência do sistema que vendem/compram. Este "à vontade"que se experiencia na contratualização de um projeto é a razão pela qual existem tantas falhas entre o que um cliente pede e aquilo que obtêm no final.

Posto isto, como é que é possível um cliente/vendedor comprar/vender um produto sem referir com exatidão aquilo que se pretende? É neste sentido que a terminologia é importante e uma pessoa que esteja na linha da frente do negócio deve munir-se dos termos corretos para não correr riscos.

Como é referido num artigo da área, existem palavras que são erradamente usadas como substitutas de outras, tudo porque aquele que as usa fá-lo pensando no significado que as mesmas têm na gíria popular, como por exemplo, *threat*, *risk* e *vulnerability*:

- *threat* - causa potencial de um acontecimento perigoso para uma organização/sistema
- *risk* - consequência da incerteza inerente nos objetivos;
- *vulnerability* - um problema que se manifesta numa determinada implementação

Outra confusão acontece com os termos *weakness* e *vulnerability*: uma diferença fulcral entre ambas é visível durante a modelação - nesta fase inicial é possível prever *weaknesses* contudo é impossível detetar vulnerabilidade porque o sistema ainda não está implementado.

2.10 Manuseamento de Sistemas Complexos

Por vezes, existem sistemas cuja modulação tem que ter em conta uma enorme quantidade de componentes ou então componentes com funções completamente distintas e nestes casos é necessário garantir que, apesar da dificuldade, tudo fica muito bem especificado. Exemplos destes sistemas são os dispositivos IoT que para além de trabalharem com serviços *cloud*, aplicações *web* e móveis, etc ainda integram componentes físicos como sensores, câmaras, entre outros.

A solução ideal para este problema implica pôr em prática o conceito "dividir e conquistar"que é transversal a muitas áreas de trabalho. Neste caso, numa primeira fase modela-se o sistema na totalidade, especificando-se os comportamentos e as interações entre os principais constituintes do mesmo. Seguidamente, faz-se uma modulação mais orientada a cada um destes constituintes, olhando-se agora para os comportamentos e interações de cada um internamente.

Desta forma, parte-se um grande problema em problemas cada vez mais pequenos, permitindo a sua modelação e ignorando problemas que podem ser vistos separadamente.

2.11 Tecnologias/Ferramentas

A questão que se impõe nesta secção é muito simples: que ferramentas e tecnologias existem para fazer a modelação de ameaças de um determinado sistema? Há ferramentas que podem ajudar no processo de modelação mas não há um substituto capaz de executar uma análise tão eficaz quanto a que envolve interação humana nem que seja capaz de cobrir todos os aspetos de um caso de estudo. Neste sentido, uma equipa de especialistas reuniu alguns factos que traduzem o estado atual da área em questão, factos esses que são apresentados a seguir:

- as soluções já existentes não são suficientes para responder à crescente necessidade de mecanismos de segurança;
- a procura e o interesse em modelação de ameaças tem vindo a crescer;
- há algumas ferramentas cuja utilização é uma barreira de entrada na área, contudo há também iniciativas para vencer essa barreira, nomeadamente o treino de pessoal interessado, o destacamento de especialistas para formarem interessados na área, etc;
- iniciar o processo de modelação pode ser complicado, contudo esse é um problema transversal a qualquer área de trabalho;
- os resultados costumam ser vistos como uma perda de tempo, visto que não há um efeito imediato daquilo que se construiu/modelou.

Com base nos factos anteriores, a mesma equipa de especialistas deduz o formato da ferramenta que seria ideal para a modelação de ameaças: algo que apresentasse resultados que tivessem impacto imediato no que se estava a construir, bem como a possibilidade de adaptação da ferramenta às necessidades daquele que a usa. Assim, este grupo criou uma lista de requisitos que se crêem ser fundamentais para que uma ferramenta deste tipo seja útil e interessante:

- a possibilidade de modelar a arquitetura de um sistema através de diagramas e de separação de modelos ("dividir para conquistar");
- a existência de um espaço para anotar problemas que devem ser solucionados, bem como *updates* e tarefas similares;
- complementarmente ao requisito anterior, deve existir um rastreamento e uma classificação dos problemas anotados de acordo com as normas que já existem;
- a solução deve abranger qualquer método que exista e permitir ao utilizador escolher aquele que lhe apetecer;
- deve haver uma parte da ferramenta que permita construir um relatório onde são automaticamente incluídos problemas anteriores que não tenham ainda sido resolvidos;
- deve ser possível exportar/importar modelos do género dos que já existem para modelação de problemas noutras áreas, para que a ferramenta se adapte ao que já existe no mercado.

Com os requisitios anteriores e uma pesquisa rápida na rede, é possível verificar que ainda não existe uma ferramenta que ofereça todas aquelas opções e as que existem baseiam-se essencialmente em anotações ou contrução de diagramas. Atualmente, as medidas que as empresas tem tomado passam pela construção das suas próprias ferramentas ou então pelo uso de ferramentas mais básicas, como *Visio* e/ou *Excel*.

2.12 Inclusão da modelação de ameaças no ciclo de vida do desenvolvimento

Qualquer pessoa que trabalhe em desenvolvimento de sistemas de software já se deparou com as várias fases por onde passa um produto antes deste estar concluido. Ora, aqueles que estudam a modelação de ameaças acham que esta prática deve ser formalmente incluída neste ciclo uma vez que, à semelhança de outros aspetos, consequências futuras graves podem ser evitadas. Atualmente, somos capazes de identificar exemplos de alguns sistemas que são previamente modelados e que permitem a antevisão de diversos problemas:

- conceção de páginas *web* que permitem testar dispositivos médicos *online* através de máquinas de estados;
- modelação de problemas aplicacionais com recurso a ferramentas formais (*alloy*, *electrum*, *TLA+*, etc) como por exemplo, sistemas de controlo de velocidade de automóveis.

Posto isto, não há razão para que a conceção de um modelo de ameaças não seja feito aquando da modelação da arquitetura. Contudo há ainda outro aspeto importante a focar da modelação de ameaças que é a dinâmica do sistema, pois como é sabido, as aplicações de *software* estão em constante mudança e a introdução de novas funcionalidades requer uma modelação prévia. Para além disso, é frequente observar-se a modificação de requisitos da aplicação a construir e a segurança da mesma só pode ser pensada quando a arquitetura do sistema estiver bem modelada e bem fundamentada. Em suma, podemos agrupar as diferentes fases do ciclo de vida de desenvolvimento de uma aplicação pela seguinte ordem: definição da estratégia de segurança, avaliação da arquitetura, conceção do modelo de ameaças, análise ao design, plano de teste. Note-se que estas são as fases que interessam ao assunto que aqui se está a abordar, havendo outras que não foram aqui expressas mas que são igualmente importantes.

2.13 Exemplos de Modelação de Ameaças

Fazendo um ponto de situação, tudo o que se tem falado até agora está relacionado com as boas práticas de segurança que devemos ter quando desenvolvemos ou construimos uma aplicação de um qualquer tipo. Por isso, faz todo o sentido apresentar uns quantos exemplos, retirados de um artigo escrito pela organização *SAFECODE*, pelo qual se tem guiado este relatório.

2.13.1 Aplicação *Web* de Atribuição de *Feedbacks*

Comece-se pelo exemplo que consiste na modelação de ameaças de uma aplicação *web* de atribuição de *feedbacks*. Essa aplicação permite que os utilizadores façam um registo inicial na plataforma, que façam o início de sessão, que atribuam o *feedback* desejado e que terminem a sessão na aplicação. Atente-se agora na modelação de ameaças da aplicação referida anteriormente:

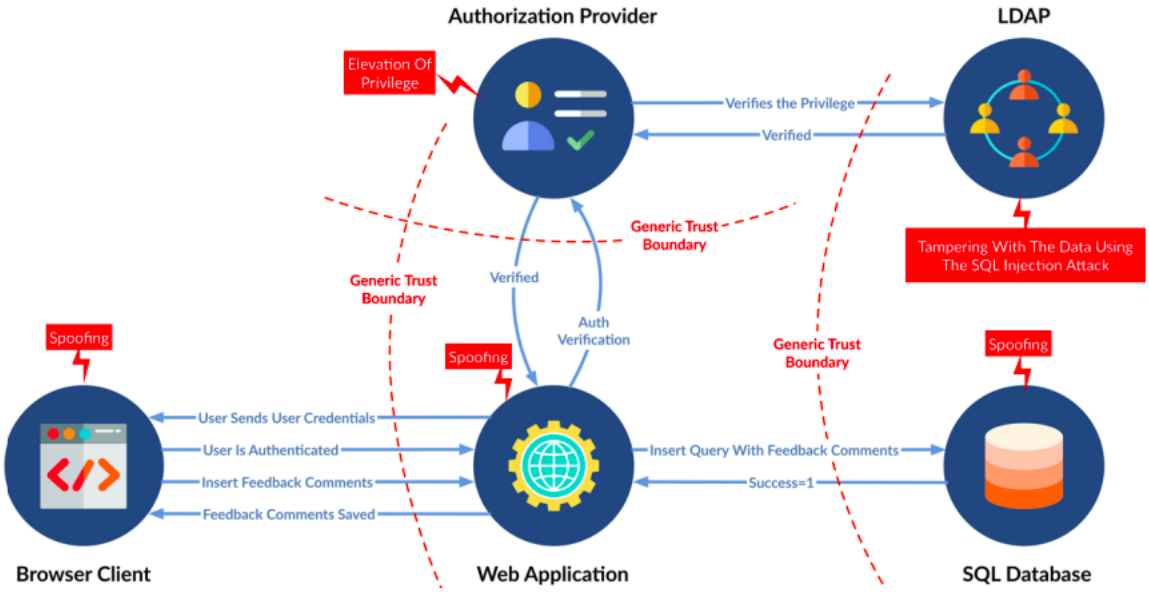


Figura 2.1: Diagrama de fluxo de dados

Como se pode ver, modelar não é mais do que discriminar num diagrama o máximo possível de sistemas que existem na aplicação que permite pensar em ameaças que possam existir. Vemos vários pontos onde podem ocorrer ataques de *spoofing* e *elevation of privileges* e vemos também as *trust boundaries* (zonas de confiança).

2.13.2 Autenticação em Dispositivos IoT

Considera-se uma rede IoT aquela que engloba vários dispositivos digitais que comunicam ativamente entre si com o intuito de automatizar e simplificar determinadas tarefas do quotidiano das pessoas. Contudo, com a praticidade é inevitável o aparecimento de novas ameaças à nossa segurança, algo que é evidente com as constantes notícias de documentos que são roubados de *armazenamentos online*, *emails*, etc, documentos que dantes eram guardados em locais físicos onde facilmente se sabia se alguém lá tinha entrado. Nos dias contemporâneos é tudo mais subtil e para o ser humano comum é muito mais difícil perceber as ameaças que estão inerentes aos sistemas que usa no seu quotidiano.

Tendo em conta tudo o que foi dito anteriormente, a autenticação é um método de segurança que deve estar implementado sempre que possível. É verdade que por vezes é complicado e cansativo estar constantemente a fazê-lo mas é também uma das barreiras mais simples de implementar, rouba apenas alguns segundos ao utilizador comum e se bem implementada é impossível que qualquer *hacker* a vença. Existem vários tipos de autenticação:

- **pessoa e dispositivo** - Este tipo de dispositivos são normalmente aqueles que nos acompanham no dia a dia, como o *smartphone* (no entanto, podem haver outros de outro género); este é um bom exemplo para pensar no que era possível um *hacker* fazer caso não possuísse qualquer tipo de autenticação. Ainda assim, ter um sistema de autenticação robusto nem sempre resolve o problema na sua totalidade - é necessário fazer uma mudança regular destas chaves;

- **dispositivo e serviço** - No canal de comunicação que envolve estes dois lados, um dos problemas insurgentes é a crescente quantidade de dispositivos que participa passiva e ativamente no sistema IoT. É fundamental que a gestão do serviço que lida com estes

dispositivos todos seja feita da melhor forma, não permitindo que os dispositivos sejam associados a utilizadores errados. No que diz respeito ao sistema de autenticação, o mesmo pode usar algoritmos simétricos ou assimétricos, sendo que para ambos os casos, a informação que diz respeito a estes algoritmos deve ser guardada em *chips* protegidos contra *tamper attacks*.

- **computador e serviço** - Esta é talvez a forma de autenticação com que o ser humano comum tem mais contacto, visto que é a forma de autenticação que envolve o par nome de utilizador e palavra-chave. Normalmente, ambos estão associados a várias aplicações e por esse motivo é importante diferenciá-los e estabelecer um número limite de tentativas de início de sessão. Assim caso um *hacker* descubra uma das palavras-chave de uma aplicação, não tem acesso à informação de todas.

- **serviço e serviço** - Este tipo de autenticação acontece quando um determinado serviço pede a outro informação de um utilizador seu. Aqui, o serviço que tem a informação do utilizador necessita de pedir autorização a este último para que permita que essa informação seja passada a outro serviço. Um exemplo deste tipo de partilha são os *cookies* para efeitos de publicidade, sendo ainda um exemplo muito básico. Alguns protocolos utilizados neste tipo de autenticação são o *OAuth* e o *OpenID*.

- **dispositivo para dispositivo** - Este é um tipo de autenticação que a maioria dos sistemas evitam pelo facto da sua ligação ser feita com base numa única chave partilhada por todos os dispositivos. É evidente que um possível atacante descobrindo esse segredo, era capaz de aceder a qualquer dispositivo sem grande dificuldade.

É imperativo estar-se atento a todas estas formas de ataques que existem no mundo tecnológico e esperar sempre a existência de *hackers* que tentam invadir os sistemas que se acham seguros, seja o auricular *bluetooth* ou o frigorífico ligado a uma rede *IoT*.

Como se pode ver, fazer esta modelação é bastante simples e o que se defende é isto: deve ser feita uma modelação de ameaças do sistema o mais aprimorada e exaustiva possível para que se diminua o nível e a quantidade de ameaças do sistema em questão.

2.14 Práticas de Modelação de Ameaças e Desenvolvimento *Agile*

Como se tem vindo a referir neste relatório, a modelação de ameaças é uma tarefa que deve ser incluída no ciclo de vida de desenvolvimento de uma aplicação e deve ser feita na fase da modelação da arquitetura. Não obstante, fazê-lo nem sempre é tão linear quando utilizamos a metodologia *Agile* pelo facto de determinadas partes do sistema poderem não estar suficientemente definidas para modelar o que quer que seja. Assim, a solução adotada é fazer o planeamento e a construção do modelo de ameaças e atualizá-lo sempre que for feita alguma alteração. Isto traduz-se nas seguintes etapas:

- **Sprint 0** - Iniciação e construção do modelo de ameaças com base no projeto geral;

- **Sprints 1** - Sempre que surja uma alteração ao código ou algo que invalide o modelo de ameaças, o mesmo deve ser alterado;

- **Release** - Fase em que se verifica se o modelo de ameaças reflete a segurança a ser implementada naquele sistema.

Por outro lado, temos a modelação de ameaças num ambiente *DevOps* onde o desenvolvimento e as operações de *software* são feitas em simultâneo. Posto isto, neste ambiente de desenvolvimento temos alguns pontos, que se podem materializar em problemas para a aplicação final. Um deles é a existência de ferramentas que automaticamente adicionam componentes ao *software*; ora se essas ferramentas forem comprometidas e começarem a adicionar componentes indesejados, ter-se-ão os produtos finais todos com a mesma falha, potenciando-se consequentemente o número de ameaças aos sistemas. Outro serão as mudanças que existem no ambiente que é executado, nomeadamente adição de portas, adição de protocolos, alterações ao servidor, alterações às permissões, configuração do sistema operativo, etc.

3. Conclusão

Tudo aquilo que se relatou neste documento não é mais do que um conjunto de boas práticas que desde sempre devem ser implementadas em qualquer tipo de projeto. Contudo, no caso em questão, as mesmas focam a área da engenharia de segurança e exemplificam tipos de medidas que se podem tomar para evitar determinados problemas extremamente simples e que podem implicar custos extremamente avultados para as organizações que neles incorrem.

A modelação de ameaças de um sistema deve indiscutivelmente, ser uma preocupação daqueles que querem construir *software* fidedigno e funcional, visto que uma análise prévia dos problemas que podem ocorrer no futuro permite solucioná-los mesmo antes da construção do sistema e se tal não for possível, permite delinear soluções de resposta imediata caso os mesmos se venham a verificar.

É agora mais do que evidente que argumentos tais como "é uma perda de tempo", "não existem ferramentas indicadas", "ninguém quer saber disso", entre outros, são argumentos infundados que já se podem refutar com situações práticas. Há relatos de várias organizações que começaram a investir nesta área após verificarem a presença de erros banais nos seus sistemas, alguns que resultaram em repercussões económicas impensáveis.

O mote para a inclusão destas técnicas nos ciclos de vida do desenvolvimento está lançado e a acompanhá-lo existem especialistas dotados de conhecimentos absolutamente úteis disponíveis a ajudar quem se mostre interessado em ingressar neste ramo. O objetivo é muito simples, angariar o máximo de recursos possíveis para combater este tipo de problemas e mais importante ainda: prevenir para não remediar.