Modelo de ameaças

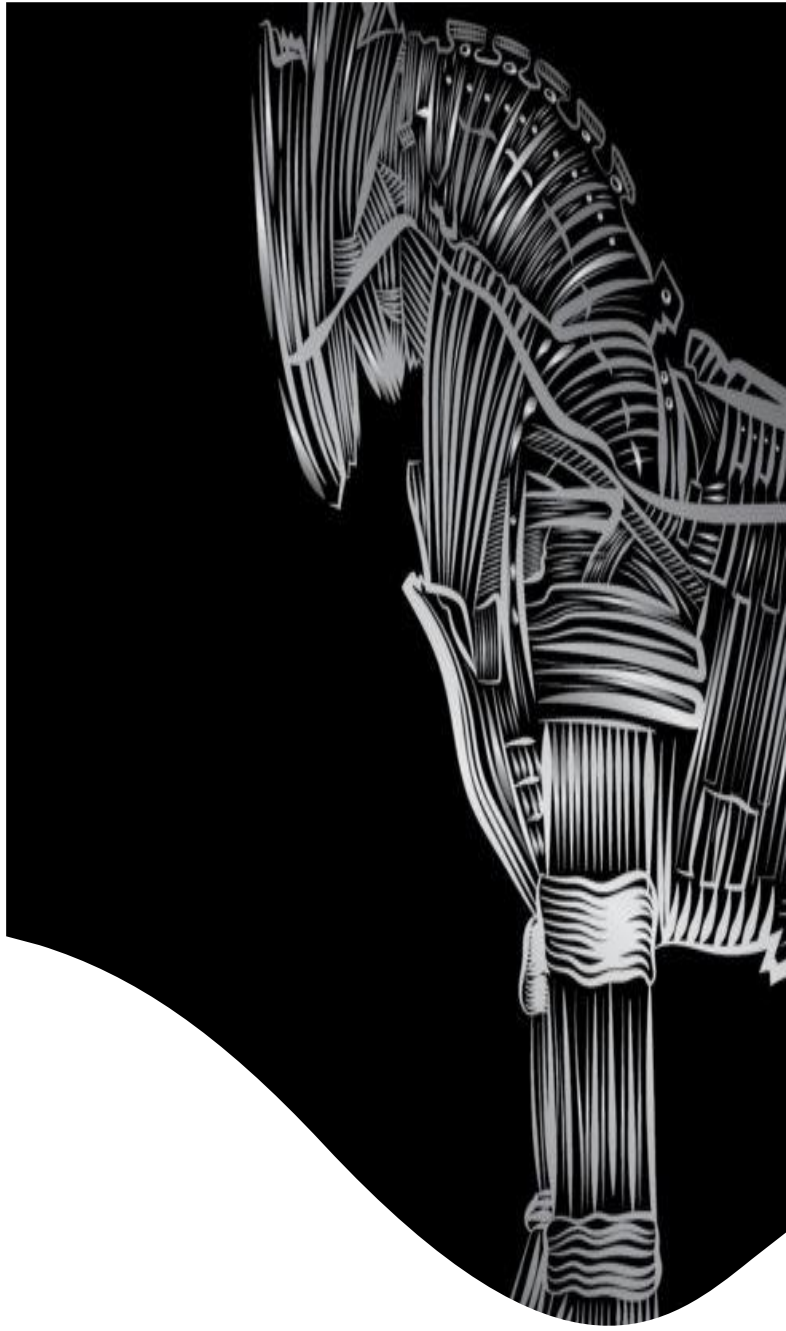
Operacionalizando o modelo

Operacionalizando o modelo

Comece registrando o objetivo do atacante. Essa é a raiz da sua árvore e o grande objetivo do atacante.

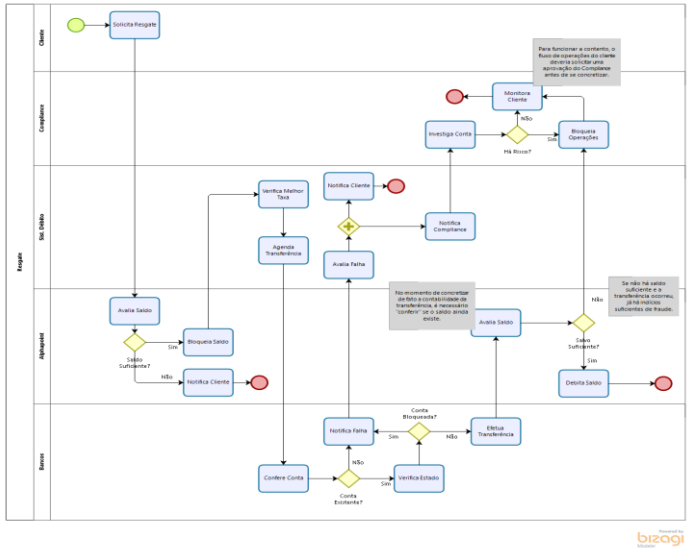
Derive então o passo imediatamente anterior que viabiliza aquele objetivo. Provavelmente há mais de um caminho que viabilize o ataque, então estes serão os ramos da árvore. Siga derivando, passo a passo, até encontrar as folhas que são os primeiros passos que o atacante terá que dar pra chegar a seu objetivo.

O raciocínio é construído de trás para frente, partindo do objetivo final para os primeiros passos.



Operacionalizando o modelo

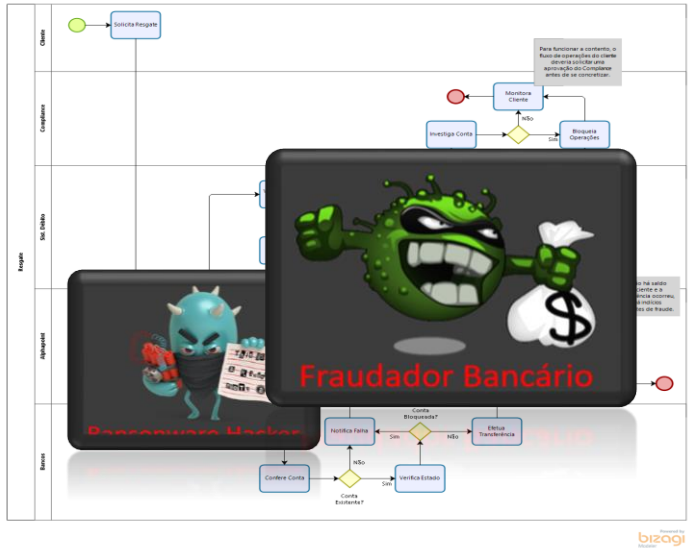
1. Identifique os processos de negócio envolvidos e sistemas e informações necessárias para operacionalizar esse negócio.



Operacionalizando o modelo

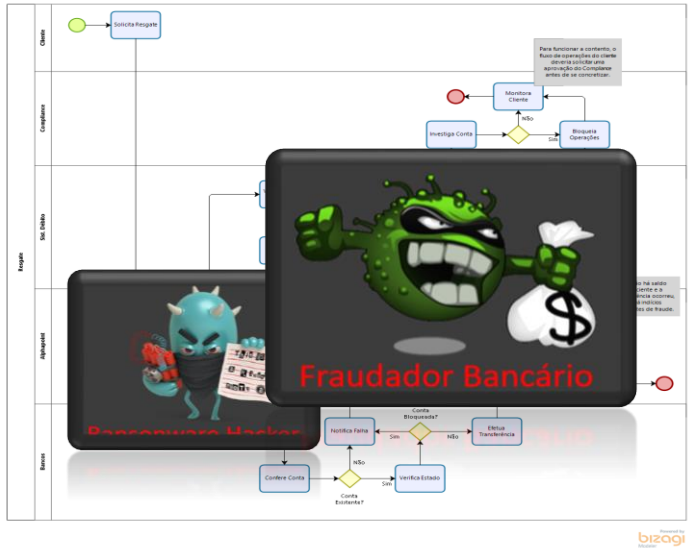
1. Identifique os processos de negócio envolvidos e sistemas e informações necessárias para operacionalizar esse negócio.

2. Identifique os atacantes mais prováveis desse negócio - Quem teria algum benefício abusando desse negócio?



Operacionalizando o modelo

1. Identifique os processos de negócio envolvidos e sistemas e informações necessárias para operacionalizar esse negócio.
2. Identifique os atacantes mais prováveis desse negócio - Quem teria algum benefício abusando desse negócio?
3. Identifique os macro objetivos desses atacantes, como, furto de cartões de crédito, furto de commodities, resgate por devolução de um ativo etc.

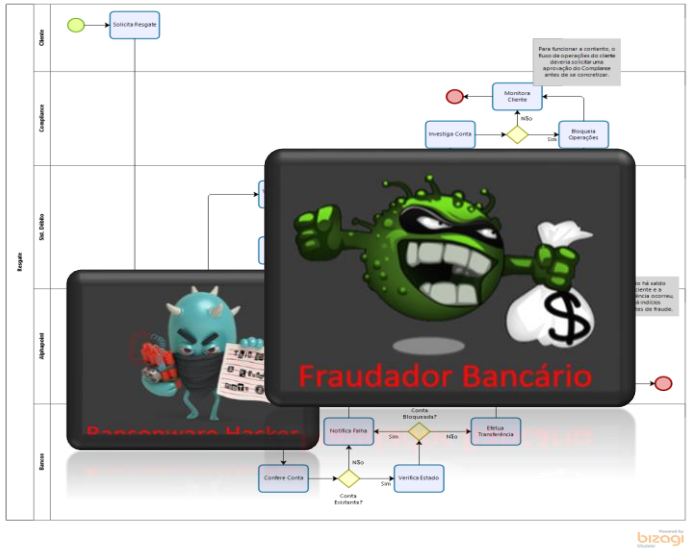


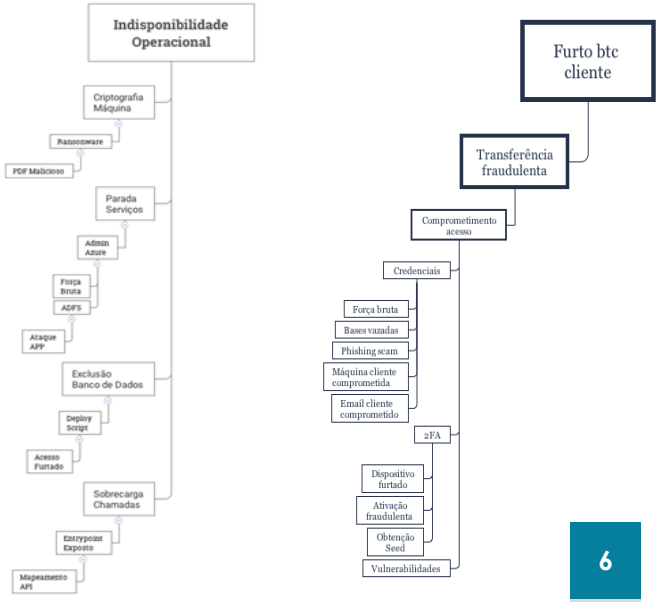
Indisponibilidade Operacional

Furto btc cliente

Operacionalizando o modelo

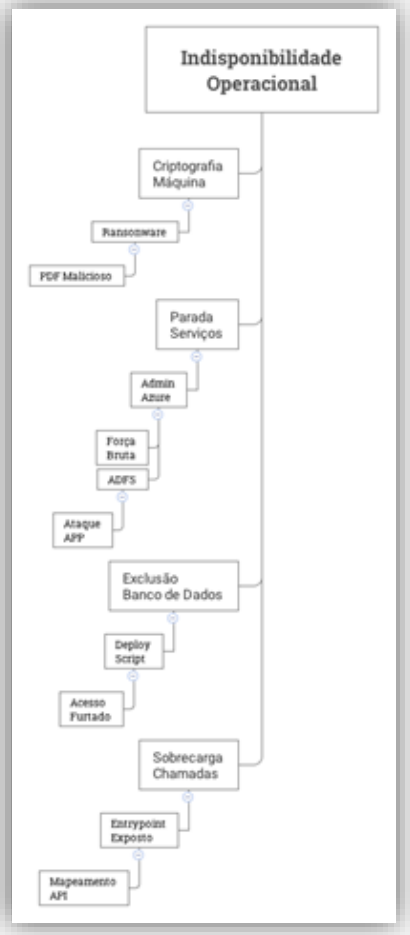
1. Identifique os processos de negócio envolvidos e sistemas e informações necessárias para operacionalizar esse negócio.
2. Identifique os atacantes mais prováveis desse negócio - Quem teria algum benefício abusando desse negócio?
3. Identifique os macro objetivos desses atacantes, como, furto de cartões de crédito, furto de commodities, resgate por devolução de um ativo etc.
4. Identifique os passos necessários (mecanismos) pelos quais esses atacantes conseguiriam chegar em seu intento macro.





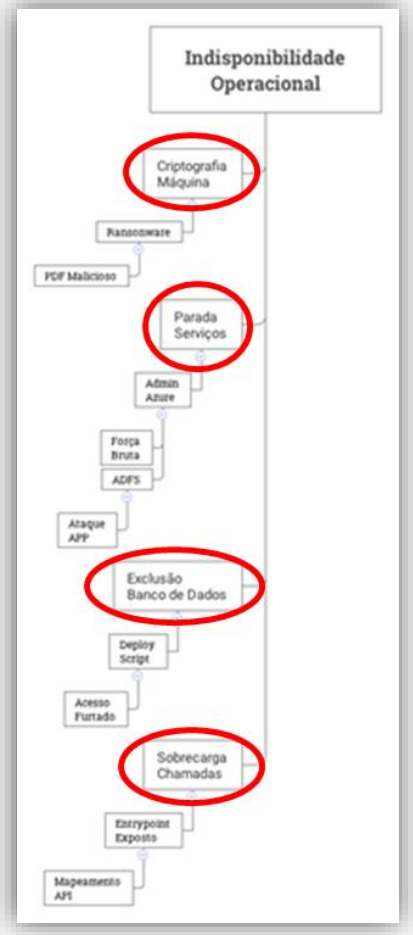
Operacionalizando o modelo

Neste exemplo, que propõe uma árvore de ataque de um Ransomware Hacker exigindo o resgate (Ransom) em criptomoeda. Basicamente ele consegue seu intento exigindo resgate pela "Disponibilidade do ambiente", logo ele quer causar uma indisponibilidade.



Operacionalizando o modelo

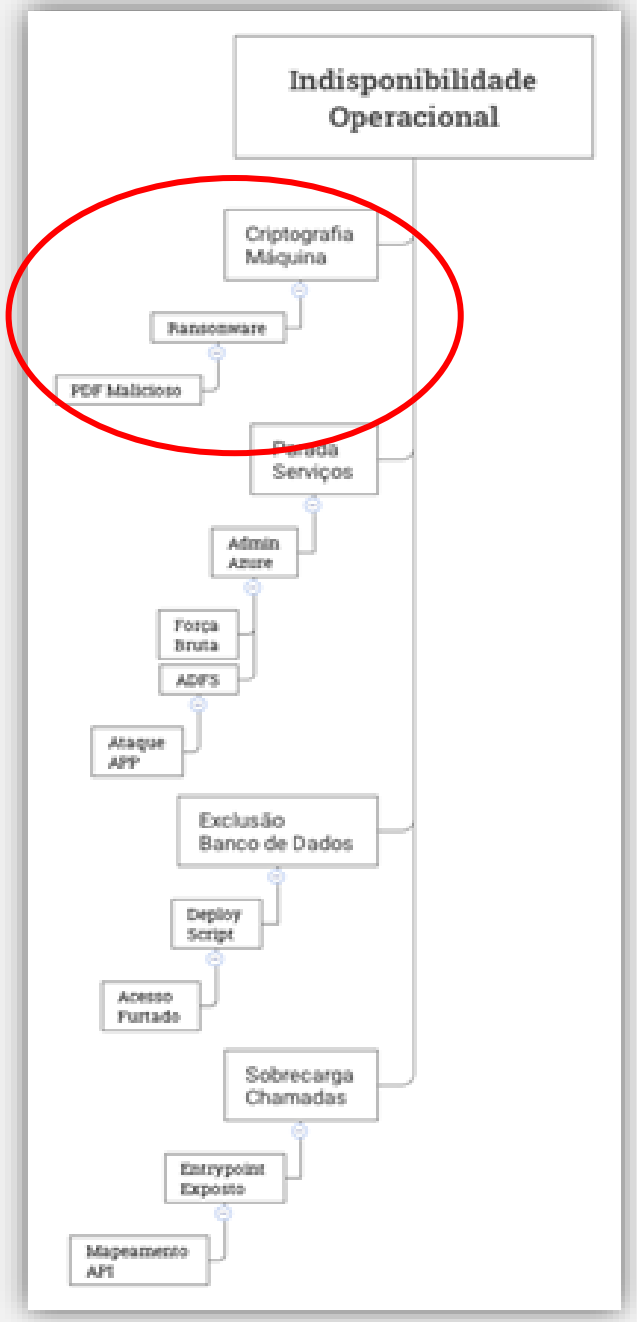
O primeiro ponto importante é notar que o objetivo final "Indisponibilidade Operacional" pode ser viabilizado através de quatro ramos "Criptografia das máquinas operacionais", "Parada dos serviços", "Exclusão dos Bancos de dados" e "Sobrecarga de chamadas". Não é incomum termos processos de negócio aparecendo nos ramos, mas fatalmente chegamos em tecnologia quando caminhamos para as folhas, pois é a tecnologia que provêm os serviços.



Operacionalizando o modelo

Descrevendo o raciocínio apresentado como exemplo nesse slide temos:

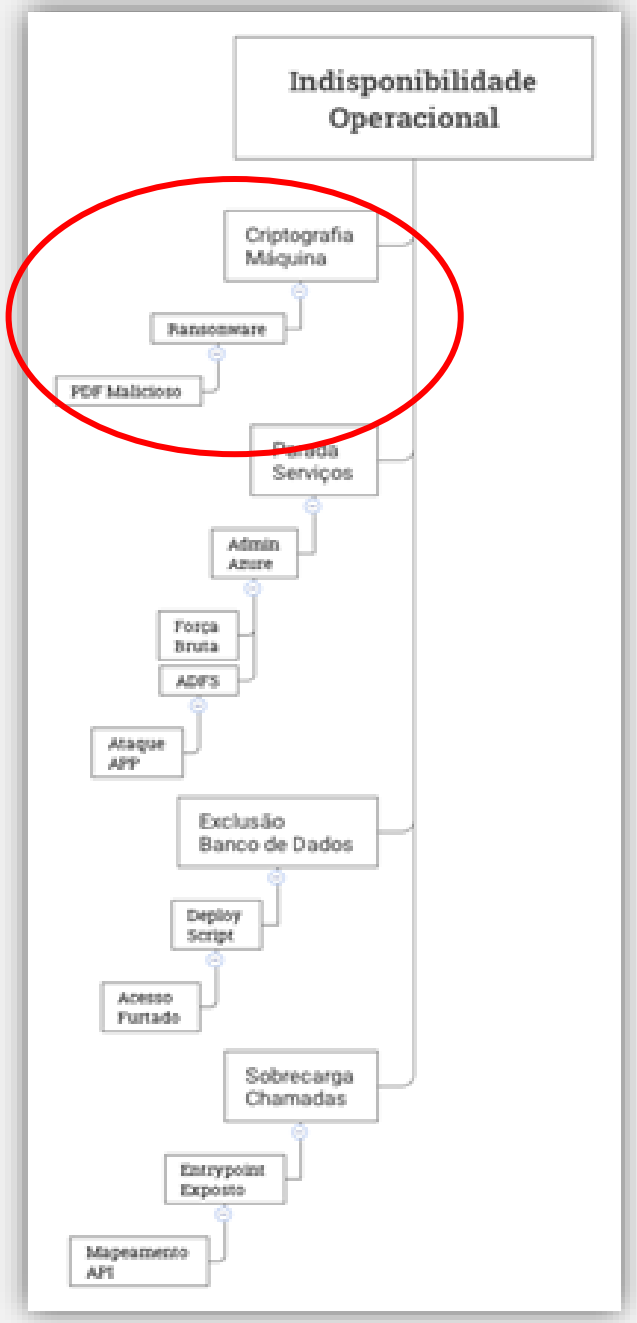
Através do envio de dados cadastrais de onboarding, esperados pelo processo de negócio, nosso atacante (Ransomware Hacker) enviará um PDF malicioso almejando a contaminação da máquina dos analistas, que interagem com essa documentação, dentre eles, analistas de Cadastro, Atendimento, Financeiro, Compliance, etc. A partir do controle de uma máquina do time operacional, dispara-se o Ransomware que vai se propagar e criptografar as máquinas do parque. Com o parque criptografado atinge-se o objetivo que era a Indisponibilidade operacional.



Operacionalizando o modelo

É importante que se mantenha a lógica na qual um alvo do passo analisado anteriormente (”Falha de validação” → “enviar Ransomware”) seja o mecanismo no próximo passo: “Ransomware enviado” → “Criptografar máquina”.

Essa árvore serve para se determinar os requisitos de implementação e arquiteturais (para impedir o ataque), o que e onde monitorar para se detectar e conter o ataque, viabilizando o planejamento da resposta a incidentes, com a validação dos processos de contingência, contenção e detecção dos incidentes.

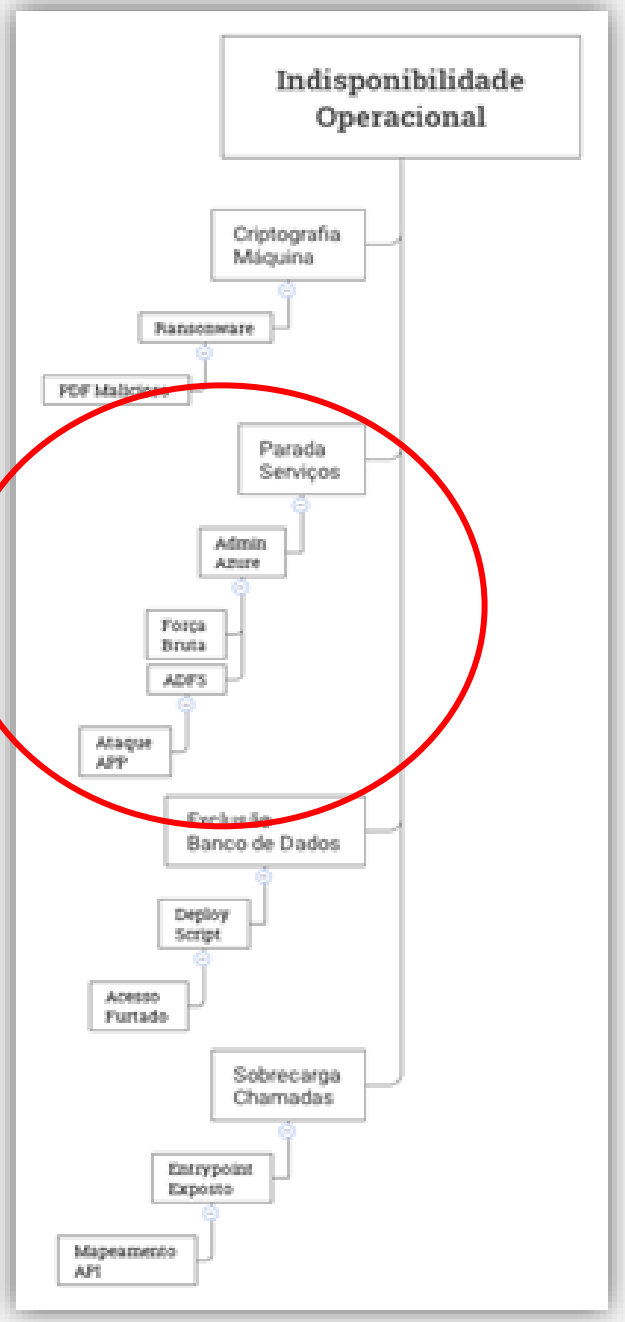


Operacionalizando o modelo

Ataques de força bruta às aplicações expostas e a interface administrativa. O atacante foca no controle das contas do ambiente, demandando resgate pelas mesmas.

O atacante pode implantar um mecanismo de acesso que o permitirá voltar ao ambiente sem esforço.

Neste ramo explora-se o cenário de controle operacional do ambiente.



Requisitos de segurança

O passo da “derivação de requisitos” percorre o modelo de ameaças e busca meios de eliminar ou restringir ao máximo as chances do atacante ter sucesso em cada etapa, isto porque, em geral, o atacante leva dias para executar os primeiros passos, e faz “barulho” no ambiente, permitindo que time de resposta a incidente o encontre. A medida que ele avança na árvore rumo a raiz, cada vez mais ele tende a ser sorrateiro e impedir que suas ações sejam detectadas.

No exemplo, quando uma máquina for identificada e criptografada (penúltimo passo do ataque), basicamente não há mais como impedir o ataque, apenas conter sua propagação e disparar as medidas para conter seus impactos.

A famosa lista de requisitos

1. **Módulo para validação e limpeza de arquivos (anexos) recebidos**
2. Múltiplos fatores de autenticação
3. Política de senhas robusta
4. Troca periódica da senha
5. Senhas armazenadas na forma de hashes criptográficos com salt
6. Bloqueio de conta por falhas consecutivas na submissão do 2FA
7. Segregação de funções para autorização de ações “críticas”
8. Medidas de segurança quanto ao desligamento de um colaborador
9. Monitoração de segurança e auditoria
10. Prevenção a ataques de sobrecarga contra APIs
11. Hardenização do arquivo local de resolução de nomes
12. Estação de trabalho hardenizada autônoma para gerenciamento

Módulo para validação e limpeza de arquivos (anexos) recebidos

Dentre tantos outros requisitos, agora demos chance do desenvolvedor entender o propósito deste requisito "Módulo para validação e limpeza de arquivos (anexos) recebidos". Agora o desenvolvedor sabe que o PDF que virá do "suposto cliente" pode ser um arquivo malicioso e portanto terá que desenvolver um módulo que "filtra" o conteúdo e exibe o mesmo de forma controlada e segura, como, por exemplo, transformando o PDF em um JPEG e exibindo seu conteúdo sem o risco de execução de um ActionScript. O Simples fato do desenvolvedor saber de que ele está se defendendo, aumenta em muito a eficácia da implementação da proteção.

Módulo para validação e limpeza de arquivos (anexos) recebidos

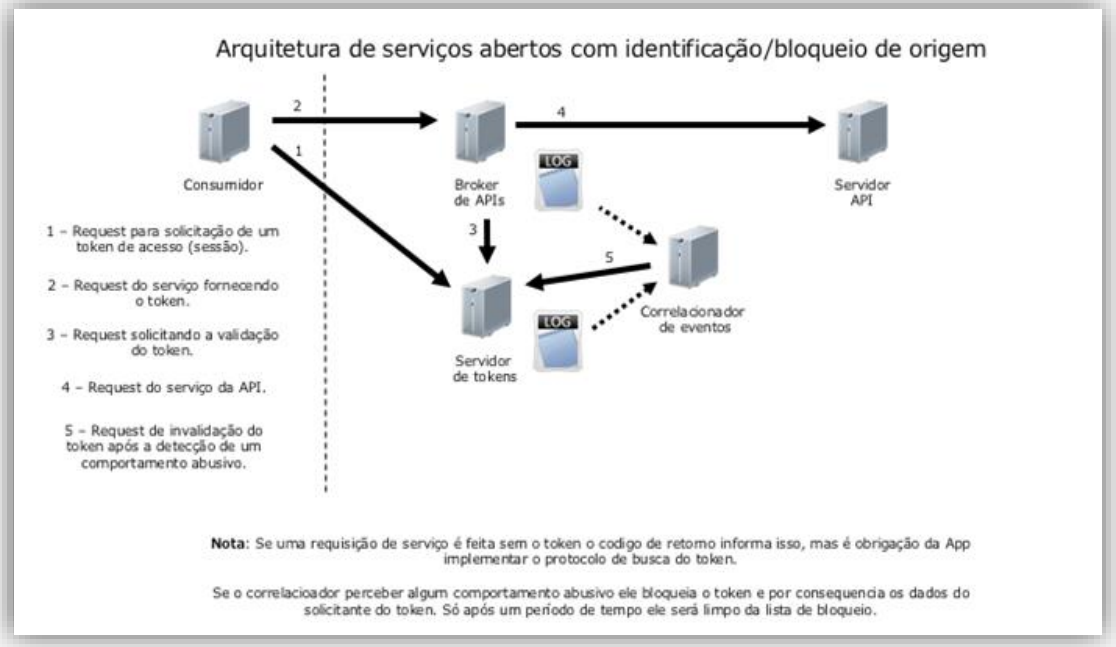
O requisito não versa sobre a simples limpeza, mas já prevê a validação da entrada do usuário para viabilizar a monitoração de origens de dados e contas maliciosas, garantindo uma certa "atenção especial" da equipe de monitoração e resposta a incidentes.

A famosa lista de requisitos

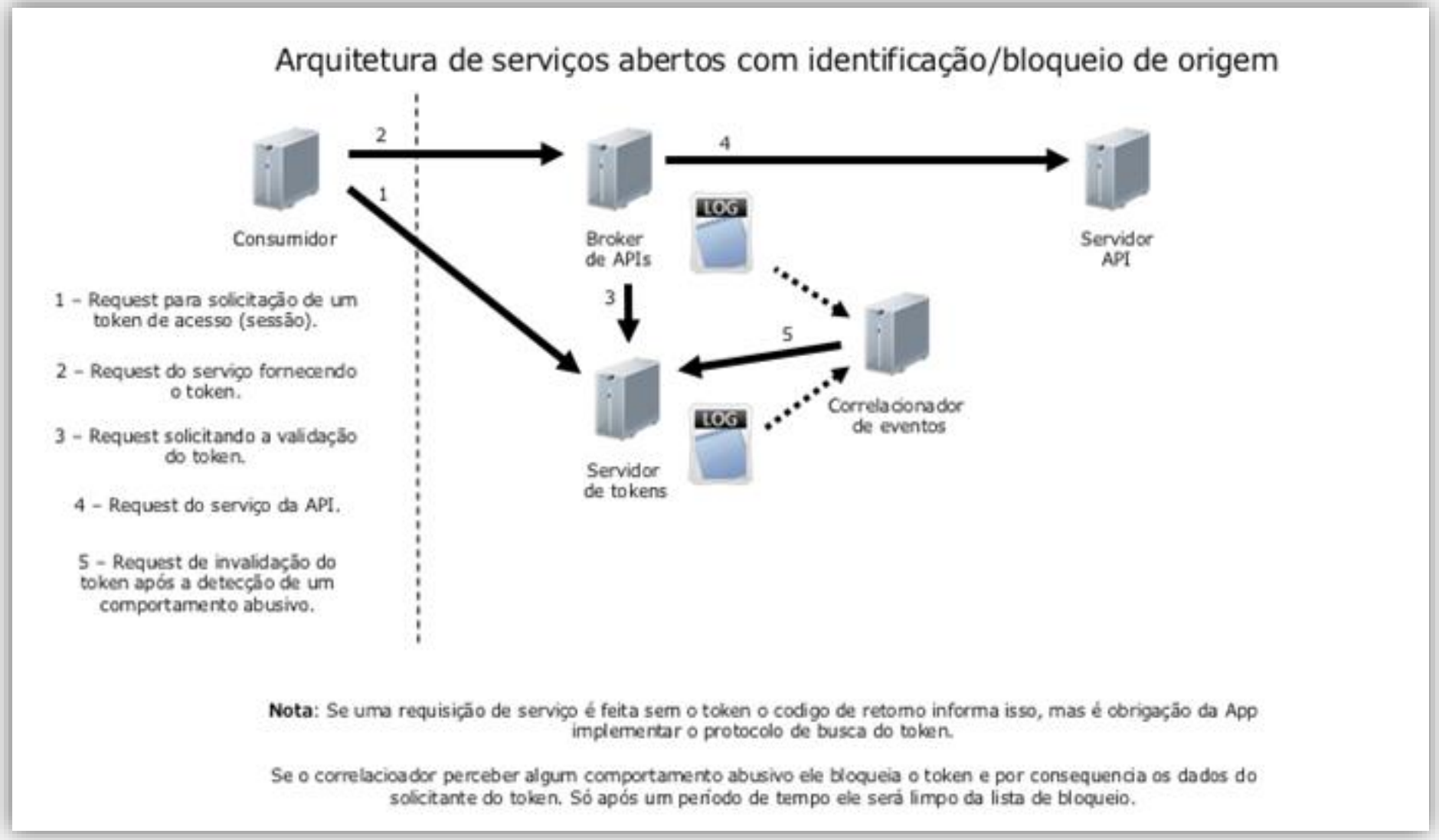
1. **Módulo para validação e limpeza de arquivos (anexos) recebidos**
2. **Prevenção a ataques de sobrecarga contra APIs**
3. Múltiplos fatores de autenticação
4. Política de senhas robusta
5. Troca periódica da senha
6. Senhas armazenadas na forma de hashes criptográficos com salt
7. Bloqueio de conta por falhas consecutivas na submissão do 2FA
8. Segregação de funções para autorização de ações “críticas”
9. Medidas de segurança quanto ao desligamento de um colaborador
10. Monitoração de segurança e auditoria
11. Hardenização do arquivo local de resolução de nomes
12. Estação de trabalho hardenizada autônoma para gerenciamento

Prevenção a ataques de sobrecarga contra APIs

Em outro exemplo, algumas necessidades derivadas do modelo de ameaças só poderão ser endereçadas com uma abordagem muito mais sofisticada exigindo mudanças e/ou complementações arquiteturais para o projeto.

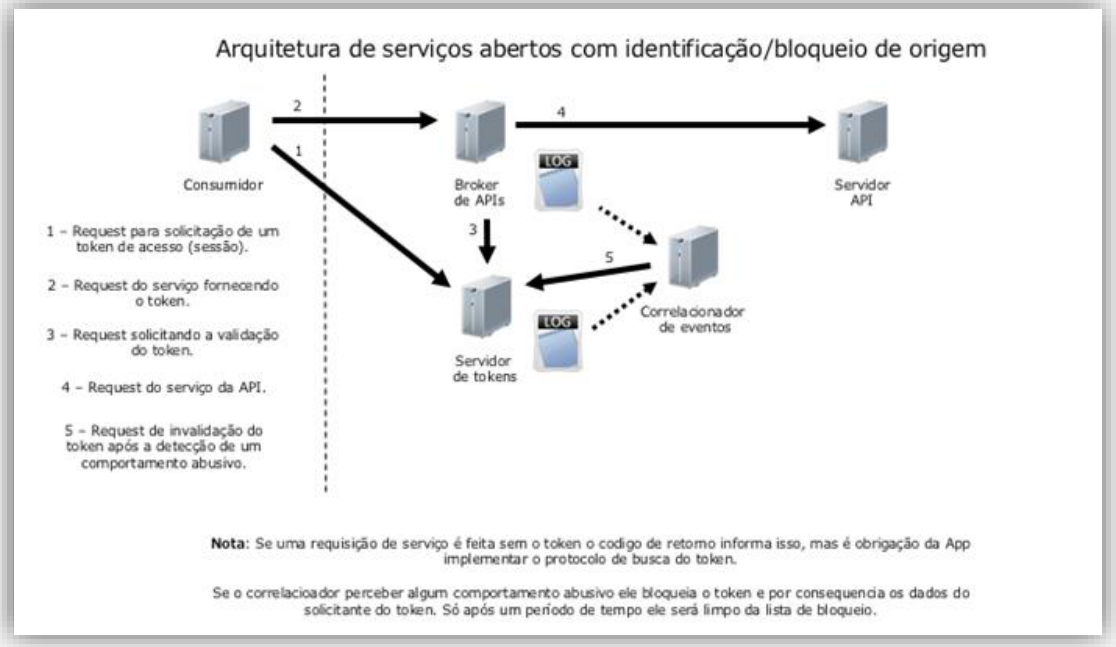


Prevenção a ataques de sobrecarga contra APIs



Prevenção a ataques de sobrecarga contra APIs

Eis o valor da modelagem para o time de arquitetura. Direcionar a arquitetura da aplicação para endereçar cenários de segurança viabilizando a monitoração e a resposta ao ataque. Note por exemplo que toda arquitetura desenhada teve como origem a necessidade de se proteger os entry points das APIs (Prevenção a ataques de sobrecarga contra APIs). A base da solução construída prevê a adição de certos elementos para marcação das origens e análise de seu comportamento.



Prevenção a ataques de sobrecarga contra APIs

Para a proteção total dos entry points das APIs, ainda há decisões arquiteturais relacionadas a existência de filas que coletem os pedidos "externos de fontes não confiáveis" e os coloque sob análise. Há um componente da arquitetura dedicado a remover solicitações maliciosas uma vez que a origem for marcada como maliciosa. Isso evita que o processamento de solicitações maliciosas onerem os serviços sem necessidade. Viabilizando desta forma uma resposta a incidente de sobrecarga quase 100% automatizada.

Prevenção a ataques de sobrecarga contra APIs