RELATÓRIO DE AUDITORIA PROGRAMADA

1. ORDEM DE SERVIÇO

OS nº 2019.0513

2. IDENTIFICAÇÃO

2.1. Objeto

Sistema Eletrônico de Informações (SEI).

2.2. Objetivo

Avaliar sob a ótica da Segurança da Informação em aplicações *web* a efetividade dos controles de segurança implementados na aplicação Sistema Eletrônico de Informações (SEI), utilizado pela Prefeitura Municipal de São Paulo (PMSP), sob a gestão da Secretaria Municipal de Gestão (SG), disponível no sítio da internet no endereço https://sei.prefeitura.sp.gov.br.

2.3. Metodologia

As principais técnicas utilizadas nos trabalhos de auditoria foram: entrevistas, análise documental, testes sistêmicos manuais e automatizados.

2.4. Critérios de Avaliação

Common Weakness Enumeration (CWE).

2.5. Área auditada

Secretaria Municipal de Gestão (SG)

2.6. Período da realização

18.09.19 a 30.09.20.

2.7. Período de abrangência

Não aplicável.

2.8. Siglas

Cade	Conselho Administrativo de Defesa Econômica
CWE	*Common Weakness Enumeration*
DPU	Defensoria Pública da União
Gati	Grupo de Auditoria de Tecnologia da Informação
GCM	Guarda Civil Metropolitana
HTTP	*HyperText Transfer Protocol*
HTTPS	*HyperText Transfer Protocol Secure*
MC	Ministério das Comunicações
ME	Ministério da Economia
PEN	Processo Eletrônico Nacional
PHP	*HyperText Preprocessor*
PMSP	Prefeitura Municipal de São Paulo
SEI	Sistema Eletrônico de Informações
SG	Secretaria Municipal de Gestão
SMSU	Secretaria Municipal de Segurança Urbana
SSL	*Secure Sockets Layer*
TCMSP	Tribunal de Contas do Município de São Paulo
TJMSP	Tribunal de Justiça Militar do Estado de São Paulo
TRF-4	Tribunal Regional Federal da 4ª Região
URL	*Uniform Resource Locator*

2.9. Equipe técnica

Adriano Gonçalves Zambon	RF nº 20.309
Carlos Albuquerque Lemos	RF nº 20.289
Helio Ricardo Guimarães Murci de Azevedo	RF nº 20.302

2.10. Escopo da auditoria

O escopo dos trabalhos de auditoria teve como alvo a aplicação Sistema Eletrônico de Informações (SEI). Os trabalhos foram executados por meio de testes, manuais e automatizados, que compreenderam algumas das funcionalidades (internas ou externas) disponibilizadas pelo sistema.

3. RESULTADO

3.1. Introdução

O Sistema Eletrônico de Informações (SEI), localizado no endereço eletrônico https://sei.prefeitura.sp.gov.br, é um sistema de gestão de processos e documentos eletrônicos, desenvolvido pelo Tribunal Regional Federal da 4ª Região (TRF-4), que engloba diversos módulos e funcionalidades, a fim de promover a eficiência da administração pública em diferentes esferas (peça 7).

A PMSP assinou um acordo de cooperação técnica com o Tribunal Regional Federal da 4ª Região para uso do SEI em 01.12.2014.

Atualmente, o SEI é utilizado em diversos órgãos nacionais, como os Tribunais Federais da 1ª, 3ª e 4ª regiões, o Conselho Administrativo de Defesa Econômica (Cade), a Defensoria Pública da União (DPU), o Tribunal de Justiça Militar do Estado de São Paulo (TJMSP), o Ministério das Comunicações (MC) e o Ministério da Economia (ME), entre outros. É o sistema escolhido para o projeto do Processo Eletrônico Nacional (PEN), do qual a PMSP também participa.

3.2. Reunião de abertura

Inicialmente, por conta da Pandemia Covid-19, a reunião de abertura aconteceu por meio telefônico, na qual foram passados ao Sr. Waldir Agnello, Chefe de Gabinete da Secretaria Municipal de Gestão, os detalhes dos trabalhos a serem realizados por esta Corte de Contas.

Posteriormente, recebemos da Sra. Malde Maria Vilas Bôas, Secretária Municipal de Gestão, o acesso ao ambiente de teste do sistema SEI (https://sei.treinamento.prefeitura.sp.gov.br) junto com dois usuários, também para propósitos de teste, que são TCM_USER_01 e TCM_USER_02.

3.3. *Common Weakness Enumeration* (CWE)

A avaliação da aplicação SEI teve como base a lista de fragilidades e vulnerabilidades de *software* catalogadas na *Common Weakness Enumeration* (CWE).

A CWE é uma lista desenvolvida pela comunidade de profissionais de desenvolvimento e segurança de *software*, na qual os tipos comuns de fragilidades e vulnerabilidades são catalogados em classes de vulnerabilidades de segurança.

O principal objetivo da CWE é a resolução das vulnerabilidades existentes na fonte, auxiliando e orientando arquitetos, *designers* e programadores de *software* na eliminação de erros e de vulnerabilidades mais comuns antes que o *software* e o *hardware* sejam entregues aos usuários.

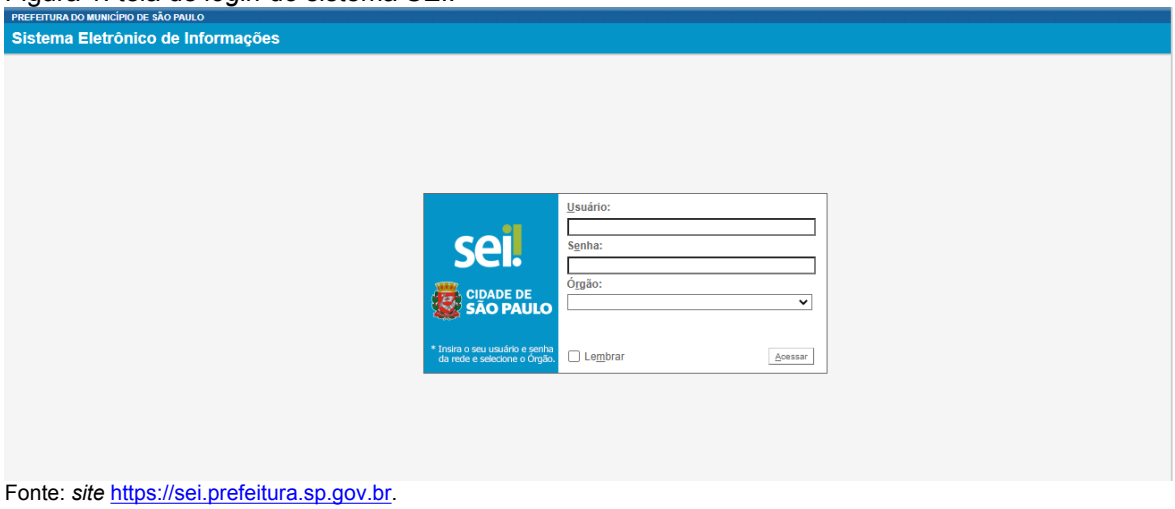
Segundo definição dada pela CWE, vulnerabilidades e erros são as possíveis falhas, *bugs* e demais problemas encontrados na implementação de *software* ou de *hardware*, de código fonte, de *design* ou de arquitetura que, se deixados sem tratamento, podem resultar em sistemas vulneráveis a ataques.

A lista CWE e a taxonomia a ela associada servem como um guia a ser utilizado para identificar e descrever essas vulnerabilidades sob o prisma de uma linguagem comum e padronizada.

3.4. Sistema SEI

O acesso ao sistema SEI pode ser classificado em dois níveis. O primeiro nível de acesso é aquele em que qualquer cidadão consegue visualizar a tela de *login* do sistema. A figura 1 demostra esse tipo de acesso:

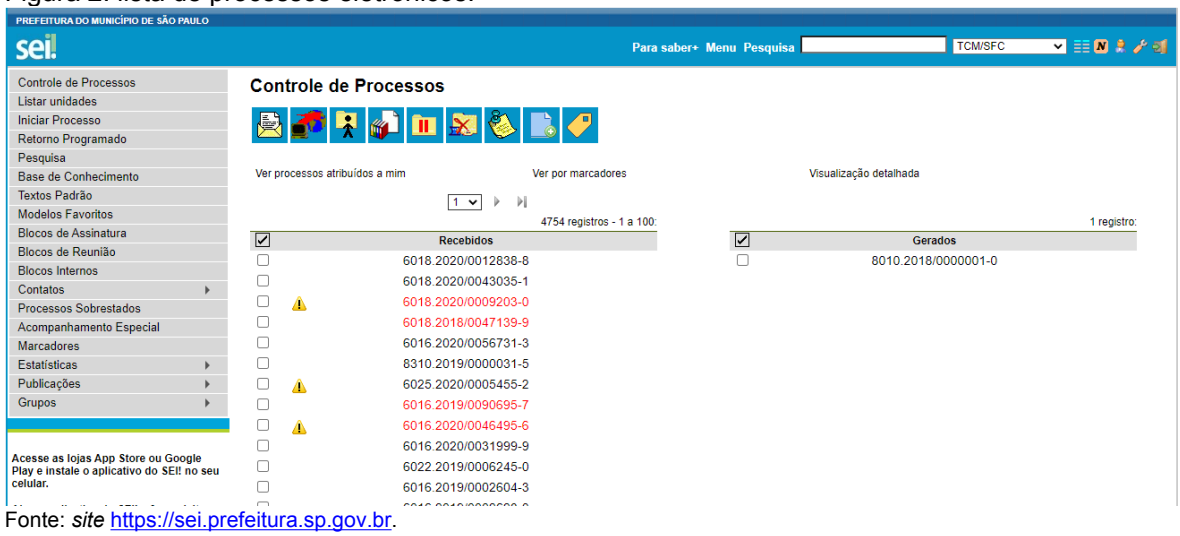
Figura 1: tela de *login* do sistema SEI.



Fonte: *site* https://sei.prefeitura.sp.gov.br.

O segundo nível de acesso é aquele em que usuários credenciados têm acesso aos processos eletrônicos da PMSP, dependendo de seus privilégios no sistema. A figura 2 demostra a lista de processos eletrônicos disponibilizados para um usuário cadastrado no sistema SEI:

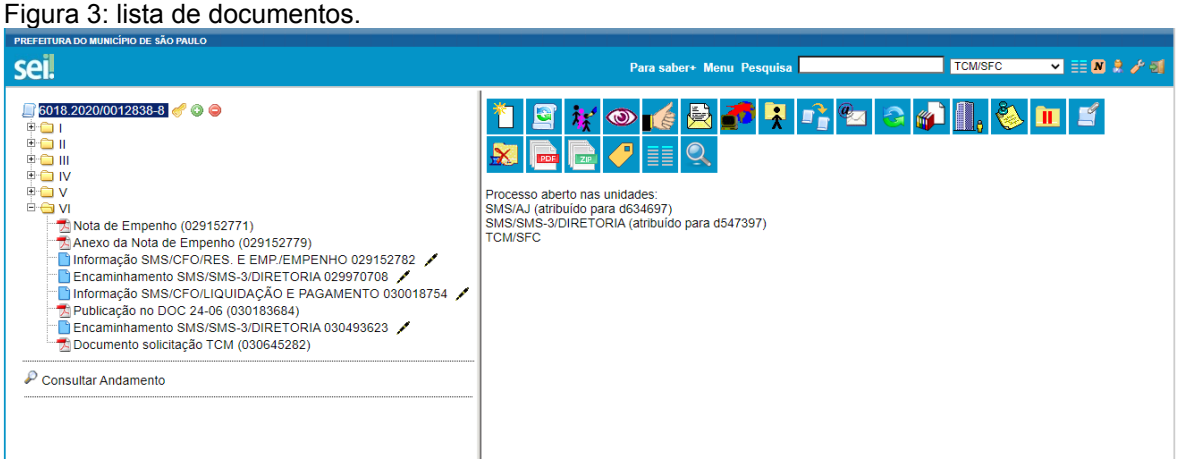
Figura 2: lista de processos eletrônicos.



Fonte: *site* https://sei.prefeitura.sp.gov.br.

A figura 3 mostra os documentos pertencentes a um processo eletrônico escolhido aleatoriamente:

Figura 3: lista de documentos.



Fonte: *site* https://sei.prefeitura.sp.gov.br.

3.4.1. Análise do *site* (usuário de primeiro nível)

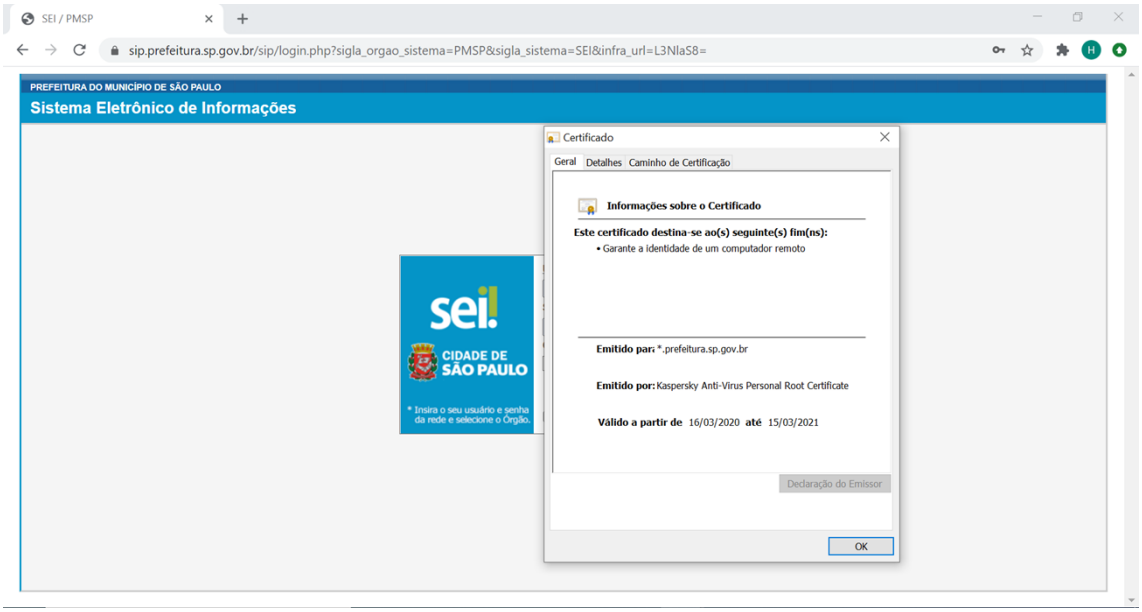
A fim de avaliar a segurança oferecida pelo sistema no primeiro nível de acesso (sem identificação no sistema), a equipe do Grupo de Auditoria de Tecnologia da Informação (GATI) realizou testes de vulnerabilidade usando ferramentas automatizadas para esse propósito.

O sistema SEI não demonstrou vulnerabilidades sob a análise dessas ferramentas. Em sua maioria, não foram encontradas vulnerabilidades com grandes gravidades. Contudo, alguns pontos de risco foram encontrados e são a seguir descritos:

3.4.1.1. *Wildcard Certificate*

Durante varreduras realizadas no Sistema Eletrônico de Informações (SEI), fazendo uso da ferramenta de escaneamento Nikto, foi detectada a utilização de *Wildcard Certificates* (Certificados SSL Curingas) no subdomínio https://sei.prefeitura.sp.gov.br (peça 8), conforme ilustra a figura 4:

Figura 4: evidência da utilização de *Wildcard Certificates*.



Fonte: *site* https://sei.prefeitura.sp.gov.br.

De modo geral, um certificado curinga funciona de forma semelhante aos certificados convencionais, isto é, um par de chaves criptografadas e auto assinadas, baseadas no protocolo *Security Socket Layer* (SSL), que protegem um domínio específico. Entretanto, no caso do certificado curinga, devido a sua versatilidade inerente, poderá ser utilizado para estender a proteção criptográfica a um domínio único e a todos os seus subdomínios de primeiro nível.

Por padrão, a opção pela utilização de certificados curingas como forma de proteção criptográfica na maioria das vezes dá-se pelo menor custo financeiro envolvido na aquisição, manutenção e gerenciamento desse tipo de certificado, assim como pela facilidade na extensão do uso e aparente facilidade no gerenciamento operacional.

Por outro lado, a utilização de um mesmo certificado curinga em inúmeros subdomínios é uma situação que poderá ocasionar diversas vulnerabilidades de segurança, entre elas a possibilidade de cibercriminosos se valerem da oportunidade de maior superfície de ataque, aumentando assim a probabilidade de êxito no comprometimento do sigilo e da integridade das chaves criptográficas

utilizadas no certificado curinga vinculado ao domínio e seus respectivos subdomínios.

Entre as possíveis consequências da exploração desse tipo de vulnerabilidade está o comprometimento da segurança, não só da aplicação ora analisada, mas também, dos demais subdomínios vinculados ao domínio principal, neste caso, o domínio da PMSP (prefeitura.sp.gov.br).

Por fim, é importante que os gestores compreendam os riscos assumidos quando da utilização inadvertida desse tipo de certificado, pois é necessário que exista uma análise de riscos adequada e a aplicação de controles de segurança aderentes aos resultados da análise e da avaliação, a fim de sejam evitados ou ao menos mitigados os impactos de possíveis ataques, garantindo assim a integridade e o sigilo das chaves criptográficas utilizadas no *Wildcard Certificate* e, consequentemente, evitando o comprometimento da segurança de toda hierarquia de domínios e subdomínios da PMSP.

Dessa forma, é necessário que a Secretaria de Gestão apresente a análise e avaliação de riscos que subsidiaram a adoção da solução criptográfica por meio de *Wildcard Certificate* ou Certificados Curingas e apresente os controles de segurança utilizados para garantir a segurança dos certificados, domínios e subdomínios envolvidos. (**Conclusão 4.1**).

3.4.1.2. PhpInfo().php

O sistema SEI foi escrito na linguagem *Hypertext Preprocessor* (PHP), umas das mais populares linguagens para o desenvolvimento de aplicações *web*.

O arquivo PhpInfo().php é um arquivo que contém informações valiosas sobre a aplicação, tais como endereço do servidor (IP), a versão do sistema operacional, *e-mail* do administrador do sistema, entre outros. O *site* da PHP (https://www.php.net/manual/pt_BR/function.phpinfo.php) traz os seguintes comentários sobre o arquivo:

> *Mostra uma grande quantidade de informações sobre o estado atual do PHP. Isto inclui informações sobre as opções de compilação do PHP e extensões, a versão do PHP, informações do servidor e ambiente (se compilado como um módulo), o ambiente PHP, informação da versão do SO, caminhos, valores principais e locais das opções de configuração, cabeçalhos HTTP e a licença do PHP.*

As varreduras realizadas na página principal do SEI revelaram a exposição do arquivo PhpInfo().php, tanto para o ambiente de Produção (peça 9) quanto no ambiente de Testes (peça 10).

Conforme a referência **CWE-200: *Information Exposure*** do CWE, no intuito de mitigar os riscos de exposição de informações sensíveis, a SG deve tomar providências para corrigir a exposição dessas informações. (**Conclusão 4.2**).

3.4.2. Política de senhas

A política de senhas é um conjunto de regras destinadas a aumentar a segurança dos sistemas de computadores por meio do incentivo aos usuários para utilização de senhas fortes e o uso correto delas.

Verificamos que o SEI possui uma política de senhas fortes com exigência de troca de senha a cada noventa dias.

3.4.3. Ataque de força bruta

O ataque de força bruta (ou *brute force*) em sistemas consiste na busca exaustiva de chave, por tentativa e erro, usado contra quaisquer dados criptografados, tais como nome de usuário e senha e, assim, permitir a execução de processos, a escalação de privilégios e o acesso a *sites*, a computadores e a serviços em nome e com os mesmos privilégios de acesso do usuário atacado.

Apesar dos ataques de força bruta poderem ser realizados manualmente, na grande maioria dos casos são realizados com o uso de ferramentas especializadas, facilmente obtidas na internet, as quais permitem a execução automática de ataques.

Nos testes de ataque, tentou-se a realização de *login* com um usuário conhecido e diversas senhas diferentes, por meio de tentativas e erros, no intuito de se conseguir acesso ao sistema.

Os testes foram frustrados, demonstrando que o SEI, em diversas situações de ataque, possui um bom sistema de defesa.

3.4.3.1. Ferramenta de automação

Considerando que as ferramentas usadas para força bruta, comumente usadas nos testes de vulnerabilidade, apresentaram resultados frustrados, o GATI desenvolveu sua própria ferramenta de automação para realizar o ataque de força bruta.

O objetivo da ferramenta é automatizar o procedimento de usuário e senha, inserindo, para um mesmo usuário, várias senhas para conseguir a quebra de acesso ao sistema SEI, usando uma abordagem diferente das ferramentas automatizadas.

O resultado foi bem sucedido, pois a ferramenta conseguiu se desviar do sistema de defesa do SEI.

O sistema alvo foi o SEI do ambiente de Teste e o usuário foi o TCM_USER_01. Com a ferramenta, foi possível realizar, com base em uma lista aleatória de senhas (peça 11) o credenciamento do usuário na oitava tentativa (peça 12).

Consignamos que, embora a ferramenta tenha sido bem sucedida em seu propósito de comprovação da vulnerabilidade, a probabilidade de sucesso em situação real

é baixa, pois com a política de senhas fortes o processo de tentativa e erro realizado pela ferramenta pode levar muito tempo para ser bem sucedida.

Contudo, **comprova-se a vulnerabilidade** e, conforme a referência **CWE-307: *Improper Restriction of Excessive Authentication Attempts*** do CWE, a fim de mitigar os riscos de acessos indevidos, de quebra de integridade dos dados e de exposição de informações sensíveis, a SG deve tomar providências **urgentes** para corrigir o problema. (**Conclusão 4.3**).

3.4.4. Análise do *site* (usuário de segundo nível)

O objetivo dos testes neste nível de acesso serve para verificar se o *site* apresenta vulnerabilidades para usuários credenciados com acesso ao sistema.

3.4.5. Quebra de acesso

Esse tipo de ataque se baseia na alteração de valores de uma determinada página e na tentativa de conseguir acesso a informações às quais determinado usuário não está autorizado.

Uma vez credenciados e com acesso ao sistema SEI, acessamos aletoriamente um processo no qual tínhamos o privilégio de leitura de todos os seus documentos.

Usando credenciais desse processo, tentamos acessar outro processo no qual não tínhamos acesso e o sistema SEI bloqueou o acesso, como demonstrado na Figura 5.

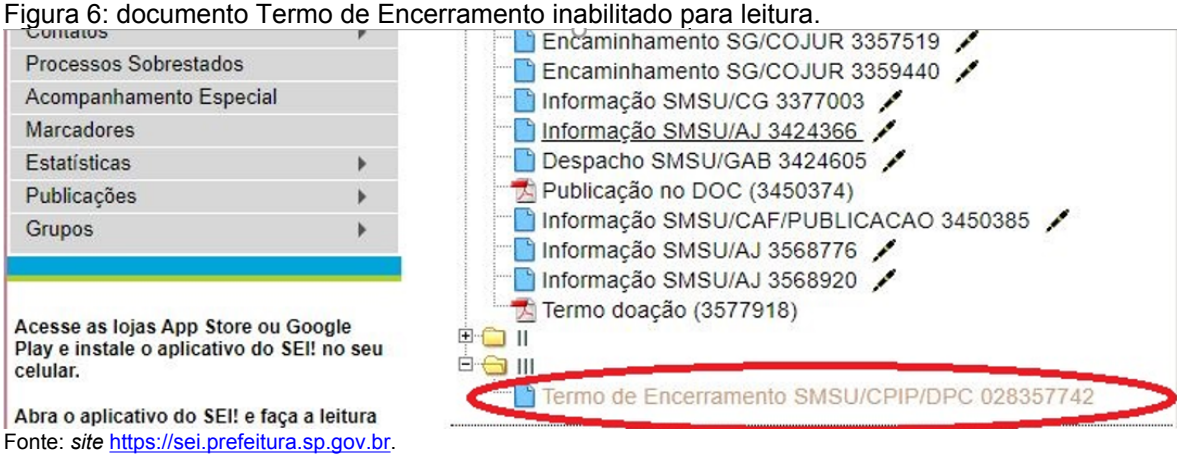
Figura 5: *X-Frame Options*.

Fonte: *site* https://sei.prefeitura.sp.gov.br.

Outra tentativa foi o acessar um processo aleatoriamente e tentar habilitar um documento inabilitado para leitura.

Escolhemos um processo em que havia um documento inabilitado para leitura, ou seja, para as credenciais do usuário acessando esse processo, o acesso ao documento não estava habilitado, conforme demonstrado na figura 6:

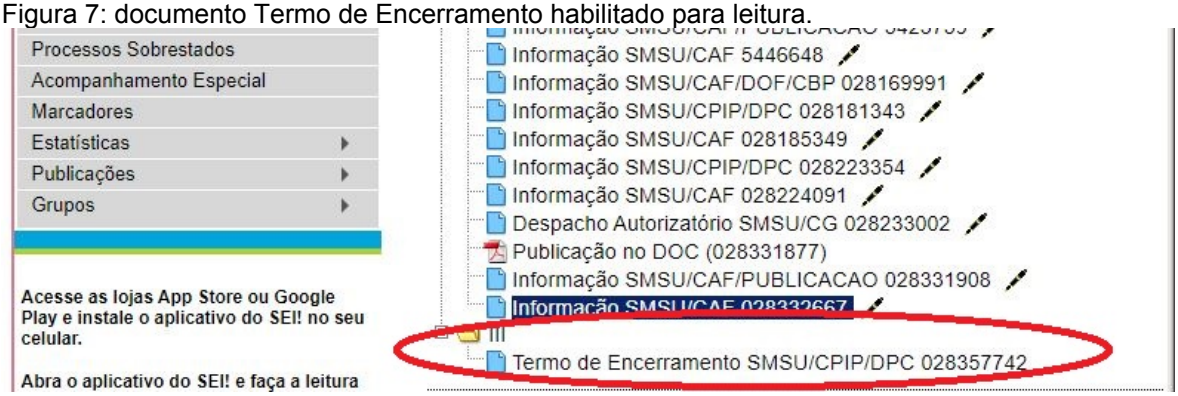
Figura 6: documento Termo de Encerramento inabilitado para leitura.



Fonte: *site* https://sei.prefeitura.sp.gov.br.

Conseguimos destravar o status de inabilitado e o documento passou a ser habilitado para o usuário, como demonstrado na figura 7:

Figura 7: documento Termo de Encerramento habilitado para leitura.



Fonte: *site* https://sei.prefeitura.sp.gov.br.

Contudo, mesmo habilitando o documento, não foi possível ter acesso ao seu conteúdo. Mais uma vez, o sistema SEI apresentou uma defesa eficiente contra acessos indevidos.

3.4.6. *Upload* de arquivos e sanitização

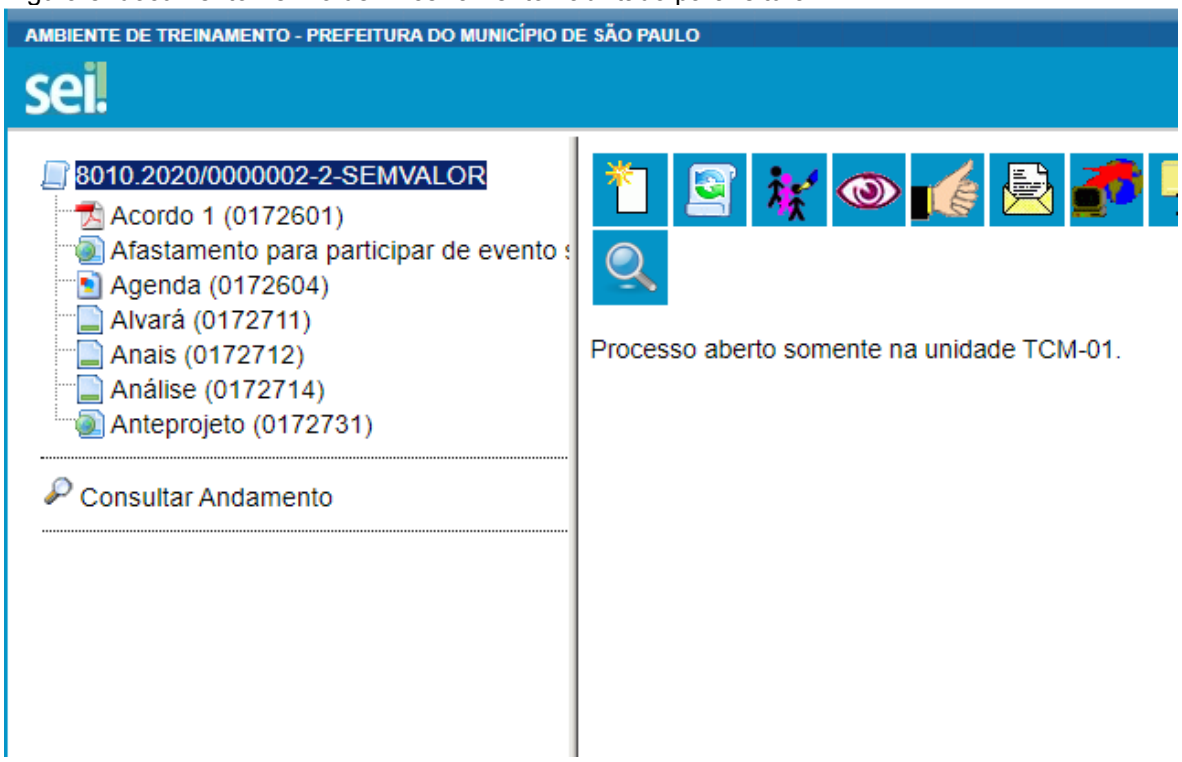
Falhas de injeção, em geral, ocorrem quando dados não confiáveis são enviados para uma aplicação como parte de um comando ou de consulta legítima. Esses dados hostis podem enganar a aplicação, levando-a a executar comandos não pretendidos ou a acessar a dados sem a devida autorização. No caso das vulnerabilidades de injeção de código, os dados não confiáveis são códigos a serem executados pela aplicação alvo.

Para evitar problemas de injeção, os sítios devem realizar um trabalho de sanitização, isto é, aplicar técnicas e ferramentas diversas para identificar e remover dados e códigos maliciosos das entradas dos usuários, a fim de que não obtenham sucesso em enganar a aplicação.

No teste em comento, o alvo foi o procedimento de inclusão de documentos com códigos maliciosos, ou seja, para um determinado processo fizemos a inclusão de documentos maliciosos para testar a vulnerabilidade do sistema.

Inicialmente, criamos um processo no ambiente de testes com o número 8010.2020/0000002-2 (ver figura 8) e iniciamos a inclusão de documentos com códigos maliciosos.

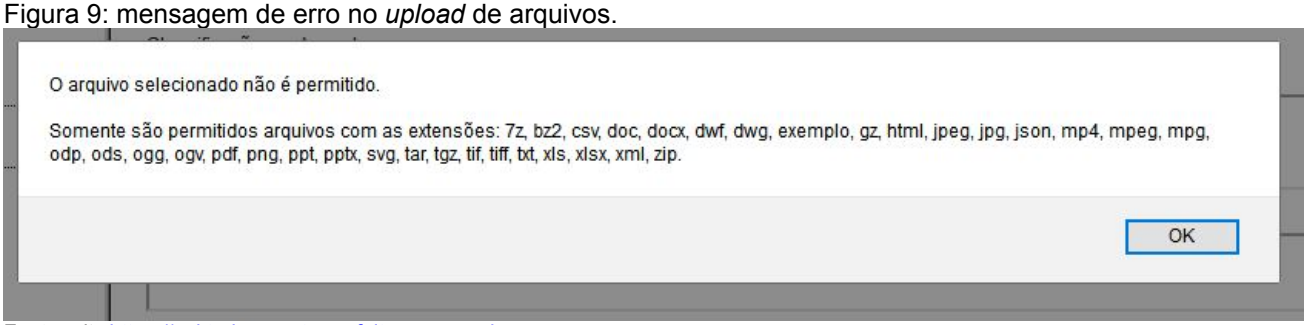
Figura 8: documento Termo de Encerramento habilitado para leitura.



Fonte: *site* https://sei.treinamento.prefeitura.sp.gov.br.

O sistema SEI apresentou defesa para códigos *JavaScript* e PHP com correta sanitização para muitas tentativas de injeção de código. Além disso, o SEI só permite alguns tipos de arquivos para serem carregados no sistema. Tentamos inserir um arquivo com código malicioso e obtivemos resposta de negação desse procedimento, conforme demonstrado na figura 9:

Figura 9: mensagem de erro no *upload* de arquivos.



Fonte: *site* https://sei.treinamento.prefeitura.sp.gov.br.

Com base na lista de arquivos permitidos (figura 9), iniciamos uma nova investida de ataques.

3.4.6.1. *Upload* de arquivo SVG

Scalable Vector Graphics (SVG), que pode ser traduzido do inglês como gráficos vetoriais escalonáveis, baseia-se na linguagem XML para descrever de forma vetorial desenhos e gráficos bidimensionais.

Como este arquivo tem a base XML, fizemos a inserção de um código malicioso (*JavaScript*) e realizamos o *upload* do arquivo no processo em comento.

O resultado foi que o SEI não eliminou o código malicioso desse arquivo e, caso os usuários compartilhassem o processo e acessassem o arquivo, poderiam sofrer um ataque do referido código. As figuras 10, 11 e 12 ilustram os procedimentos de ataque.

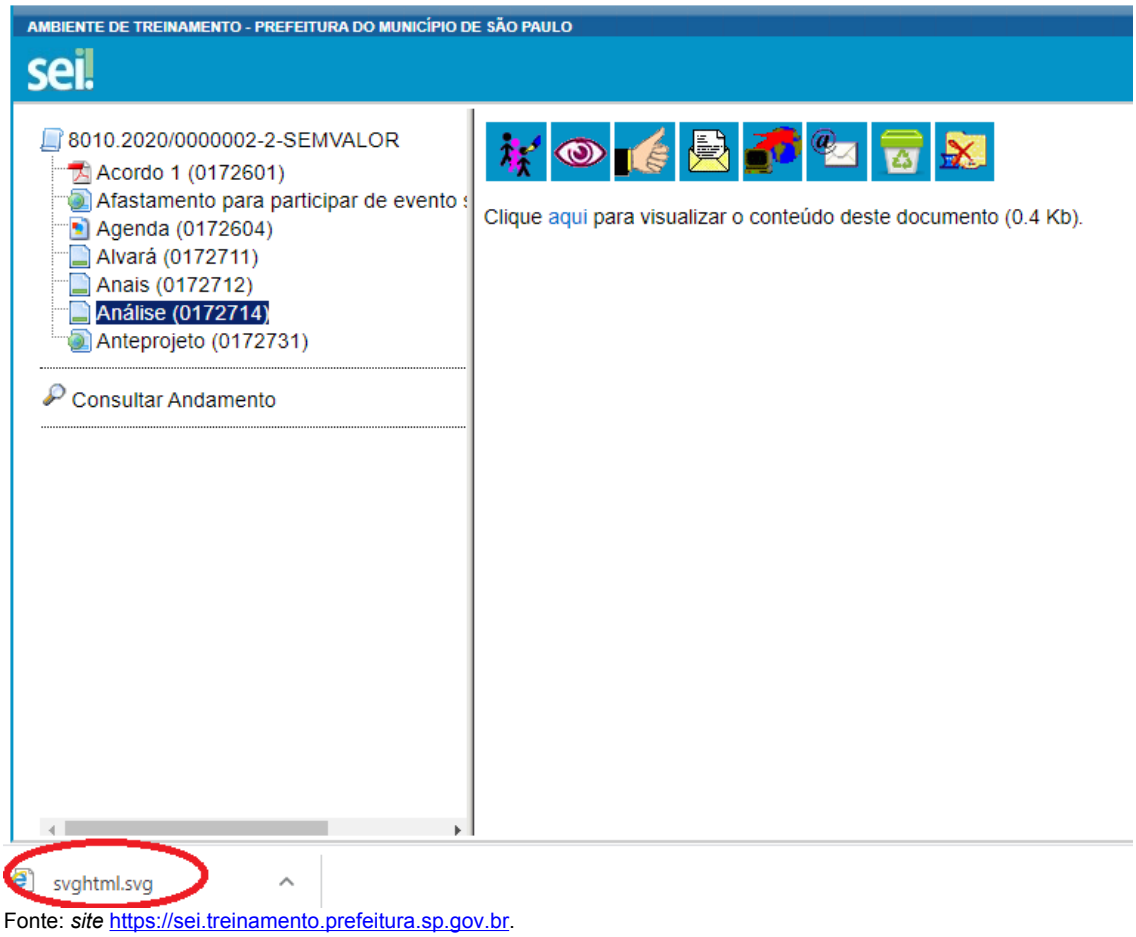
A figura 10 mostra o arquivo svghtml.svg sendo carregado no sistema:

Figura 10: arquivo SVG carregado no sistema SEI.

Fonte: *site* https://sei.treinamento.prefeitura.sp.gov.br.

A figura 11 mostra o acesso ao arquivo svghtml.svg. Nesse procedimento, notamos que o arquivo não é mostrado no sistema SEI, mas é feito um *download* dele.

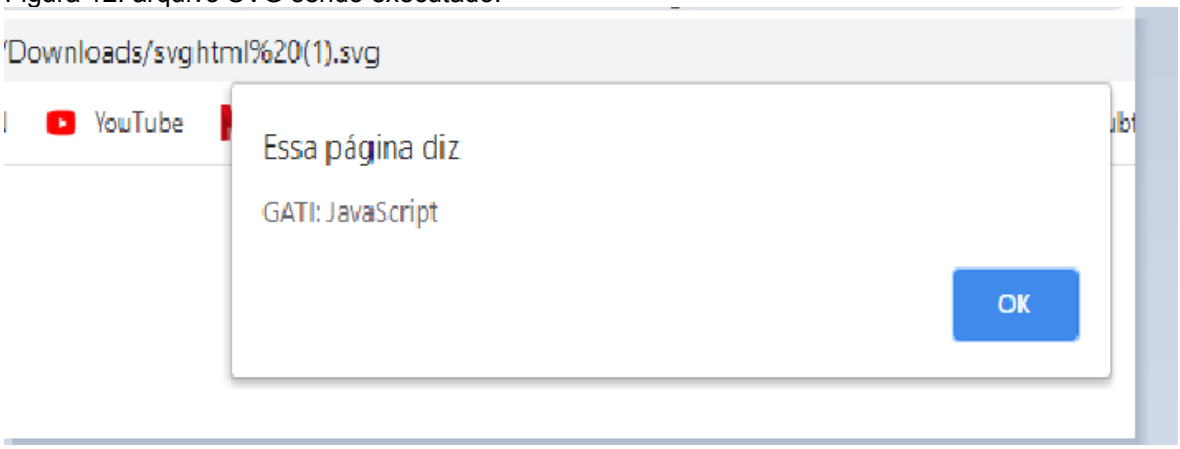
Figura 11: arquivo SVG sendo acessado no sistema SEI.



Fonte: *site* https://sei.treinamento.prefeitura.sp.gov.br.

Na figura 12, vemos que, quando o usuário executa o arquivo svghtml.svg, um código *JavaScript* é executado, comprovando não houve sanitização do arquivo svghtml.svg no momento de carga do arquivo no sistema SEI.

Figura 12: arquivo SVG sendo executado.



Fonte: *site* https://sei.treinamento.prefeitura.sp.gov.br.

Após a execução do *JavaScript*, o código malicioso direcionou o usuário ao *site* do Tribunal de Contas do Município de São Paulo, como demonstrado na figura 13. Contudo, o usuário poderia ter sido direcionado a um *site* com as mesmas características do SEI, exigindo novamente as credenciais do usuário e assim haveria a quebra de acesso do sistema SEI. (**Conclusão 4.4**).

Figura 13: arquivo SVG sendo executado.

Fonte: *site* https://portal.tcm.sp.gov.br.

O apontamento para esta vulnerabilidade se encontra na referência na CWE em **CWE-611: *Improper Restriction of XML External Entity Reference.***

3.4.6.2. *Upload* de arquivo ZIP

O SEI também permite o *upload* de arquivos do tipo ZIP, formato de compactação de dados muito difundido pela internet. Esse formato pode ser usado para armazenar arquivos de outros formatos, tais como programas executáveis.

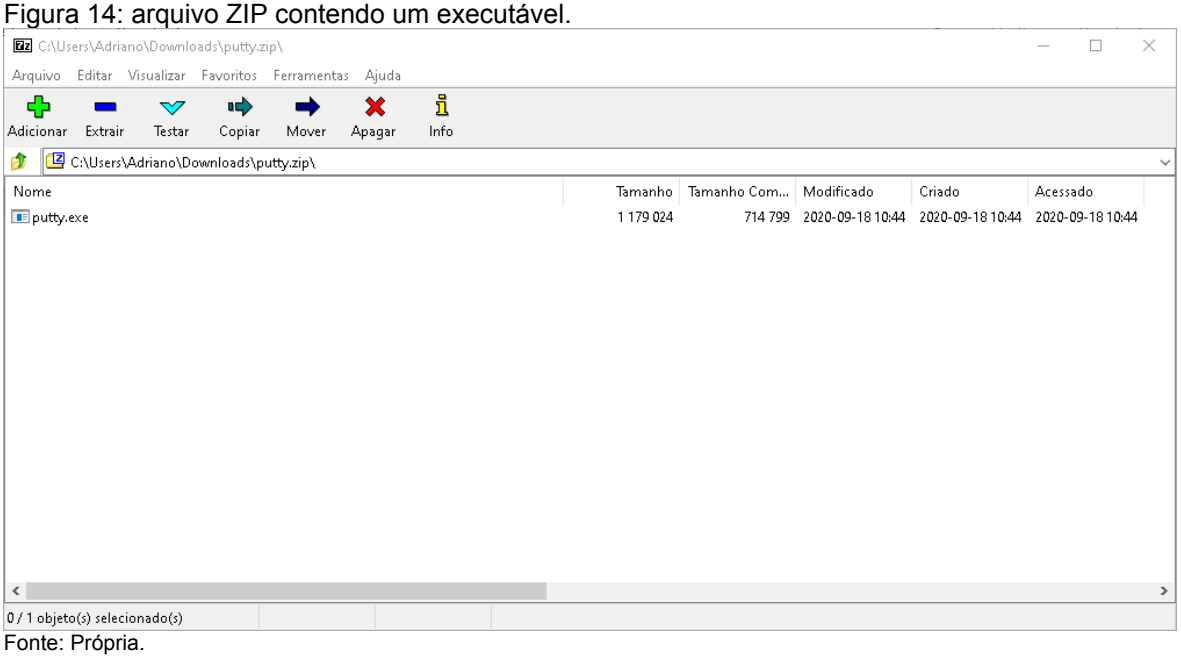
O SEI verifica e restringe o tipo de arquivo sendo carregado, inclusive o formato ZIP. Contudo, como o ZIP pode conter outros formatos, um atacante poderia armazenar suas ferramentas em um arquivo do tipo ZIP e utilizá-las como forma de distribuir um vírus ou uma aplicação indevida, por exemplo.

Para efeito de teste, um arquivo ZIP contendo um executável foi carregado em um processo SEI, da mesma forma que demonstrado no subitem 3.4.6.1. Assim como naquele caso, quando um usuário acessa esse arquivo, seu conteúdo não é exibido em tela, mas ele é copiado para o dispositivo do usuário, seja um computador ou *notebook*.

As figuras 14 e 15 ilustram os procedimentos de ataque que podem ser efetuados por meio da utilização de arquivos do tipo ZIP cujo conteúdo inclua outros arquivos com código malicioso.

A figura 14 mostra o conteúdo de um arquivo ZIP que inclui um programa executável, o qual poderia ser executado pelo usuário e causar danos tais como roubo de informações, exclusão de dados ou danos ao computador:

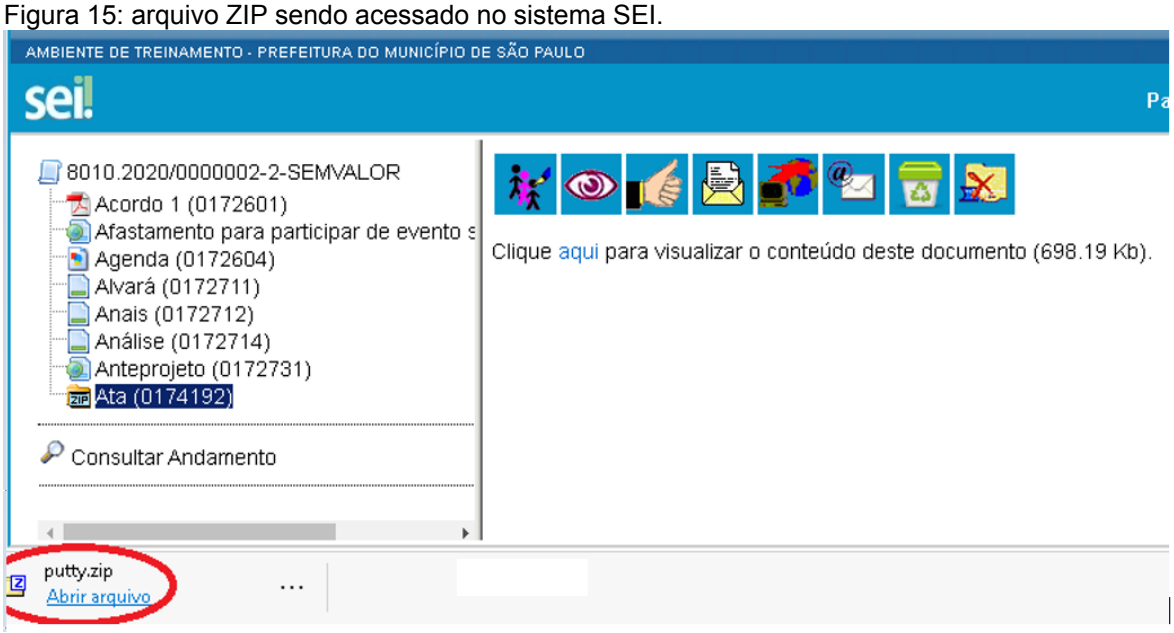
Figura 14: arquivo ZIP contendo um executável.



Fonte: Própria.

A figura 15 mostra o arquivo ZIP carregado no SEI. Nesse caso, notamos que seu conteúdo não é mostrado no sistema, mas quando ele é acessado, o *download* é feito para o dispositivo do usuário, que fica vulnerável ao executá-lo devido à confiança no conteúdo armazenado no SEI.

Figura 15: arquivo ZIP sendo acessado no sistema SEI.



Fonte: *site* https://sei.treinamento.prefeitura.sp.gov.br.

Quando aberto pelo usuário, o executável pode ser rodado no dispositivo, realizando as tarefas para as quais foi programado. No exemplo do teste, tratava-se de um programa benigno, mas essa vulnerabilidade poderia ser utilizada para causar danos aos usuários do SEI ou mesmo a organizações e/ou às estruturas computacionais.

Dessa forma, ainda que os arquivos compactados do tipo ZIP sejam importantes para a inclusão de documentos em formatos não suportados pelo SEI ou mesmo para documentos muito grandes, é necessário que seja implementada alguma forma de sanitização para o bloqueio de arquivos ZIP com conteúdo potencialmente danoso ou ainda a verificação automática desse tipo de arquivo por meio de ferramentas antivírus. (**Conclusão 4.5**).

O apontamento para esta vulnerabilidade se encontra na referência na CWE em **CWE- 434: *Unrestricted Upload of File with Dangerous Type.***

3.5. Responsável pela área auditada

Nome	Cargo
Malde Maria Vilas Bôas	Secretária Municipal de Gestão

4. CONCLUSÕES

À vista dos testes e análises realizadas no Sistema Eletrônico de Informações – SEI, a auditoria chegou aos seguintes apontamentos referentes à segurança da informação em aplicações *web*:

4.1. A SG deve apresentar a análise e a avaliação de riscos que embasaram tecnicamente a adoção da solução criptográfica por meio de *Wildcard Certificate* ou Certificados Curingas, assim como, deve apresentar os controles de segurança utilizados para garantir a segurança dos certificados, dos domínios e dos subdomínios envolvidos. **(subitem 3.4.1.1)**;

4.2. A SG deve tomar providências para evitar a exposição de informações

sensíveis contidas no arquivo PhpInfo().php dos ambientes de Produção e Teste do sistema SEI. (**subitem 3.4.1.2**);

4.3. A SG deve tomar providências urgentes a fim de mitigar os riscos de acessos indevidos, de quebra de integridade dos dados e de exposição de informações sensíveis provenientes de ataques de força bruta ao sistema SEI. **(subitem 3.4.3.1)**;

4.4. A SG deve implementar a devida sanitização durante o *upload* de arquivos do tipo SVG, a fim de que sejam mitigados os riscos de execução de códigos maliciosos no sistema SEI. **(subitem 3.4.6.1)**;

4.5. A SG deve implementar a devida sanitização durante o *upload* de arquivos do tipo ZIP, a fim de que sejam mitigados os riscos de execução de códigos maliciosos no sistema SEI. **(subitem 3.4.6.2)**.

Consignamos que, embora as vulnerabilidades demonstradas no sistema SEI sejam de baixo risco, esta auditoria não procurou apontar as probabilidades de ataques em um cenário real no qual isso poderia acontecer. Nosso objetivo é confirmar que existem vulnerabilidades e que a Administração deve corrigi-las.

Sugerimos restrição inicial completa à publicidade deste TC, pois o presente relatório expõe vulnerabilidades de segurança da informação que ainda serão objeto de análise e correção pelo órgão responsável. As vulnerabilidades do sistema SEI, reveladas nesta auditoria, podem ser utilizadas para ataques maliciosos de pessoas mal intencionadas.

Em 28.09.20.

Grupo de Auditoria de Tecnologia da Informação - GATI

ADRIANO GONÇALVES ZAMBON
Agente de Fiscalização

CARLOS ALBUQUERQUE LEMOS
Agente de Fiscalização

HÉLIO RICARDO GUIMARÃES MURCI DE AZEVEDO
Agente de Fiscalização

RENATO SAMBRA SUYAMA
Supervisor de Equipes de Fiscalização e Controle 5

LUÍS GUILHERME RIBEIRO DO VALLE DAMIANI
Coordenador do GATI

RAFAEL ALEXANDRE CAVALCANTI DA SILVA
Coordenador Chefe de Fiscalização e Controle III

RP: APV