CURSO DE TECNOLOGIA EM DEFESA CIBERNÉTICA

Projeto Serasa - Feedback para clientes

SÃO PAULO – SP

<Maio/2024>

<Paulo Condutta Villas Boas - RM96662>

<Gabriel Mozelli Moreto – RM97522>

<Mariana Amorim – RM92078>

<Caio Ferreira - RM96354>

Fase 4 – Entrega final - POC

SÃO PAULO – SP

<Maio/2024>

SUMÁRIO

1. INTRODUÇÃO

Dando continuidade ao projeto de mitigação de riscos com a utilização de modelagem de ameaças, trazemos a conclusão do trabalho desenvolvido apresentando a finalização do projeto de modelagem de ameaças a SERASA EXPERIAN com a união das 4 fazes desenvolvidas durante os últimos 2 semestres.

Apresentamos os links para acesso aos dados do projeto assim como o video de apresentação, POC do projeto e repositório.

Link apresentação: https://www.youtube.com/watch?v=Mp6VFHPMMjA

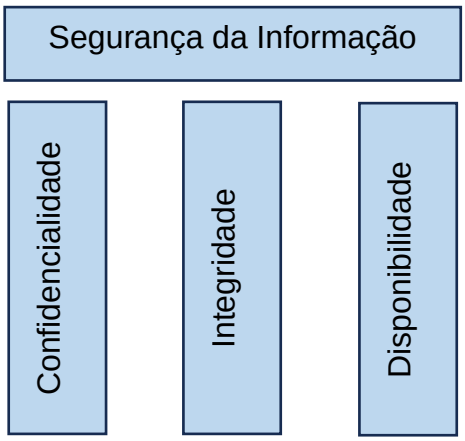
Link da POC: https://www.figma.com/proto/1tV09iLbYpTNmMQ0fTrTze/SERASA-CHALLENGE-USER-VIEW?node-id=2-1205&t=8HkOq2XtKtPht9U8-0&scaling=min-zoom&page-id=0%3A1

Link do repositório: https://github.com/GabrielMoreto/fiap_serasa

Agradecemos a oportunidade aberta pela SERASA EXPERIAN, pela FIAP e seus orientadores. Esperamos a evolução desse projeto e satisfação com entrega.

2. CONCEITO

Para estabelecer uma estrutura consistente de segurança, a modelagem de ameaças desempenha um papel fundamental. Atuando sobre os pilares da segurança da informação (TRIADE), quando relacionados à modelagem de ameaças, formam uma abordagem abrangente para proteger os sistemas, informações e usuários.



Confidencialidade

A confidencialidade concentra-se em garantir que os dados e informações sejam acessíveis apenas por agentes autorizados. A modelagem de ameaças é essencial para identificar ameaças à confidencialidade, como o acesso não autorizado a dados confidenciais.

Integridade

A integridade atua na proteção dos dados mantendo sua situação e propósitos confiáveis e íntegros. A modelagem de ameaças ajuda a identificar ameaças que podem comprometer a integridade como ataques de modificação ou corrompimento da informação.

Disponibilidade

A disponibilidade é a garantia de que os sistemas e informações estarão disponíveis para uso quando necessário. A modelagem de ameaças ajuda a identificar ameaças que podem levar à indisponibilidade, como ataques de negação de serviço (DDoS).

Ao relacionar a modelagem de ameaças aos pilares da segurança da informação, podemos desenvolver estratégias de segurança mais robustas, protegendo seus ativos e mantendo a integridade e a confiança nas informações. A segurança da informação é uma jornada contínua, e a modelagem de ameaças desempenha um papel central na manutenção da integridade e da segurança das informações em um mundo cada vez mais digital.

Adotada principalmente em um processo (até mesmo antes do início) do desenvolvimento seguro, onde desejamos trazer a segurança “mais à esquerda”, utilizar a modelagem possibilita reconhecer nossas fraquezas quando elas nem ao menos existem na realidade (estão ainda na ideia do Design).

A modelagem de ameaças faz parte do **Ciclo de Vida de Desenvolvimento de Segurança (SDLC) da Microsoft**. Ele identifica entidades do sistema, eventos e limites e, em seguida, aplica um conjunto de ameaças conhecidas. Usando-o, as equipes de segurança podem identificar ameaças potenciais. Essa característica a torna a prática de segurança mais eficaz que você pode aplicá-la. A imagem abaixo

demostra parte do SDLC focada na concepção, seguranca e design incluindo atividade de Threat Model.

Figure 1. Architecture task flow when a project is new or a redesign – Ibid, p. 278

Existem diferentes tipos de modelagem disponíveis a serem utilizadas, uma das mais comuns, criada nos anos 90's por dois pesquisadores de segurança da Microsoft Praerit Garg e Loren Kohnfelder, o **STRIDE** usa uma abordagem completa que pode auxiliar a identificar de forma contínua, sendo útil para detectar potencial problemas.

Propriedade	Ameaça	Definição
Autenticação	Spoofing	Personificar algo ou outra pessoa.
Integridade	Tampering	Modificar dados ou código
Não repúdio	Repudiação	Alegar não ter realizado uma ação.
Confidencialidade	Information disclosure	Expor informações a alguém não autorizado a vê-las
Disponibilidade	Denial of Service	Negar ou degradar o serviço aos usuários
Autorização	Elevation of Privilege	Obter recursos sem a autorização adequada

Tabela STRIDE com explicação de cada inicial de ameaça

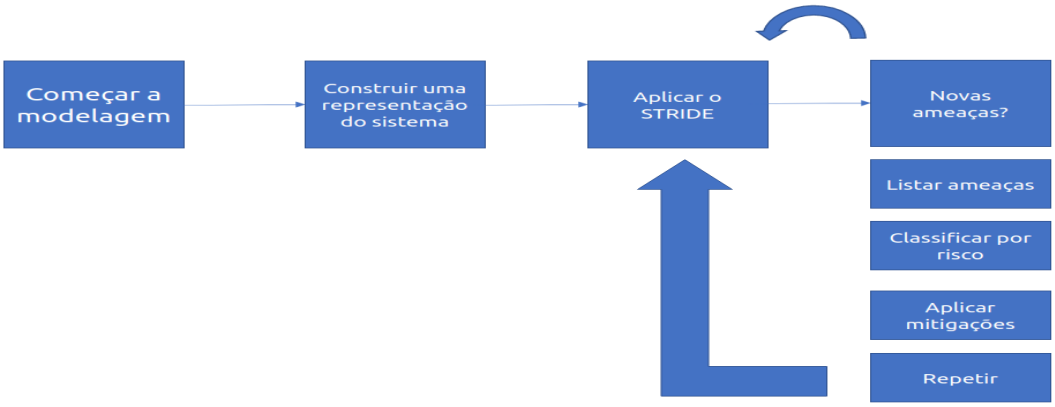
Todo desenvolvimento do trabalho será baseado nessa metodologia com auxílio da ferramenta **Microsoft Threat Modeling**, uma ferramenta sem custo que pode ser utilizada em conjunto pelos times de Desenvolvimento, de Cibersegurança ou Gestão de Riscos e Auditoria para criação do modelo de linha de defesa de 3 camadas.

3. OBJETIVOS

A segurança cibernética é uma preocupação essencial em ambientes web, especialmente em sistemas que lidam com informações sensíveis, como um ambiente web de feedback. Modelar ameaças é uma abordagem eficaz para identificar potenciais riscos de segurança e desenvolver estratégias de mitigação.

Alinhado aos desafios de segurança, a proposta tem que levar em consideração a **experiência do usuário (UX)** na navegação fluida e sem "bloqueios" que impactem a navegação do usuário e o objetivo principal da plataforma.

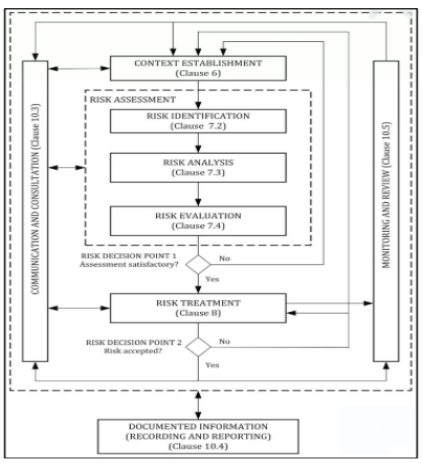
Nessa fase, avaliamos o cenário atual adotado e propomos ajustes nos fluxos e proteções para atingir o objetivo. Esse processo deve ser evoluído nas próximas fases além de ser contínuo e reaplicado a cada nova atualização tecnologia ou novo fluxo.



Pelas informações que temos até o presente momento, a abordagem recomendada de implantação da modelagem é sua introdução junto ao SDLC, colocando aqui o papel de Champion's de Segurança junto ao time de Desenvolvimento, que fará o papel de desempenhar o design e prototipagem do sistema.

O time de Cibersegurança fará o papel de alinhar as contramedidas e classificar o risco apresentando e baseando sobre análise de risco e avaliação seguindo normas e boas práticas da família ISO 27001, atualizada recentemente em 2022 a qual introduziu o ANEXO A 8.28 o processo de desenvolvimento seguro.

Adicionalmente, a modelagem de ameaças pode trazer um papel importante para a Gestão de Riscos (como informado anteriormente) utilizando uma análise de abordagem de riscos onde podemos expandir o uso para sua aplicação dentro da norma 27005:2022, pertencente à família ISO 27001



O uso da ISO 27005:2022 permite maior eficácia na avaliação e classificação de riscos direcionados a segurança da informação. Do contexto em que inserida a informação a ser protegida, inicia-se o processo de avaliação de riscos para que o tratamento ou aceitação dos riscos seja definido de acordo com o impacto e a probabilidade dos riscos avaliados.

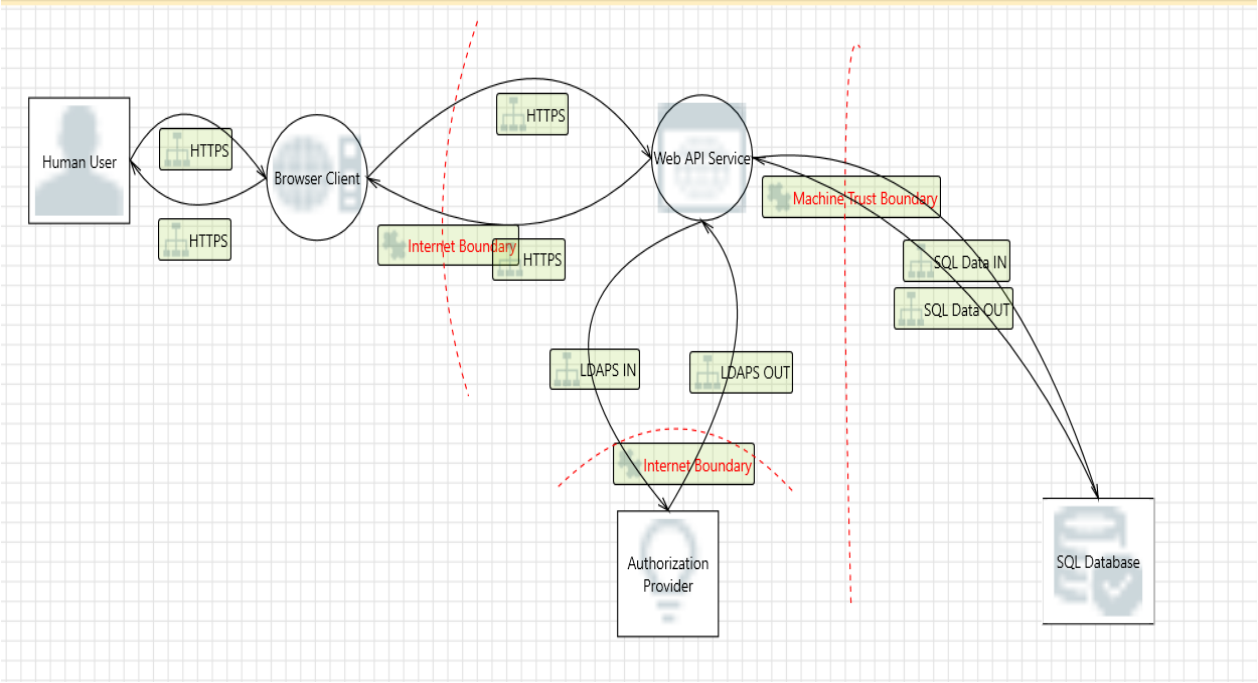
Deixamos claro que a modelagem pode ser de grande avalia para uso da SERASA EXPERIAN e fazer parte contínua dos processos e áreas que dão suporte a sustentação e visão sobre Riscos. Seu uso amplo depende de um estudo mais profundo e uma proposta de trabalho mais próxima ao Cliente, com informações detalhadas sobre capacidade técnica, assim como time disponível para atuação.

4. CENARIO ATUAL

FLUXO FUNCIONAL ATUAL

1. O usuário realiza o cadastro no sistema
2. O usuário faz o login com usuário e senha cadastrados
3. O usuário realiza o registro contendo seu feedback
4. O usuário faz o logoff

DFS ATUAL (PREVISTO)



Os diagramas de fluxo de dados (DFDs) representam graficamente o atual cenário existente, decompondo o sistema em partes e mostrando que cada parte é não suscetível a ameaças relevantes. Esse DFD apresentado traz algumas informações sugestivas uma vez que não temos informações detalhadas do atual cenário existente da aplicação na SERASA EXPIRIAN.

REPORT ATUAL

Este relatório apresenta o ATUAL report da análise de modelagem de ameaças STRIDE para um sistema web de feedback.

	Ameaça	Vulnerabilidade/ Risco	Contramedida/ Mitigação
Spoofing Falsificação	Spoofing of Destination Data Store SQL Database	SQL Database may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of SQL Database. Consider using a standard authentication mechanism to identify the destination data store.	Autenticação via MFA (token, mail link, sms) e monitoria
	Spoofing of Destination Data Store SQL Database	SQL Database may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of SQL Database. Consider using a standard authentication mechanism to identify the destination data store.	Autenticação via MFA (token, mail link, sms) e monitoria
	Spoofing the Human User External Entity	Human User may be spoofed by an attacker and this may lead to unauthorized access to Browser Client. Consider using a standard authentication mechanism to identify the external entity.	Autenticação via MFA (token, mail link, sms) e monitoria
	Spoofing of the Authorization Provider External Destination Entity	Authorization Provider may be spoofed by an attacker and this may lead to data being sent to the attacker's target instead of Authorization Provider. Consider using a standard authentication mechanism to identify the external entity.	Autenticação via MFA (token, mail link, sms) e monitoria
Tampering Violação	Risks from Logging	Log readers can come under attack via log files. Consider ways to canonicalize data in all logs. Implement a single reader for the logs, if possible, in order to reduce attack surface area. Be sure to understand and document log file elements which come from untrusted sources.	LOG centralizados via SIEM
	Potential SQL Injection Vulnerability for SQL Database	SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. Any procedure that constructs SQL statements should be reviewed for injection vulnerabilities because SQL Server will execute all syntactically valid queries that it receives. Even parameterized data can be manipulated by a skilled and determined attacker.	Filtros de requisição, controle de integridade com HASH e assinaturas dos dados, assinatura digital, controles de versão e backup
	Potential SQL Injection Vulnerability for SQL Database	SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. Any procedure that constructs SQL statements should be reviewed for injection vulnerabilities because SQL Server will execute all syntactically valid queries that it receives. Even parameterized data can be manipulated by a skilled and determined attacker.	Filtros de requisição, controle de integridade com HASH e assinaturas dos dados, assinatura digital, controles de versão e backup
	Risks from Logging	Log readers can come under attack via log files. Consider ways to canonicalize data in all logs. Implement a single reader for the logs, if possible, in order to reduce attack surface area. Be sure to understand and document log file elements which come from untrusted sources.	LOG centralizados via SIEM
	Browser Client Process Memory Tampered	If Browser Client is given access to memory, such as shared memory or pointers, or is given the ability to control what Web API Service executes (for example, passing back a function pointer.), then Browser Client can tamper with Web API Service. Consider if the function could work with less access to memory, such as passing data rather than pointers. Copy in data provided, and then validate it.	Controle memoria Browser
	Web API Service Process Memory Tampered	If Web API Service is given access to memory, such as shared memory or pointers, or is given the ability to control what Browser Client executes (for example, passing back a function pointer.), then Web API Service can tamper with Browser Client. Consider if the function could work with less access to memory, such as passing data rather than pointers. Copy in data provided, and then validate it.	Controle memoria e integridade WebAPI
	The SQL Database Data Store Could Be Corrupted	Data flowing across SQL Data IN may be tampered with by an attacker. This may lead to corruption of SQL Database. Ensure the integrity of the data flow to the data store.	Controle de integridade com HASH e assinaturas dos dados, assinatura digital, controles de versão e backup
	The SQL Database Data Store Could Be Corrupted	Data flowing across SQL Data OUT may be tampered with by an attacker. This may lead to corruption of SQL Database. Ensure the integrity of the data flow to the data store.	Controle de integridade com HASH e assinaturas dos dados, assinatura digital, controles de versão e backup
Repudiation Repúdio	Potential Weak Protections for Audit Data	Consider what happens when the audit mechanism comes under attack, including attempts to destroy the logs, or attack log analysis programs. Ensure access to the log is through a reference monitor, which controls read and write separately. Document what filters, if any, readers can rely on, or writers should expect	Logs detalhados e centralizados em SIEM e rotatividade de LOGs

	Insufficient Auditing	Does the log capture enough data to understand what happened in the past? Do your logs capture enough data to understand an incident after the fact? Is such capture lightweight enough to be left on all the time? Do you have enough data to deal with repudiation claims? Make sure you log sufficient and appropriate data to handle a repudiation claims. You might want to talk to an audit expert as well as a privacy expert about your choice of data.	Registrar detalhadamente todas as interações dos usuários, incluindo envios de feedback, para fins de auditoria.
	Data Logs from an Unknown Source	Do you accept logs from unknown or weakly authenticated users or systems? Identify and authenticate the source of the logs before accepting them.	Logs detalhados e centralizados em SIEM
	Lower Trusted Subject Updates Logs	If you have trust levels, is anyone other outside of the highest trust level allowed to log? Letting everyone write to your logs can lead to repudiation problems. Only allow trusted code to log.	Logs detalhados e centralizados em SIEM
	Lower Trusted Subject Updates Logs	If you have trust levels, is anyone other outside of the highest trust level allowed to log? Letting everyone write to your logs can lead to repudiation problems. Only allow trusted code to log.	Logs detalhados e centralizados em SIEM
	Data Logs from an Unknown Source	Do you accept logs from unknown or weakly authenticated users or systems? Identify and authenticate the source of the logs before accepting them.	Autenticação forte pra envio de LOGS
	Insufficient Auditing	Does the log capture enough data to understand what happened in the past? Do your logs capture enough data to understand an incident after the fact? Is such capture lightweight enough to be left on all the time? Do you have enough data to deal with repudiation claims? Make sure you log sufficient and appropriate data to handle a repudiation claims. You might want to talk to an audit expert as well as a privacy expert about your choice of data.	Registrar detalhadamente todas as interações dos usuários, incluindo envios de feedback, para fins de auditoria.
	Potential Weak Protections for Audit Data	Consider what happens when the audit mechanism comes under attack, including attempts to destroy the logs, or attack log analysis programs. Ensure access to the log is through a reference monitor, which controls read and write separately. Document what filters, if any, readers can rely on, or writers should expect	Autenticação forte e rotatividade de Logs
	Potential Data Repudiation by Web API Service	Web API Service claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.	Monitoramento
	Potential Data Repudiation by Browser Client	Browser Client claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.	N/A
	Potential Data Repudiation by Web API Service	Web API Service claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.	Monitoramento
	External Entity Authorization Provider Potentially Denies Receiving Data	Authorization Provider claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.	Monitoramento e Controle de chamadas
	Data Store Denies SQL Database Potentially Writing Data	SQL Database claims that it did not write data received from an entity on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.	Monitoramento e Controle de chamadas
	Data Store Denies SQL Database Potentially Writing Data	SQL Database claims that it did not write data received from an entity on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.	Monitoramento e Controle de chamadas
Information Disclosure **Divulgação de Informações**	Authorization Bypass	Can you access SQL Database and bypass the permissions for the object? For example by editing the files directly with a hex editor, or reaching it via filesharing? Ensure that your program is the only one that can access the data, and that all other subjects have to use your interface.	Implementar autenticação forte, criptografia para proteger informações sensíveis e garantir que apenas usuários autorizados tenham acesso.
	Weak Credential Storage	Credentials held at the server are often disclosed or tampered with and credentials stored on the client are often stolen. For server side, consider storing a salted hash of the credentials instead of storing the credentials themselves. If this is not possible due to business requirements, be sure to encrypt the credentials before storage, using an SDL-approved mechanism. For client side, if storing credentials is required, encrypt them and protect the data store in which they're stored	Implementar autenticação forte, criptografia para proteger informações sensíveis e garantir que apenas usuários autorizados tenham acesso.
	Weak Credential Storage	Credentials held at the server are often disclosed or tampered with and credentials stored on the client are often stolen. For server side, consider storing a salted hash of the credentials instead of storing the credentials themselves. If this is not possible due to business requirements, be sure to encrypt the credentials before storage, using an SDL-approved mechanism. For client side, if storing credentials is required, encrypt them and protect the data store in which they're stored	Implementar autenticação forte, criptografia para proteger informações sensíveis e garantir que apenas usuários autorizados tenham acesso.
	Authorization Bypass	Can you access SQL Database and bypass the permissions for the object? For example by editing the files directly with a hex editor, or reaching it via filesharing? Ensure that your program is the only one that can access the data, and that all other subjects have to use your interface.	Implementar autenticação forte, criptografia para proteger informações sensíveis e garantir que apenas usuários autorizados tenham acesso.
	Weak Authentication Scheme	Custom authentication schemes are susceptible to common weaknesses such as weak credential change management, credential equivalence, easily guessable credentials, null credentials, downgrade authentication or a weak credential change management system. Consider the impact and potential mitigations for your custom authentication scheme.	Implementar autenticação forte, criptografia para proteger informações sensíveis e garantir que apenas usuários autorizados tenham acesso.
	Weak Credential Transit	Credentials on the wire are often subject to sniffing by an attacker. Are the credentials re-usable/re-playable? Are credentials included in a message? For example, sending a zip file with the password in the email. Use strong cryptography for the transmission of credentials. Use the OS libraries if at all possible, and consider cryptographic algorithm agility, rather than hardcoding a choice.	Implementar autenticação forte, criptografia para proteger informações sensíveis e garantir que apenas usuários autorizados tenham acesso.

	Weak Credential Transit	Credentials on the wire are often subject to sniffing by an attacker. Are the credentials re-usable/re-playable? Are credentials included in a message? For example, sending a zip file with the password in the email. Use strong cryptography for the transmission of credentials. Use the OS libraries if at all possible, and consider cryptographic algorithm agility, rather than hardcoding a choice.	Implementar autenticação forte, criptografia para proteger informações sensíveis e garantir que apenas usuários autorizados tenham acesso.
Denial of Service **Negação de Serviço**	Potential Excessive Resource Consumption for Web API Service or SQL Database	Does Web API Service or SQL Database take explicit steps to control resource consumption? Resource consumption attacks can be hard to deal with, and there are times that it makes sense to let the OS do the job. Be careful that your resource requests don't deadlock, and that they do timeout.	Controles e limites de requisição/segundo
	Potential Excessive Resource Consumption for Web API Service or SQL Database	Does Web API Service or SQL Database take explicit steps to control resource consumption? Resource consumption attacks can be hard to deal with, and there are times that it makes sense to let the OS do the job. Be careful that your resource requests don't deadlock, and that they do timeout.	Controles e limites de requisição/segundo
	Data Flow HTTPS Is Potentially Interrupted	An external agent interrupts data flowing across a trust boundary in either direction.	Controles via Firewall de limites de requisição ip / requisições por segundo
	Potential Process Crash or Stop for Web API Service	Web API Service crashes, halts, stops or runs slowly; in all cases violating an availability metric.	Controles e limites de requisição/segundo
	Data Flow HTTPS Is Potentially Interrupted	An external agent interrupts data flowing across a trust boundary in either direction.	Controles e limites de requisição/segundo
	Potential Process Crash or Stop for Browser Client	Browser Client crashes, halts, stops or runs slowly; in all cases violating an availability metric.	Controles de Memoria
	Data Flow LDAPS OUT Is Potentially Interrupted	An external agent interrupts data flowing across a trust boundary in either direction.	Controles e limites de requisição/segundo
	Potential Process Crash or Stop for Web API Service	Web API Service crashes, halts, stops or runs slowly; in all cases violating an availability metric.	Servidores escaláveis em Alta Disponibilidade
	Data Flow LDAPS IN Is Potentially Interrupted	An external agent interrupts data flowing across a trust boundary in either direction.	Controles e limites de requisição/segundo
	Data Store Inaccessible	An external agent prevents access to a data store on the other side of the trust boundary.	Bancos em Alta Disponibilidade
	Data Flow SQL Data IN Is Potentially Interrupted	An external agent interrupts data flowing across a trust boundary in either direction.	Controles e limites de requisição/segundo
	Data Store Inaccessible	An external agent prevents access to a data store on the other side of the trust boundary.	Bancos em Alta Disponibilidade
	Data Flow SQL Data OUT Is Potentially Interrupted	An external agent interrupts data flowing across a trust boundary in either direction.	Controles e limites de requisição/segundo
	Potential Excessive Resource Consumption for Web API Service or SQL Database	Does Web API Service or SQL Database take explicit steps to control resource consumption? Resource consumption attacks can be hard to deal with, and there are times that it makes sense to let the OS do the job. Be careful that your resource requests don't deadlock, and that they do timeout.	Controles e limites de requisição/segundo
Elevation of Privilege **Elevação de Privilégio**	Elevation Using Impersonation	Browser Client may be able to impersonate the context of Human User in order to gain additional privilege.	Implementar o princípio do menor privilégio, onde os usuários têm apenas o acesso necessário para realizar suas tarefas e monitorar atividades suspeitas.
	Elevation Using Impersonation	Web API Service may be able to impersonate the context of Authorization Provider in order to gain additional privilege.	Implementar o princípio do menor privilégio, onde os usuários têm apenas o acesso necessário para realizar suas tarefas e monitorar atividades suspeitas.

	Elevation Using Impersonation	Web API Service may be able to impersonate the context of Browser Client in order to gain additional privilege.	Implementar o princípio do menor privilégio, onde os usuários têm apenas o acesso necessário para realizar suas tarefas e monitorar atividades suspeitas.
	Elevation Using Impersonation	Browser Client may be able to impersonate the context of Web API Service in order to gain additional privilege.	Implementar o princípio do menor privilégio, onde os usuários têm apenas o acesso necessário para realizar suas tarefas e monitorar atividades suspeitas.
	Elevation by Changing the Execution Flow in Web API Service	An attacker may pass data into Web API Service in order to change the flow of program execution within Web API Service to the attacker's choosing.	Implementar o princípio do menor privilégio, onde os usuários têm apenas o acesso necessário para realizar suas tarefas e monitorar atividades suspeitas.
	Web API Service May be Subject to Elevation of Privilege Using Remote Code Execution	Browser Client may be able to remotely execute code for Web API Service.	Implementar o princípio do menor privilégio, onde os usuários têm apenas o acesso necessário para realizar suas tarefas e monitorar atividades suspeitas.
	Browser Client May be Subject to Elevation of Privilege Using Remote Code Execution	Web API Service may be able to remotely execute code for Browser Client.	Implementar o princípio do menor privilégio, onde os usuários têm apenas o acesso necessário para realizar suas tarefas e monitorar atividades suspeitas.
	Elevation by Changing the Execution Flow in Browser Client	An attacker may pass data into Browser Client in order to change the flow of program execution within Browser Client to the attacker's choosing.	Implementar o princípio do menor privilégio, onde os usuários têm apenas o acesso necessário para realizar suas tarefas e monitorar atividades suspeitas.
	Elevation by Changing the Execution Flow in Web API Service	An attacker may pass data into Web API Service in order to change the flow of program execution within Web API Service to the attacker's choosing.	Implementar o princípio do menor privilégio, onde os usuários têm apenas o acesso necessário para realizar suas tarefas e monitorar atividades suspeitas.
	Web API Service May be Subject to Elevation of Privilege Using Remote Code Execution	Authorization Provider may be able to remotely execute code for Web API Service.	Implementar o princípio do menor privilégio, onde os usuários têm apenas o acesso necessário para realizar suas tarefas e monitorar atividades suspeitas.

Podemos avaliar de forma mais detalhada o atual cenário nos anexos abaixo, o relatório apresentado pela ferramenta Microsoft Threat Modeling Tool onde exportamos as ameaças identificadas, além do Report completo da ferramenta.


report-ameacas-atual .xlsx


report STRIDE.htm

Reforçamos que a análise foi efetuada com informações disponíveis, em um cenário limitado, gerando assim uma ideia de como o atual ambiente está desenhado e como podemos propor melhorias que apresentaremos mais à frente.

5. PROPOSTA

Propomos que a abordagem de proteção cibernética seja um esforço contínuo, e a análise de ameaças deve ser atualizada regularmente para acompanhar as mudanças no ambiente de ameaças e novas vulnerabilidades que possam surgir. Nessa fase, avaliamos o fluxo atual e iniciamos ajustes de fluxo funcional, fluxo da aplicação, e

proteções baseadas em boas práticas. Adicionamos ações de mitigação para cada processo e um fluxo funcional mais coeso com boas práticas propostas.

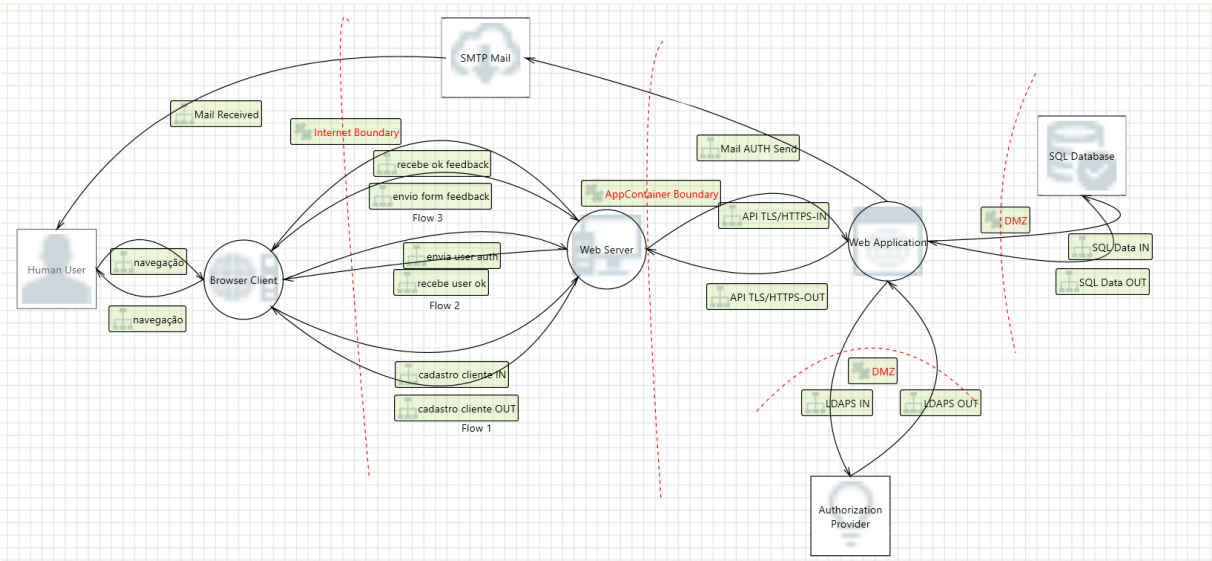
NOVO FLUXO FUNCIONAL PROPOSTO – FASE 2

1. O usuário realiza o cadastro no sistema
2. Sistema dispara um e-mail para usuário com mail já cadastrado na base
3. O usuário faz o login com usuário e senha cadastrados
4. O usuário realiza o registro contendo seu feedback
5. O usuário faz o logoff

Esse novo fluxo traz uma visão de acesso somente a usuários já devidamente existentes na base de clientes da SERASA EXPERIAN. Entendemos que existem dados mínimos na base e propomos a implantação desse controle para a abertura de feedback de forma controlada.

Adicionalmente, apontamos a necessidade de criação de canal de ouvidoria para reclamações de usuários que não detém cadastro na base ou não sejam clientes diretamente atendidos. Esse canal de ouvidoria também será uma contramedida para casos de indisponibilidade da plataforma e avaliação de casos não atendidos ou não possíveis de serem atendidos pelo canal de feedback.

NOVO DFS PROPOSTO – FASE 2



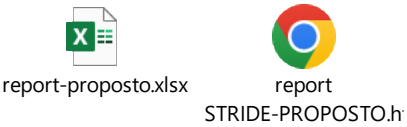
O DFD apresentado aqui traz uma evolução do cenário apresentado inicialmente. Ações e contramedidas estão apresentadas no relatório abaixo com uma análise mais completa.

	Ameaça	Vulnerabilidade/ Risco	Contramedida/ Mitigação
Spoofing **Falsificação**	Spoofing of the External Web Service External Destination Entity	External Web Service may be spoofed by an attacker and this may lead to data being sent to the attacker's target instead of External Web Service. Consider using a standard authentication mechanism to identify the external entity.	Autenticação via MFA (mail link, sms) e monitoria
	Spoofing of the External Web Service External Destination Entity	External Web Service may be spoofed by an attacker and this may lead to data being sent to the attacker's target instead of External Web Service. Consider using a standard authentication mechanism to identify the external entity.	Autenticação via MFA (mail link, sms) e monitoria
	Spoofing the Browser Client Process	Browser Client may be spoofed by an attacker and this may lead to information disclosure by External Web Service. Consider using a standard authentication mechanism to identify the destination process.	Autenticação via MFA (mail link, sms) e monitoria
	Spoofing of the External Web Service External Destination Entity	External Web Service may be spoofed by an attacker and this may lead to data being sent to the attacker's target instead of External Web Service. Consider using a standard authentication mechanism to identify the external entity.	Autenticação via MFA (mail link, sms) e monitoria
	Spoofing of Destination Data Store SMTP Mail	SMTP Mail may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of SMTP Mail. Consider using a standard authentication mechanism to identify the destination data store.	Limite de acesso ao envio SMTP e autenticação
	Spoofing of Destination Data Store SQL Database	SQL Database may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of SQL Database. Consider using a standard authentication mechanism to identify the destination data store.	Autenticação forte
	Spoofing of Source Data Store SMTP Mail	SMTP Mail may be spoofed by an attacker and this may lead to incorrect data delivered to Human User. Consider using a standard authentication mechanism to identify the source data store.	Limite de acesso ao envio SMTP e autenticação
	Spoofing of Source Data Store SQL Database	SQL Database may be spoofed by an attacker and this may lead to incorrect data delivered to External Web Application. Consider using a standard authentication mechanism to identify the source data store.	Autenticação forte
	Spoofing of Destination Data Store SMTP Mail	SMTP Mail may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of SMTP Mail. Consider using a standard authentication mechanism to identify the destination data store.	Limite de acesso ao envio SMTP e autenticação
Tampering **Violação**	Authenticated Data Flow Compromised	An attacker can read or modify data transmitted over an authenticated dataflow.	Uso SSL/TLS
	The SQL Database Data Store Could Be Corrupted	Data flowing across SQL Data IN may be tampered with by an attacker. This may lead to corruption of SQL Database. Ensure the integrity of the data flow to the data store.	Autenticação forte e limite acesso
	Possible SQL Injection Vulnerability for SQL Database	SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. Any procedure that constructs SQL statements should be reviewed for injection vulnerabilities because SQL Server will execute all syntactically valid queries that it receives. Even parameterized data can be manipulated by a skilled and determined attacker.	Filtros de requisição, controle de integridade com HASH e assinaturas dos dados, assinatura digital, controles de versão e backup
	The SMTP Mail Data Store Could Be Corrupted	Data flowing across Mail AUTH Send may be tampered with by an attacker. This may lead to corruption of SMTP Mail. Ensure the integrity of the data flow to the data store.	Limite de acesso ao envio SMTP e autenticação
Repudiation **Repúdio**	External Entity External Web Service Potentially Denies Receiving Data	External Web Service claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.	Controles e limites de requisição/segundo, LOG centralizados via SIEM
	External Entity External Web Service Potentially Denies Receiving Data	External Web Service claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.	Controles e limites de requisição/segundo, LOG centralizados via SIEM
	External Entity External Web Service Potentially Denies Receiving Data	External Web Service claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.	Controles e limites de requisição/segundo, LOG centralizados via SIEM
	External Entity External Web Application Potentially Denies Receiving Data	External Web Application claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.	Controles e limites de requisição/segundo, LOG centralizados via SIEM
	Potential Data Repudiation by Browser Client	Browser Client claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.	N/A
	Potential Data Repudiation by Browser Client	Browser Client claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.	N/A
	External Entity External Web Service Potentially Denies Receiving Data	External Web Service claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.	Controles e limites de requisição/segundo, LOG centralizados via SIEM
	Potential Data Repudiation by Browser Client	Browser Client claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.	N/A
	Data Store Denies SQL Database Potentially Writing Data	SQL Database claims that it did not write data received from an entity on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.	Monitoramento e Controle de chamadas
	Data Logs from an Unknown Source	Do you accept logs from unknown or weakly authenticated users or systems? Identify and authenticate the source of the logs before accepting them.	Logs detalhados e centralizados em SIEM
	Lower Trusted Subject Updates Logs	If you have trust levels, is anyone other outside of the highest trust level allowed to log? Letting everyone write to your logs can lead to repudiation problems. Only allow trusted code to log.	Logs detalhados e centralizados em SIEM
	External Entity Authorization Provider Potentially Denies Receiving Data	Authorization Provider claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.	Controles e limites de requisição/segundo, LOG centralizados via SIEM
	External Entity Human User Potentially Denies Receiving Data	Human User claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.	Controles e limites de requisição/segundo, LOG centralizados via SIEM
	External Entity External Web Application Potentially Denies Receiving Data	External Web Application claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.	Controles e limites de requisição/segundo, LOG centralizados via SIEM

	External Entity External Web Application Potentially Denies Receiving Data	External Web Application claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.	Controles e limites de requisição/segundo, LOG centralizados via SIEM
	Data Store Denies SMTP Mail Potentially Writing Data	SMTP Mail claims that it did not write data received from an entity on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.	Limite de acesso ao envio SMTP e autenticação
	Data Logs from an Unknown Source	Do you accept logs from unknown or weakly authenticated users or systems? Identify and authenticate the source of the logs before accepting them.	Controles e limites de requisição/segundo, LOG centralizados via SIEM
	Lower Trusted Subject Updates Logs	If you have trust levels, is anyone other outside of the highest trust level allowed to log? Letting everyone write to your logs can lead to repudiation problems. Only allow trusted code to log.	Logs detalhados e centralizados em SIEM
Information Disclosure **Divulgação de Informações**	Weak Access Control for a Resource	Improper data protection of SMTP Mail can allow an attacker to read information not intended for disclosure. Review authorization settings.	Limite de acesso ao envio SMTP e autenticação
	Weak Access Control for a Resource	Improper data protection of SQL Database can allow an attacker to read information not intended for disclosure. Review authorization settings.	Limite de acesso ao envio SMTP e autenticação
Denial of Service **Negação de Serviço**	Data Flow envio form feedback Is Potentially Interrupted	Browser Client crashes, halts, stops or runs slowly; in all cases violating an availability metric.	N/A
	Data Flow envia user auth Is Potentially Interrupted	An external agent interrupts data flowing across a trust boundary in either direction.	Controles e limites de requisição/segundo
	Data Flow API TLS/HTTPS-OUT Is Potentially Interrupted	An external agent prevents access to a data store on the other side of the trust boundary.	Controles e limites de requisição/segundo
	Data Flow API TLS/HTTPS-IN Is Potentially Interrupted	An external agent interrupts data flowing across a trust boundary in either direction.	Controles e limites de requisição/segundo
	Data Flow cadastro cliente OUT Is Potentially Interrupted	An external agent prevents access to a data store on the other side of the Controles e limites de requisição/segundo trust boundary.	Controles e limites de requisição/segundo
	Potential Process Crash or Stop for Browser Client	An external agent interrupts data flowing across a trust boundary in either direction.	Controles de Memoria
	Data Flow recebe user ok Is Potentially Interrupted	An external agent interrupts data flowing across a trust boundary in either direction.	Controles e limites de requisição/segundo
	Potential Process Crash or Stop for Browser Client	An external agent prevents access to a data store on the other side of the trust boundary.	Controles de Memoria
	Data Flow cadastro cliente IN Is Potentially Interrupted	An external agent prevents access to a data store on the other side of the trust boundary.	Controles e limites de requisição/segundo
	Potential Process Crash or Stop for Browser Client	An external agent interrupts data flowing across a trust boundary in either direction.	Controles de Memoria
	Data Flow recebe ok feedback Is Potentially Interrupted	An external agent interrupts data flowing across a trust boundary in either direction.	Controles e limites de requisição/segundo
	Data Store Inaccessible	An external agent interrupts data flowing across a trust boundary in either direction.	Bancos em Alta Disponibilidade
	Data Flow Mail Received Is Potentially Interrupted	An external agent prevents access to a data store on the other side of the trust boundary.	Controles e limites de requisição/segundo
	Data Store Inaccessible	An external agent interrupts data flowing across a trust boundary in either direction.	Bancos em Alta Disponibilidade
	Data Flow SQL Data IN Is Potentially Interrupted	Browser Client crashes, halts, stops or runs slowly; in all cases violating an availability metric.	Controles e limites de requisição/segundo
	Data Flow LDAPS IN Is Potentially Interrupted	An external agent interrupts data flowing across a trust boundary in either direction.	Controles e limites de requisição/segundo
	Data Store Inaccessible	An external agent prevents access to a data store on the other side of the trust boundary.	Bancos em Alta Disponibilidade
	Data Store Inaccessible	An external agent interrupts data flowing across a trust boundary in either direction.	Bancos em Alta Disponibilidade
	Data Flow Mail AUTH Send Is Potentially Interrupted	An external agent prevents access to a data store on the other side of the trust boundary.	Controles e limites de requisição/segundo
	Data Flow SQL Data OUT Is Potentially Interrupted	An external agent interrupts data flowing across a trust boundary in either direction.	Controles e limites de requisição/segundo
	Data Flow LDAPS OUT Is Potentially Interrupted	An external agent interrupts data flowing across a trust boundary in either direction.	Controles e limites de requisição/segundo
	Data Store Inaccessible	An external agent prevents access to a data store on the other side of the trust boundary.	Bancos em Alta Disponibilidade
	Data Flow Mail AUTH Send Is Potentially Interrupted	An external agent prevents access to a data store on the other side of the trust boundary.	Controles e limites de requisição/segundo
Elevation of Privilege **Elevação de Privilégio**	Elevation by Changing the Execution Flow in Browser Client	An attacker may pass data into Browser Client in order to change the flow of program execution within Browser Client to the attacker's choosing.	Implementar o princípio do menor privilégio, onde os usuários têm apenas o acesso necessário para realizar suas tarefas e monitorar atividades suspeitas.
	Browser Client May be Subject to Elevation of Privilege Using Remote Code Execution	External Web Service may be able to remotely execute code for Browser Client.	Implementar o princípio do menor privilégio, onde os usuários têm apenas o acesso necessário para realizar suas tarefas e monitorar atividades suspeitas.
	Elevation Using Impersonation	Browser Client may be able to impersonate the context of External Web Service in order to gain additional privilege.	Implementar o princípio do menor privilégio, onde os usuários têm apenas o acesso necessário para realizar suas tarefas e monitorar atividades suspeitas.
	Elevation by Changing the Execution Flow in Browser Client	An attacker may pass data into Browser Client in order to change the flow of program execution within Browser Client to the attacker's choosing.	Implementar o princípio do menor privilégio, onde os usuários têm apenas o acesso necessário para realizar suas tarefas e monitorar atividades suspeitas.
	Browser Client May be Subject to Elevation of Privilege Using Remote Code Execution	External Web Service may be able to remotely execute code for Browser Client.	Implementar o princípio do menor privilégio, onde os usuários têm apenas o acesso necessário para realizar suas tarefas e monitorar atividades suspeitas.
	Elevation Using Impersonation	Browser Client may be able to impersonate the context of External Web Service in order to gain additional privilege.	Implementar o princípio do menor privilégio, onde os usuários têm apenas o acesso necessário para realizar suas tarefas e monitorar atividades suspeitas.

	Elevation Using Impersonation	Browser Client may be able to impersonate the context of External Web Service in order to gain additional privilege.	Implementar o princípio do menor privilégio, onde os usuários têm apenas o acesso necessário para realizar suas tarefas e monitorar atividades suspeitas.
	Browser Client May be Subject to Elevation of Privilege Using Remote Code Execution	External Web Service may be able to remotely execute code for Browser Client.	Implementar o princípio do menor privilégio, onde os usuários têm apenas o acesso necessário para realizar suas tarefas e monitorar atividades suspeitas.
	Elevation by Changing the Execution Flow in Browser Client	An attacker may pass data into Browser Client in order to change the flow of program execution within Browser Client to the attacker's choosing.	Implementar o princípio do menor privilégio, onde os usuários têm apenas o acesso necessário para realizar suas tarefas e monitorar atividades suspeitas.

Podemos avaliar de forma mais detalhada o cenário proposto nos anexos apresentados pela ferramenta Microsoft Threat Modeling Tool, onde exportamos as ameaças identificadas, além do Report completo da ferramenta.



Além disso, propomos trazer na próxima entrega um mapeamento de risco associado a cada fluxo existente que possibilite uma atuação mais direcionada do time. Para isso, precisamos entender de forma mais próxima ao Cliente, motivações de ataque, o objetivo, e o impacto que pode ser causado por uma exploração bem-sucedida.

Toda evolução do trabalho depende de uma atuação mais próxima a SERASA EXPERIAN para entender o contexto geral assim como maiores detalhes sobre.

6. DEMOSNTRAÇÃO DE VULNERABILIDADES E CONTROLES AJUSTADOS – FASE 3

Com base na modelagem executada nas fases anteriores do projeto, demonstramos aqui a execução de um ataque e como ele foi mitigado detalhando medidas para deixar nossa aplicação mais segura.

O ataque referenciado: SQL injection

É um tipo de ataque, que se baseia na execução de comandos SQL, inseridos via comandos de manipulação ou comandos de definição de dados.

Esse tipo de ataque está entre os dez mais comuns e se bem executado pode permitir o atacante ler, inserir e modificar dados sensíveis do banco de dados,

executar operações de administração (como desligamento) de um banco de dados, recuperar conteúdo de um determinado arquivo e, em alguns casos, enviar comandos para o sistema operacional.

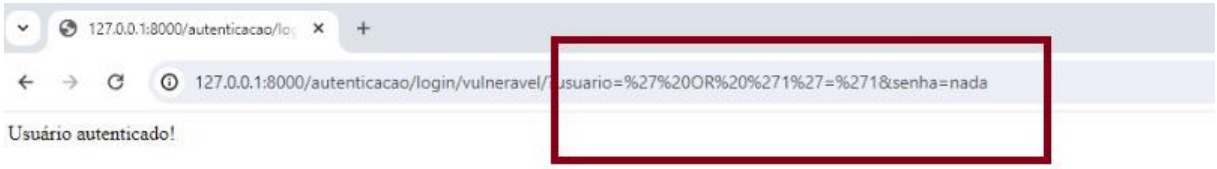


Figura 1 - Ataque bem-sucedido

Como é possível observar na imagem acima, o ataque foi utilizado oferecendo como usuário “OR ‘1’ = ‘1’. Por esse comando sempre retornar verdadeiro, a autenticação é realizada e o atacante consegue acesso a página do cliente.

A mitigação:

Para prevenir contra esse ataque, optamos como já mencionado, utilizar o framework Django que por padrão já possui diversas medidas segurança, como por exemplo: proteções contra falsificação de solicitação entre sites (CSRF), clickjacking, e outras vulnerabilidades comuns.

Também utilizamos parâmetros e consulta ao construir as consultas SQL, ao invés de concatenar os valores. Permitindo que os valores sejam corretamente escapados e evitando assim a possibilidade de injeção SQL.

Outra medida de segurança para mitigar evento, foi optar por salvar as senhas utilizando criptografia, como é possível ver na imagem abaixo:

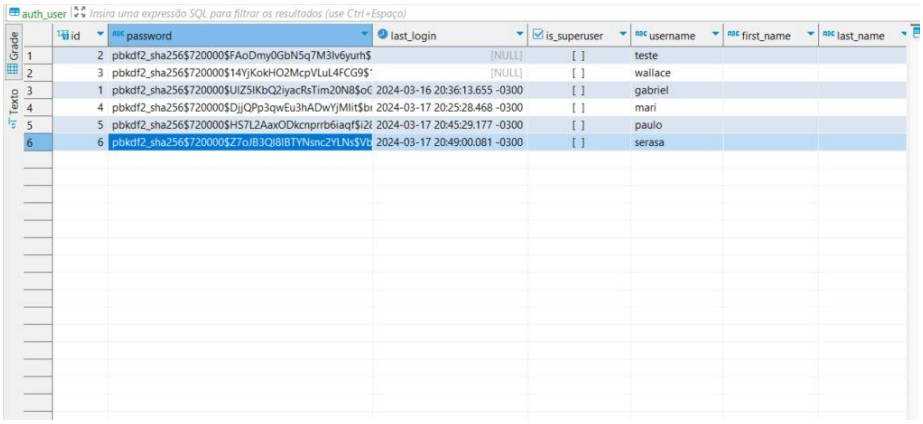


Figura 2 - Senhas criptografadas

Essas decisões tomadas com base na modelagem e aplicadas no desenvolvimento de nossa aplicação, resultam em um ataque mal-sucedido.

Figura 3 - Ataque mal-sucedido

Para a Fase 4 do projeto, separamos outros tipos de ataques e mitigações foram executadas e estão disponíveis no video detalhado mais abaixo criando em uma ambiente mocado.

7. DESCRIÇÃO DAS MITIGAÇÕES APLICADAS EM NOSSA POC – FASE 4

Separamos aqui um resumo das ações mitigatórias mapeadas nas fases anteriores e apresentamos sua aplicação pratica no ambiente proposto. A apresentação do protótipo será efetuada via Figma, já a demonstração dos controles ativos será efetuada em um ambiente controlado (mocado) e demonstrado no video preparado pelo grupo.

Vale ressaltar que controles baseados no Cliente não estão sendo detalhados aqui, o que não elimina sua recomendação/aplicação futura, assim como recomendações de proteção como Antivírus, Firewalls, treinamento de conscientização de segurança e outros do lado do usuário final.

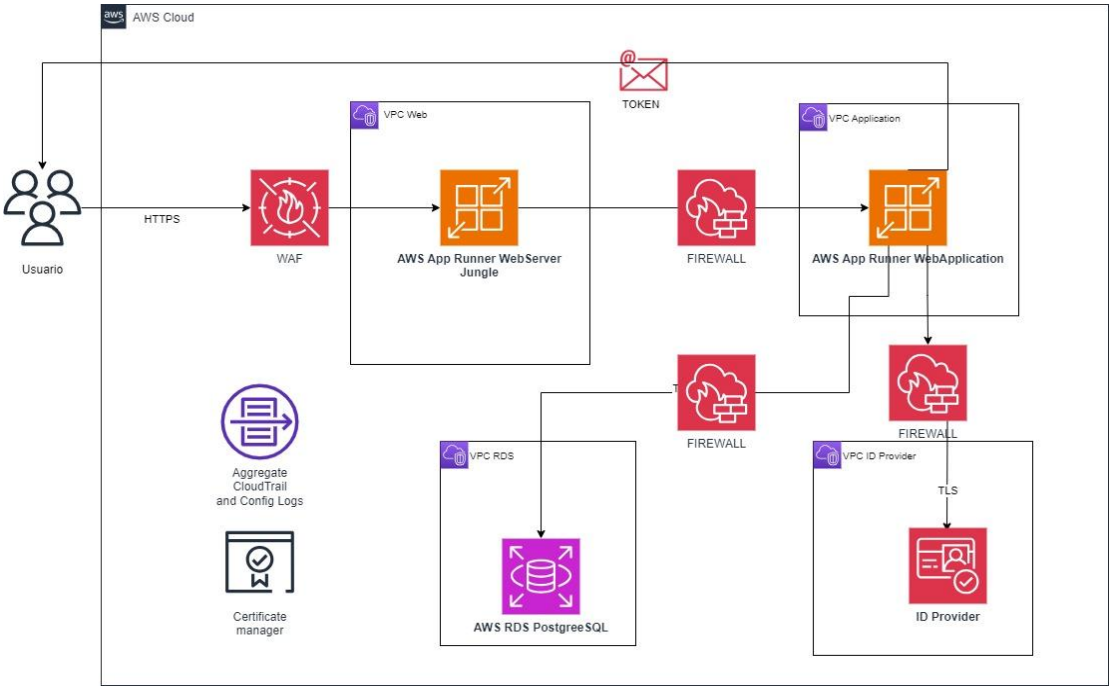
ESTRTUTURA FUNCIONAL

Como mencionado na modelagem de ameaça das entregas anteriores, nosso fluxo proposto de acesso decorre da seguinte forma:

1. O usuário acessa o site e realiza o cadastro no sistema;
2. O usuário faz o login com usuário e senha cadastrados;
3. Existe um desafio (captcha) para proteção contra ataques de robô e scripts;
4. O sistema valida o usuário/mail enviando um token para o mail cadastrado;
5. O usuário realiza o registro contendo seu feedback;
6. O usuário faz o logoff.

SEGMENTAÇÃO DE REDES E MINIMO PRIVILÉGIO

O ambiente proposto foi construído com uma segmentação de redes visando que ativos estejam isolados em redes separadas e que todo tráfego seja analisado via Firewall. Essa segmentação é demonstrada abaixo:



Essa segmentação juntamente com aplicação de controles rígidos de acesso, garantem um ambiente controlado, com menor privilégio, separando funções e acessos conforme necessidade especificas do projeto:

- A estrutura de Serviços e Servidores é segmentada por redes DMZ;
- Tráfego entre serviços WebServer, WebAplication, Authorization Provider e Banco de Dados é controlado e limitado;
- A movimentação lateral é limitada pelo isolamento dos serviços e redes;
- O impacto de um incidente é limitado ao acesso mínimo existente devido segmentação;
- As permissões são atribuídas por usuários baseado sobre sua responsabilidade;
    - DBA – Banco de Dados;
    - Frontend Developer – WebServer;
    - Backend Developer – Web Application;
    - Security Team – ID Authetication Porvider;

Outra característica que essa segmentação da, é a modularidade do projeto para um crescimento lateral em casos de pico de uso e/ou necessidades de maior consumo da plataforma.

ARQUITETURA TECNOLOGICA

Com o objetivo de aplicar a modelagem de ameaças conforme planejamento, tomamos a decisão de criar uma aplicação web utilizando Django com uso de um banco de dados PostgreSQL apoiada em uma estrutura da AWS.

A adoção de um framework ajuda nos processos de proteção por disponibilizar controles mitigatórios de segurança embarcados e de maturidade pelo uso do mercado.

Optamos por utilizar Django por este ser considerado um framework mais seguro com ações de proteção a SQL. Essa fama se dá por alguns motivos, como por exemplo:

- O ORM (Object-Relational Mapping) de Django protege automaticamente contra a maioria dos tipos de ataques de injeção de SQL ao usar consultas parametrizadas.
- Ao utilizar o sistema de templates de Django, os caracteres que poderiam ser usados em ataques de injeção de SQL ou XSS (Cross-site Scripting) são escapados automaticamente. Permitindo que as strings usadas dentro de templates sejam tratadas de maneira a evitar que sejam interpretadas como código malicioso.
- Django possui várias características de segurança ativadas por padrão e embora algumas dessas características não estejam diretamente relacionadas ao SQL, elas contribuem para um ambiente de desenvolvimento mais seguro e proteção dos dados.
- A comunidade Django é ativa e sempre atenta às últimas vulnerabilidades de segurança. O framework é regularmente atualizado para corrigir as falhas de segurança mais recentes.

Fonte: https://docs.djangoproject.com/en/2.0/topics/security/

Adicionamos nessa fase, o uso do Flask para facilitar a geração de evidências nos testes apresentados de simulação de ataque ao ambiente.

CONTROLE DE REQUISIÇÃO E PROTEÇÕES DO AMBIENTE

Optamos por utilizar o framework Django que por padrão já possui diversas medidas segurança, como por exemplo: proteções contra falsificação de solicitação entre sites (CSRF), clickjacking, e outras vulnerabilidades comuns.

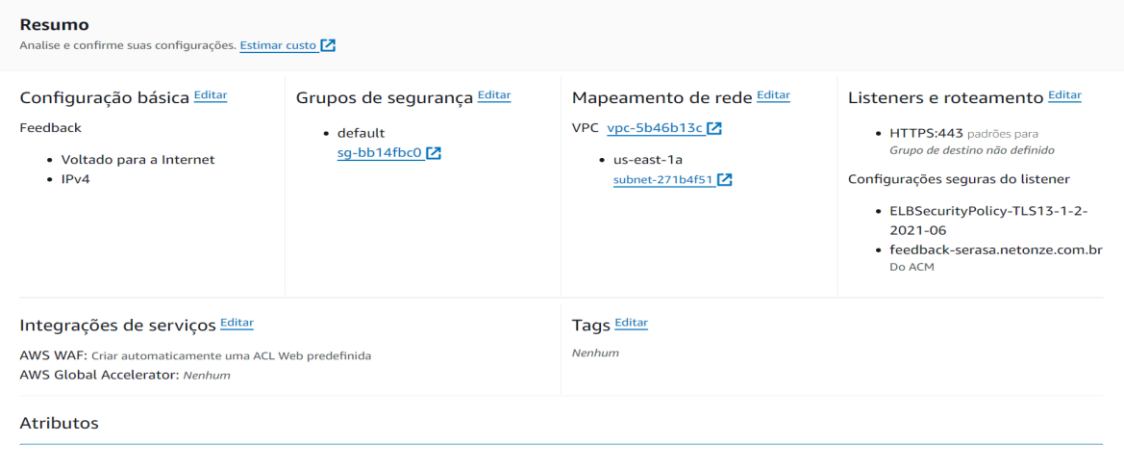
Também utilizamos parâmetros e consulta ao construir as consultas SQL, ao invés de concatenar os valores. Permitindo que os valores sejam corretamente escapados e evitando assim a possibilidade de injeção SQL.

Além disso, podemos ativar outros controles via regras de WAF para garantir outras salvaguardas necessárias e existentes dentro da política de proteção da SERASA EXPERIAN.

SSL

Para dispor a plataforma de forma segura, escolhemos utilizar um certificado SSL gerado via AWS Certificate Manager https://aws.amazon.com/pt/certificate-manager/ . Esse certificado segue padrões de segurança para garantir o tráfego protegido e autenticidade do acesso:

- Padrão TLS 1.2 ou superior
- Padrão RSA 2048



Outras configurações de seguranca associados ao SSL estão disponíveis via Django, habilitando uma camada adicional de controle:

- SECURE_SSL_REDIRECT True - Acesso somente via HTTPS
- SESSION_COOKIE_SECURE e CSRF_COOKIE_SECURE True - SeCurie Cookies
- SECURE_HSTS_SECONDS,
  SECURE_HSTS_INCLUDE_SUBDOMAINS,
  SECURE_HSTS_PRELOAD - Usar HTTP Strict Transport Security (HSTS)

AUTENTICAÇÃO FORTE E CRIPTOGRAFIA DE DADOS

Entendemos que este processo detém o maior desafio sobre toda a plataforma pois temos justamente a interação do usuário (HTTP POST) no login e a maioria dos riscos exploratórios mapeados na modelagem de ameaças. Dessa forma separamos os itens mitigatórios aplicáveis:

- Política de senha forte e bloqueio por tentativas de acesso;
- MFA/TOKEN;
- Desafio Captcha;
- Proteção de senhas criptografadas em banco de dados.

Política de senha

Aplicamos uma política baseada no uso de senhas com uma complexidade mínima além de política de bloqueio do acesso em casos de uso de senhas invalidas. Essa ação mítica por exemplo o uso de dicionários de senha ou até uso de senhas fracas na plataforma.

Mfa/Token

Por não ser uma aplicação transacional, optamos por introduzir o processo simples para validação do usuário com o envio de Token para garantir a autenticidade, sem adição por exemplo de TOTP com aplicativos autenticadores externos e mais complexos.

Esse token é enviado via e-mail para validação dos dados cadastrados para “garantir” primeiramente a identificação da pessoa e possibilidade de correlação com os dados existentes na base do SERASA EXPERIAN, além de auxiliar na proteção de uso de robôs e scripts como ataques de senha diversos.

Captcha

O uso do desafio (CAPTCHA) auxilia no controle de ataques de robôs e scripts automatizados além da sanitização dos acessos. Novamente, o Django ajuda nesse processo por ter recurso embutido em seu framework e de fácil implantação.

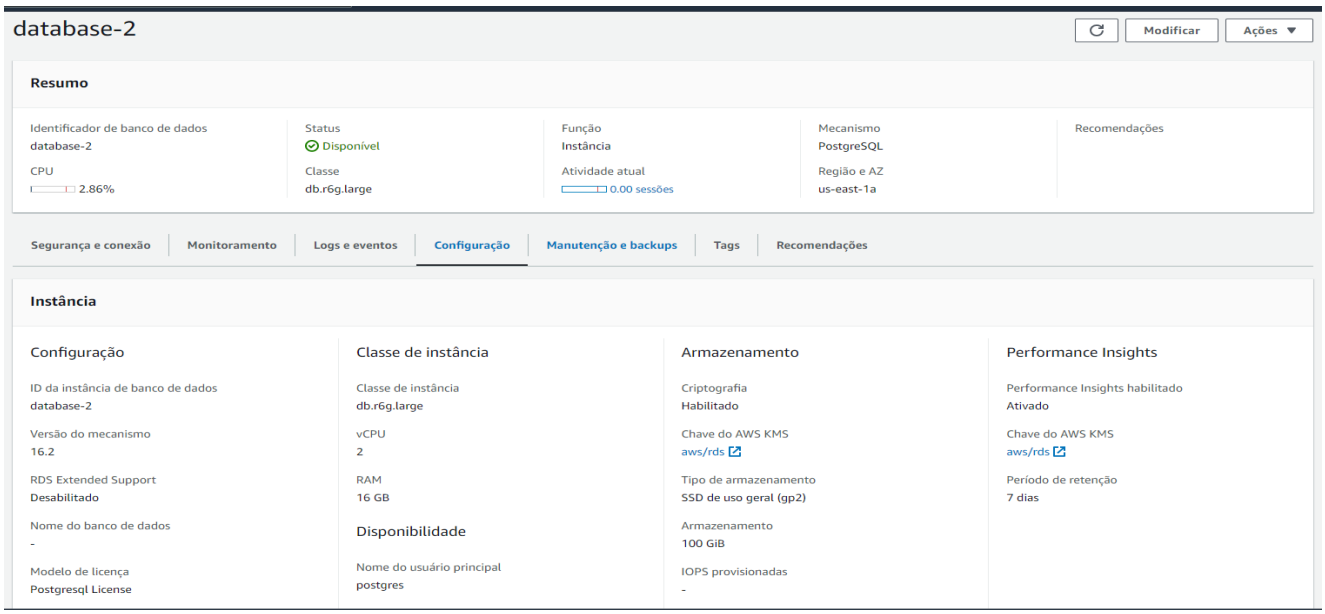
Proteção de senhas criptografadas em banco de dados

Outra medida de segurança, foi optar por salvar as senhas em banco utilizando criptografia para a guarda de dados sensíveis. Essa guarda evita o acesso indevido aos dados e protege contra seu uso abusivo.

BANCO DE DADOS ALTA DISPONIBILIDADE

Optamos pela utilização de um banco de dados PostgreSQL via RDS https://aws.amazon.com/pt/rds/postgresql/ que dispõe de recursos de segurança interessantes como:

- Segmentação de rede;
- Processos de backup e recuperação;
- Alta Disponibilidade e Escalonamento;
- Criptografia de Banco SSL



MONITORAMENTO – SIEM/SOC

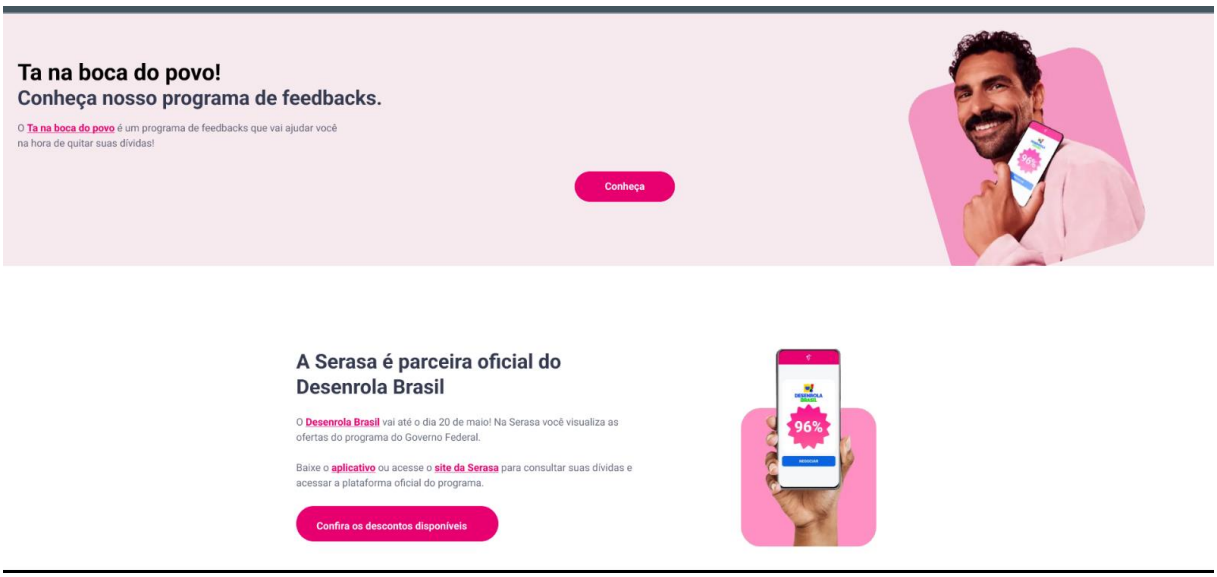
A solução foi projetada detém uma gestão de LOGs centralizados via Amazon Cloudwatch https://aws.amazon.com/pt/cloudwatch/. Sua centralização ajuda na análise de incidentes e correlação de eventos. Com base nos logs coletados via Cloudwatch, recomendamos introduzir a correlação de eventos ao ambiente de SIEM existente.

Sua integração pode seguir orientação ao conforme papper da AWS: https://aws.amazon.com/marketplace/solutions/control-tower/siem

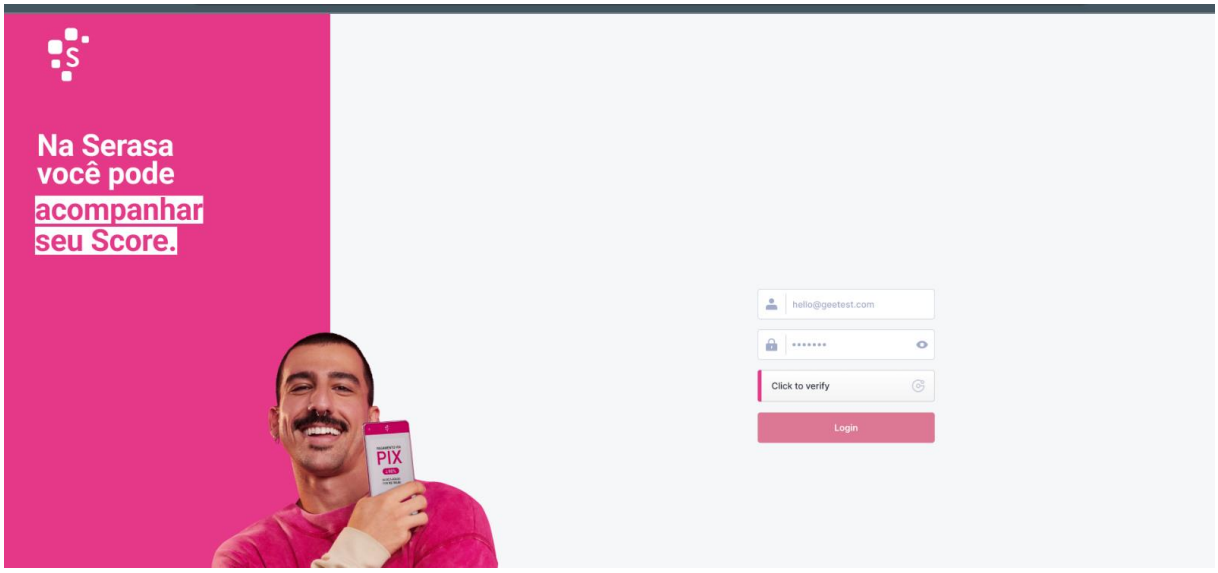
8. PROTOTIPAÇÃO

Optamos por apresentar o projeto por uma prototipação via Figma conforme telas detalhadas abaixo e video publicado e referenciado no link na introdução do projeto.

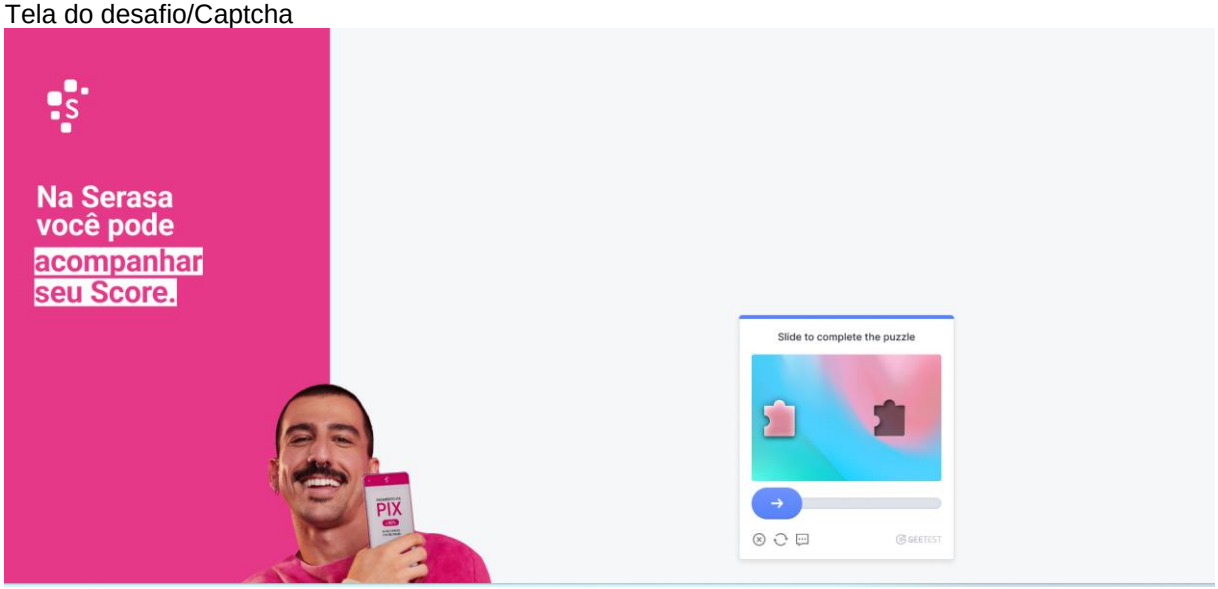
Tela inicial do projeto:



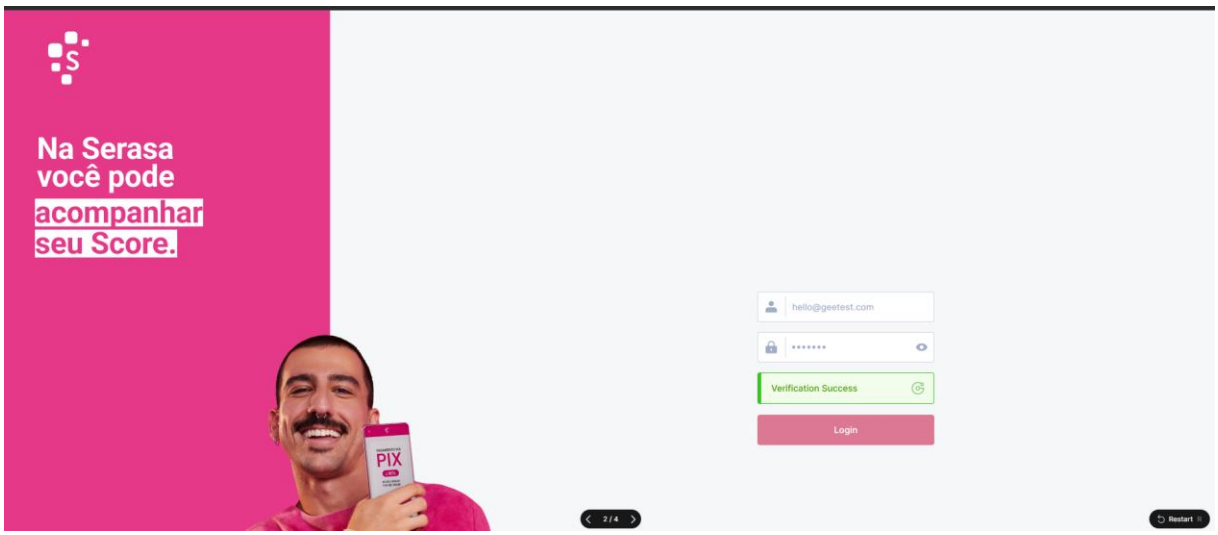
Tela de identificação do usuário:



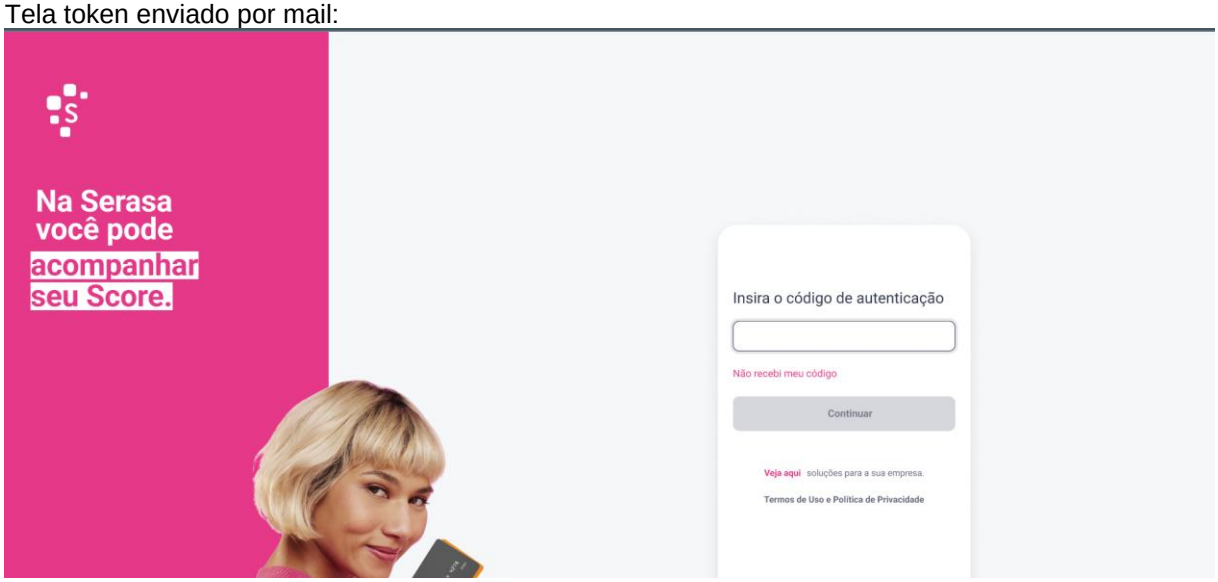
Tela do desafio/Captcha



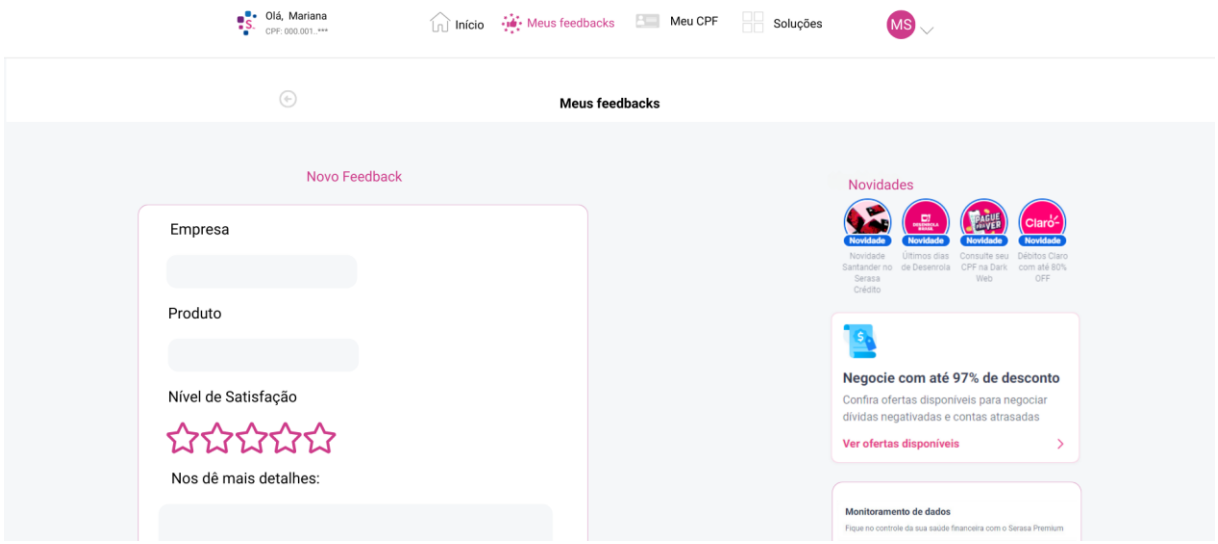
Tela desafio com sucesso:



Tela token enviado por mail:



Tela para registro do Feedback:



O acesso ao projeto está disponível em :
https://www.figma.com/proto/1tV09iLbYpTNmMQ0fTrTze/SERASA-CHALLENGE-USER-VIEW?node-id=2-1205&t=8HkOq2XtKtPht9U8-0&scaling=min-zoom&page-id=0%3A1

Para acesso completo a nosso código, por favor, acessar nosso repositório no GitHub:

https://github.com/GabrielMoreto/fiap_serasa

9. REFERÊNCIAS

- https://www.microsoft.com/en-us/security/blog/2007/09/11/stride-chart/
- https://docs.microsoft.com/en-us/azure/architecture/secure-by-design/threat-modeling-stride
- https://www.oreilly.com/library/view/threat-modeling/9781492056546/ch04.html
- https://www.iriusrisk.com/
- https://safecode.org/wp-content/uploads/2017/05/SAFECode_TM_Whitepaper.pdf
- https://www.zendesk.com.br/blog/feedback-do-cliente/
- https://www.threatmodelingmanifesto.org/
- https://learn.microsoft.com/pt-br/security/engineering/threat-modeling-with-dev-ops
- https://www.devmedia.com.br/sql-injection-em-ambientes-web/9733
- https://aiqon.com.br/blog/top-10-os-mais-comuns-ciberataques/
- https://awari.com.br/como-evitar-a-injecao-de-sql-no-django-dicas-e-melhores-praticas/?utm_source=blog&utm_campaign=projeto+blog&utm_medium=Como%20Evitar%20a%20Inje%C3%A7%C3%A3o%20de%20Sql%20no%20Django:%20Dicas%20e%20Melhores%20Pr%C3%A1ticas
- https://www.postgresql.org/
- https://docs.djangoproject.com/en/5.0/topics/security/
- www.aws.com
- https://flask.palletsprojects.com/en/3.0.x/