Revista de Administração e Negócios da Amazônia, V.13, n.1, jan/abr, 2021
ISSN:2176-8366 DOI 10.18361/2176-8366/rara.v13n1p165-192
Seção de Trabalho Técnico-Tecnológico

ANÁLISE DE SEGURANÇA DO SISTEMA DE VOTAÇÃO ELETRÔNICA PELA INTERNET IMPLEMENTADO E ADOTADO PELA UNIVERSIDADE FEDERAL DE RONDÔNIA

Ewerton Rodrigues Andrade - ewerton.andrade@unir.br
Angelo de Oliveira - angelo@unir.br
Antônio Lemos Regis - regis@unir.br
Raimundo Josedi Ramos Veloso - veloso@unir.br

* Submissão em: 18/01/2021 |Aceito em: 26/04/2021

RESUMO

Este trabalho apresenta uma análise de segurança do sistema de votação eletrônica pela Internet implementado e adotado pela Universidade Federal de Rondônia, realizada de maneira independente, somente com motivação técnico-científica. Durante esta análise, detectaram-se vulnerabilidades e fragilidades no sistema de votação e no projeto de segurança da informação. Em razão disto, este trabalho apresenta cenários onde estas vulnerabilidades e fragilidades podem ser exploradas com o intuito de promover uma fraude eleitoral, quebra de sigilo, ou, eventualmente, outros crimes de responsabilidade.

Palavras-chaves: Votação eletrônica; análise de segurança; UNIR.

www.rara.unir.br

SECURITY ANALYSIS OF THE INTERNET ELECTRONIC VOTING SYSTEM IMPLEMENTED AND ADOPTED BY FEDERAL UNIVERSITY OF RONDONIA

ABSTRACT

In this work we present an independent security analysis of the voting system implemented at Federal University of Rondonia. During our analysis, vulnerabilities and flaws were detected in the voting software and system deployment. As a result, this work presents scenarios where these problems can be explored in order to promote electoral fraud or, possibly, other responsibility crimes.

Keys words: Electronic voting; security analysis; UNIR.

1 INTRODUÇÃO

Nos últimos anos, assim como em diversas autarquias, órgãos e empresas públicas, a Fundação Universidade Federal de Rondônia (UNIR) intensificou sua política de informatização dos processos institucionais. Como exemplos desta informatização, podem-se citar a adoção do SEI (Sistema Eletrônico de Informações) para controle de processos e publicações eletrônicas (TRF4, 2020), a utilização do SOS (Sistema de Ordem de Serviço) como ferramenta de apoio à gestão (UNIR, 2020), e, ainda, a efetivação do SIGAA (Sistema Integrado de Gestão de Atividades Acadêmicas) para apoio à aprendizagem em ambiente virtual e gestão acadêmica (UFRN, 2020).

Contudo, é notório que a organização de consulta pública para reitor realizada pela Internet seja um verdadeiro marco neste processo de informatização (DTI/UNIR, 2020).

www.rara.unir.br

1.1 ELEIÇÕES ELETRÔNICAS PELA INTERNET

No contexto de informatização de processos, espera-se que todo e qualquer sistema eletrônico atenda aos princípios fundamentais de segurança da informação; ou seja, é necessária a presença de rotinas que forneçam confidencialidade, integridade e disponibilidade das informações (ABNT, 2013).

Complementarmente, ao se estabelecer os componentes básicos de um sistema eletrônico de votação e procedimentos relacionados, entende-se que a preocupação direta deva ser no incremento da segurança obtido, i.e., para que seja possível executar eleições confiáveis que conservem absolutamente o sigilo e a integridade das escolhas definidas pelo eleitor (ARANHA *et al.*, 2013; GRAAF, 2017).

Um outro aspecto relevante para qualquer sistema de votação é o objetivo da consulta que está sendo realizada. Isto ocorre porque este tipo de sistema pode ser utilizado para escolher presidentes ou autoridades de uma nação (SPRINGALL *et al.*, 2014); chefes ou representantes de uma organização (ADIDA; MARNEFFE; PEREIRA, 2020); ou até mesmo para definir a roupa de uma celebridade (FACEBOOK, 2020) ou quem irá lavar a louça em uma determinada residência (VOXVOTE, 2020). Assim, as preocupações com a segurança da informação e algoritmos criptográficos serão ajustados de acordo com o objetivo e a importância da votação que está sendo realizada.

De fato, as primitivas de segurança de segurança da informação e as preocupações com o objetivo são esperadas em todo e qualquer processo de votação democrático, e vêm sendo discutidas em diversas comunicações científicas especializadas, como, por exemplo, em ARANHA *et al.* (2013), Silva (2002), Brunazo *et al.* (2015), Gritzalis (2012), Krimmer *et al.* (2017), Hao e Ryan (2016), Graaf (2017), entre outros. E, ainda, mais especificamente em Adida, Marneffe e Pereira (2020), Springall *et al.* (2014) quando a plataforma eletrônica escolhida é a Internet.

Desta forma, além de levar em consideração preocupações genéricas da segurança da informação, sistemas eletrônicos de votação pela Internet devem atender a critérios específicos de seu contexto para que suas eleições sejam consideradas seguras e confiáveis.

www.rara.unir.br

1.2 OBJETIVO DESTE TRABALHO

O objetivo geral deste relatório é formalizar as observações realizadas pela subcomissão de auditagem (CONSUN/UNIR, 2020a) do Sistema de Eleições (SiE) desenvolvido e adotado pela Universidade Federal de Rondônia (UNIR). Mais especificamente, espera-se que este trabalho contenha uma análise técnico-científica das primitivas criptográficas do sistema de votação em análise, bem como discuta a (in)segurança das práticas e processos como um todo.

É importante salientar que o conteúdo e as conclusões aqui apresentados são de inteira responsabilidade dos autores e não representam de forma alguma a opinião do Departamento Acadêmico de Ciência da Computação (DACC) ou quaisquer outros órgãos, departamentos ou setores aos quais eventualmente os autores prestaram ou venham a prestar serviço. Além disto, destaca-se que esta análise não possui nenhuma motivação política, se tratando puramente de uma análise técnica para contribuir com os trabalhos da Comissão Eleitoral para reitor da UNIR– 2020.

1.3 MÉTODO E DESCRIÇÃO GERAL DOS TESTES

O método adotado durante o desenvolvimento deste estudo foi o da pesquisa exploratória (WAZLAWICK, 2008), uma vez que os autores, membros da subcomissão de auditagem, não tinham necessariamente uma noção de todos os problemas de segurança que

encontrariam. Assim, conforme foram tomando contato com o sistema de votação eletrônica sob análise, bem como com as técnicas utilizadas durante a sua implantação, os autores foram desenvolvendo estratégias de pesquisa e coletando informações sobre fragilidades e vulnerabilidades já conhecidas.

Vale ressaltar que, durante todo o processo de análise de segurança, a subcomissão de auditagem recebeu apoio da Comissão Eleitoral e da Diretoria de Tecnologia da Informação. Especificamente, foram compartilhadas diversas informações sobre o processo de desenvolvimento, documentação do sistema, código fonte, além da disponibilização de um ambiente de testes nos servidores da própria DTI.

1.4 ORGANIZAÇÃO DO DOCUMENTO

Para facilitar o entendimento do trabalho desenvolvido pela subcomissão, este relatório obedece a estrutura a seguir. A Seção 2 discute alguns modelos conceituais de adversário que podem ser considerados no contexto de votação eletrônica pela Internet. A Seção 3 expõe o conjunto de fragilidades e vulnerabilidades encontradas no sistema eleitoral da Fundação Universidade Federal de Rondônia, considerando diferentes cenários, baseados em cada modelo de adversário. Por fim, a Seção 4 apresenta as considerações finais sobre o trabalho de auditoria e fornece alguns apontamentos. Adicionalmente, o Apêndice A descreve as atividades realizadas durante este trabalho de auditoria.

www.rara.unir.br

2 MODELOS CONCEITUAIS DE ADVERSÁRIO

Nesta seção são descritos alguns modelos conceituais de adversário que podem ser considerados no contexto de votação eletrônica pela Internet. Não se presume exaurir todos os modelos adversariais possíveis, mas pretende-se fornecer parâmetros técnicos suficientes para que possa haver um alinhamento entre os objetivos da votação para Reitor da UNIR e as preocupações com os requisitos de segurança da informação e as primitivas criptográficas necessárias para este pleito (conforme discutido na seção 1.1).

Para isto, a organização adotada neste documento foi iniciar com uma abordagem mais restritiva para o adversário (Seção 2.1), passando por uma perspectiva intermediária (Seção 2.2), terminando com modelo adversarial significativamente poderoso (Seção 2.3).

Ademais, tem-se que, independentemente da abordagem adotada, é possível vislumbrar um adversário como sendo uma pessoa natural ou entidade contra a qual se luta ou se disputa algo. Por exemplo, em uma partida de futebol, adversário é o time contra o qual se joga; já em uma mesa de pôquer, os adversários de um jogador são todos os demais jogadores. Desta forma, para não haver dubiedade, neste trabalho adotaremos a seguinte definição para adversário:

Definição 1 *(Adversário). É um indivíduo ou serviço mal intencionado que usa suas habilidades para atacar o sistema de votação e, com isso, gerar alguma falha de segurança.*

2.1 CONFIANÇA INCONDICIONAL NO(S) ADMINISTRADOR(ES) E MANTENEDOR(ES) DO SISTEMA

Neste modelo mais restritivo de adversário, assume-se que todos os envolvidos com a manutenção, administração e desenvolvimento do sistema de votação são bem intencionados e lícitos. Assim, não existe sequer a curiosidade ou intenção em acessar as informações de modo não autorizado.

No contexto da eleição para Reitor da UNIR, isso seria o equivalente a presumir que os servidores da Diretoria de Tecnologia da Informação (DTI), alta gestão da Universidade Federal de Rondônia e demais usuários com acesso privilegiado aos dados sejam totalmente confiáveis.

Desta forma, o ataque pode ter origem somente de fontes externas, ou seja, eventos inesperados/naturais, usuários de fora da instituição ou usuários com privilégios mínimos (eleitor).

2.2 AUSÊNCIA DE CONFIANÇA NO(S) ADMINISTRADOR(ES), MANTENEDOR(ES) E ALGUM(NS) USUÁRIO(S) DO SISTEMA

Já neste modelo intermediário, a presunção de licitude começa a diminuir. Necessariamente, nem todos os envolvidos com a manutenção, administração e desenvolvimento do sistema de votação são inescrupulosos. Porém, considera-se que, dentre todos os usuários com acesso privilegiado ao sistema e/ou aos dados, possa haver algum

agente mal intencionado ou curioso, seja por razões de foro íntimo ou motivados por terceiros (e.g., via coerção ou suborno).

Neste modelo, portanto, o ataque pode ter origem também de fontes internas. No caso específico do sistema sob análise, além das fontes elencadas na seção anterior, os ataques também podem então ser perpetrados por servidores da DTI, alta gestão da UNIR e demais usuários com um nível de acesso mais elevado.

2.3 DESCONFIANÇA DE QUALQUER USUÁRIO DO SISTEMA

Complementarmente, neste modelo mais permissivo, de confiança zero (GILMAN; BARTH, 2017), não existe pressuposto de confiança incondicional em qualquer usuário do sistema. Assim, assume-se a possibilidade de quaisquer usuários cometerem equívocos que fragilizam a segurança do sistema como um todo, terem acesso a ferramentas de ataque automatizadas (*toolkit*), informações complementares sobre o sistema (documentação, manuais, diagramas, etc.), ou valerem-se de qualquer outros meios que comprometam a segurança do sistema de votação.

www.rara.unir.br

Neste cenário, o ataque pode ter origem de fontes internas, externas, mistas, ferramentas e softwares especializados, entre outros. Ou seja, o atacante pode utilizar diversas técnicas e plataformas para tentar sobrepujar a segurança do sistema.

3 FRAGILIDADES E VULNERABILIDADES

Em resumo, o resultado da análise é o descrito a seguir. O exame do código-fonte e documentação do sistema de eleição eletrônica pela Internet desenvolvido e implementado pela Diretoria de Tecnologia da Informação (DTI) da Universidade Federal de Rondônia (UNIR) evidenciou um conjunto de fragilidades e vulnerabilidades em componentes do *software*. Além disto, durante a execução de verificações de segurança realizadas no ambiente de testes disponibilizado pela DTI, foi observado que algumas rotinas podem impactar no sigilo do processo eleitoral como um todo. Assim, cada fragilidade e vulnerabilidade apresentada aqui representa um risco em potencial para a formulação de um método de ataque.

Para manter a uniformidade da organização deste relatório, as informações desta seção são apresentadas de forma similar às da seção anterior: inicia-se com fragilidades e vulnerabilidades que podem ser exploradas por um atacante com menos poderes (modelo adversarial descrito na Seção 2.1); passa-se por um atacante com privilégios e acessos intermediários (modelo adversarial descrito na Seção 2.2); e finaliza-se com um atacante com acesso diferenciado e/ou conhecimento de ferramentas especializadas (modelo adversarial descrito na Seção 2.3).

Destaca-se que, por definição, uma fragilidade ou vulnerabilidade encontrada em um cenário em que o modelo adversarial possui menos poderes e privilégios pode ser explorada em um cenário onde o atacante é mais poderoso.

3.1 CENÁRIO 1

Neste cenário foi adotado o modelo adversarial mais restrito (com menos poderes), descrito na Seção 2.1.

3.1.1 Disponibilidade Frágil

Sabe-se que, para haver disponibilidade, toda informação gerada ou adquirida por um indivíduo ou instituição deve estar à disposição de seus usuários legítimos no momento em que eles necessitem delas (SÊMOLA, 2014; ABNT, 2013).

No cenário em questão, espera-se que os sistemas de votação eletrônica pela Internet estejam disponíveis e funcionais durante todo o pleito eleitoral. Afinal, uma eventual falha de disponibilidade pode causar desistência do processo de votação e até perda de votos não armazenados.

Contudo, verificou-se que esporadicamente os sistemas da UNIR passam por problemas de instabilidade. Essa situação inclusive ocorreu durante o período de testes de segurança, mais especificamente no dia 27/06/2020.

Seja por desastres naturais ou eventos atípicos, é essencial que exista um plano de manutenção ou restabelecimento do serviço e contingenciamento de danos, para que o processo de votação não seja prejudicado e os serviços não fiquem indisponíveis por um tempo que prejudique o pleito. Todavia, a subcomissão de auditoria não identificou e também não teve acesso a qualquer plano desta natureza. Cabe notar, contudo, que os integrantes da

subcomissão não solicitaram especificamente qualquer artefato deste tipo durante o período de testes.

Recomendação: *Preparar e implementar um plano de manutenção e/ou restabelecimento do serviço e contingenciamento de danos robusto o suficiente para atender as especificidades do sistema de votação, possivelmente incluindo redundância de recursos computacionais.*

3.1.2 Escolha Inadequada de Algoritmos

3.1.2.1 Geração de números pseudo-aleatórios, assinatura e verificação de integridade dos votos

O sistema de votação eletrônica desenvolvido pela UNIR não utiliza qualquer gerador de números (pseudo-)aleatórios seguro para proteger o sigilo dos votos. Isto ocorre porque o Comprovante de Votação (número armazenado no banco de dados e exibido ao usuário no final da votação) é obtido pela função de hash MD5 (RIVEST; DUSSE, 1992) utilizando como entrada simplesmente um número inteiro (chamado de ID_USUARIO_ELEICAO) gerado sequencialmente pela classe GENERATIONTYPE do pacote JAVAX.PERSISTENCE da linguagem de programação Java (uma classe para gerar chaves primárias de tabelas de banco de dados, e, portanto, útil apenas para fins não criptográficos).

Ademais, o sistema também utiliza a função MD5 para fins de assinatura digital e verificação de integridade. Um primeiro equívoco nesta abordagem é que funções de hash não se prestam como mecanismo para gerar uma "assinatura digital" (BROWN; STALLINGS, 2017): como o cálculo de hashes não envolve qualquer informação secreta, qualquer usuário é capaz de calcular um hash para qualquer conjunto de dado do seu interesse, ferindo o princípio de inforjabilidade esperado de uma assinatura (seja ela digital ou manuscrita). Além disso, a função MD5 especificamente tem uso não recomendado desde 2005, quando se verificou que ela não fornecia a resistência a colisões esperada (WANG; YU, 2005), ficando recomendada como prudente a migração rápida para funções de hash mais seguras quando se tem por objetivo verificar a integridade de dados (NIST, 2019b; NIST, 2015; NIST, 2012).

Recomendação: *Utilizar um gerador de números pseudo-aleatórios de qualidade criptográfica, além de um algoritmo de assinatura digital de fato, como (EC)DSA (NIST, 2019a), e uma função de hash padronizada e resistente a colisões, como SHA-2 (NIST, 2015) ou SHA-3 (NIST, 2012). Caso o comprimento da cadeia de caracteres produzida como saída da função de hash seja crítico para a conferência por humanos, basta truncar a sua saída para um tamanho que seja adequado tanto do ponto de vista de usabilidade como de segurança.*

3.1.2.2 **Proteção de senhas**

Além da escolha inadequada de algoritmos para proteção do sigilo dos votos descrita anteriormente, verificou-se que, por funcionar de forma integrada ao SIGAA, o sistema de votação utiliza a função de hash MD5 para proteção das senhas de seus usuários ("hashs das senhas").

Apesar deste tipo de procedimento ser relativamente popular em sistemas com autenticação baseada em senhas, usar uma função de hash simples (e insegura, no caso do MD5) não é recomendado desde o final dos anos 90 (KALISKI, 2000; PROVOS; MAZIÈRES, 1999). A razão é que a maioria dos usuários utiliza sequências curtas e simples como senhas, fazendo com que a sua entropia (complexidade) seja muito menor do que o normalmente exigido para se obter níveis criptográficos de segurança (FLORENCIO; HERLEY, 2007). Isso acaba facilitando ataques do tipo "força bruta", como ataques de dicionário e busca exaustiva (HERLEY; OORSCHOT; PATRICK, 2009; CHAKRABARTI; SINGBAL, 2007), nos quais um atacante testa todas as senhas possíveis até determinar a correta.

Para evitar tais problemas, sistemas com este tipo de autenticação devem utilizar Esquemas de Hash de Senhas (*Password Hashing Scheme* - PHSs) para proteger-se (PHC, 2013). Basicamente, PHSs são algoritmos criptográficos que permitem a geração de uma sequência de bits pseudo-aleatórios a partir de uma senha e, ao mesmo tempo, aumentam o custo de ataques de força bruta. Esquemas como PBKDF2 (KALISKI, 2000) e bcrypt (PROVOS; MAZIÈRES, 1999), por exemplo, incluem um parâmetro configurável que controla o tempo necessário para o processo de derivação de chaves. Já esquemas mais modernos como Argon2 (BIRYUKOV; DINU; KHOVRATOVICH, 2016) e Lyra2

(ANDRADE *et al.*, 2016), além do controle do tempo de derivação, permitem uma melhor utilização dos recursos computacionais, e, consequentemente, aumentam a proteção contra ataques de força bruta.

Recomendação: *Utilizar um Esquema de Hash de Senhas criptograficamente seguro como Lyra2 (ANDRADE* et al.*, 2016) ou Argon2 (BIRYUKOV; DINU; KHOVRATOVICH, 2016), preferencialmente com uma estratégia de hashing de senhas no cliente e no servidor (CONTINI, 2015) para aliar segurança e desempenho.*

3.1.3 Autenticação sem Limite de Tentativas

Outra parte do sistema que pode sofrer ataques de "força bruta", além dos hashes das senhas descritos na Seção 3.1.2, são as páginas de autenticação/login. Neste caso, se o sistema não estabelecer qualquer tipo de limite de tentativas por endereço IP, nome de usuário, ou qualquer outro aspecto que identifique determinado acesso, o atacante pode testar um grande número de senhas possíveis até conseguir se autenticar (TEDESCO, 2017). Neste caso, a velocidade do ataque fica limitada basicamente à velocidade da conexão da rede entre o atacante e o servidor.

www.rara.unir.br

Desta forma, para prevenir-se deste tipo de ameaça, é necessário criar um mecanismo para limitar o número de vezes que um determinado usuário pode tentar se autenticar. Por exemplo, cada tentativa de acesso pode ser salva em um cache e se ultrapassado o limite estabelecido (*e.g.*, 6 tentativas), o usuário fica bloqueado por alguns minutos. Além disso, o uso de Esquemas de Hash de Senhas combinando computação no lado do cliente e no lado do servidor também auxiliam na tarefa de dificultar esse tipo de ataque (CONTINI, 2015).
Recomendação: *Criar um mecanismo eficiente para limitar o número de vezes que um determinado usuário pode tentar se autenticar.*

3.1.4 Possibilidade de Comprometimento do Sigilo do Voto

Conforme já discutido, o sigilo do voto deve ser uma das preocupações fundamentais de qualquer sistema de votação (ARANHA *et al.*, 2013; CONSUN/UNIR, 2020b). Assim,

implementar mecanismos que garantam esta primitiva se faz essencial em qualquer pleito eleitoral.

Contudo, conforme descrito a seguir, a subcomissão de auditoria formulou um estratégia de ataque que pode resultar no comprometimento do sigilo dos eleitores.

3.1.4.1 **Estratégia de ataque ao sigilo do voto *(usuário sem privilégios)***

Ainda na análise inicial, chamou atenção o fato da opção "Auditar" ficar habilitada para todos os usuários autenticados. Teoricamente, esta funcionalidade deve ser utilizada para que um usuário legítimo verifique se seu voto foi armazenado corretamente no sistema de votação. Assim, para realizar esta análise, basta informar o número do Comprovante de Votação para que o sistema exiba as informações do voto – pleito eleitoral, candidato(s) e cadeira(s).

Todavia, o sistema não limita esta auditoria/verificação somente aos votos do usuário que esteja fazendo a consulta. Desta forma, caso o usuário possua o número do Comprovante de Votação de outros eleitores, ele terá acesso às informações confidenciais do voto, violando, assim, o seu sigilo.

Cabe notar que ter acesso a este número e saber a qual usuário ele pertence não é necessariamente trivial. Porém, pelo fato de que esse número não é protegido adequadamente (conforme discutido na seção 3.1.2), foram formuladas algumas hipóteses para sua obtenção e/ou identificação, ação que pode resultar no comprometimento do sigilo voto.

Hipótese 1: *Mera exposição ou acesso ao Comprovante de Votação de um usuário específico*

Esta hipótese é a mais simples e banal, pois exige apenas que o atacante tenha acesso ao número do Comprovante de Votação de um usuário específico utilizando procedimentos do dia a dia.

Isto pode ocorrer de diversas formas, como por exemplo: através de publicações da tela final do voto em redes sociais, recolhimento da impressão do Comprovante de Votação, acesso a arquivos PDF salvos em computadores compartilhados, cache do navegador, entre outros meios corriqueiros que não aparentam oferecer ameaça inicial.

Assim, sabendo o código e a quem ele pertence, basta que o atacante se autentique no sistema de votação e informe o número do Comprovante de Votação para ter acesso aos dados sigilosos do voto.

Hipótese 2: *Acesso a lista de usuários do SIGAA e seus IDs*

Comumente, pelo sistema de votação eletrônica da Diretoria de Tecnologia da Informação (DTI), o cadastro dos eleitores é feito em massa. Conforme verificado no código fonte (em especial nas classes CADASTROELEITORESSERVICE e USUARIOELEICAODAO), para construção da estrutura de dados com a lista de eleitores é obedecida uma ordem crescente que usa como base o número de identificação (ID) do usuário no banco de dados do SIGAA. Com isto, seguindo esta ordem de identificação no SIGAA, cada usuário cadastrado em uma nova eleição recebe um número inteiro de identificação (chamado de ID_USUARIO_ELEICAO) gerado sequencialmente pela classe GENERATIONTYPE (conforme já discutido na Seção 3.1.2).

www.rara.unir.br

Desta forma, para haver a hipótese de comprometimento do sigilo do voto, o atacante precisará votar e recuperar o seu ID_USUARIO_ELEICAO a partir do próprio Comprovante de Votação; isso pode ser feito, por exemplo, com o auxílio da ferramenta <http://www.md5decrypt.org>.

Tabela 1 – Exemplo de recuperação do ID_USUARIO_ELEICAO.

Usuário	Comprovante de Votação	ID_USUARIO_ELEICAO
Isaac Newton	fae0b27c451c728867a567e8c1bb4e53	666

Em seguida, este usuário malicioso deverá conseguir acesso a lista de usuários do SIGAA e seus IDs; por exemplo, utilizando o Sistema Eletrônico do Serviço de Informações ao Cidadão (e-SIC) da UNIR (UNIR, 2016).

Tabela 2 – Exemplo de lista de usuários do SIGAA e seus IDs (em ordem alfabética).

Usuário	ID no SIGAA

Ada Lovelace	7
Alan Turing	71
Albert Einstein	3
Frida Kahlo	5
Hedy Lamarr	31
Isaac Newton	27
Marie Curie	55
Nikola Tesla	11
Noam Chomsky	29
Tiera Guinn	43

Com estas informações, o atacante pode cruzar a lista de usuários e IDs do SIGAA com a lista de eleitores homologados; como resultado, então, são descobertos todos os ID_USUARIO_ELEICAO a partir do seu próprio número de identificação na eleição.

Por fim, basta utilizar a função de hash MD5, passando como parâmetro de entrada os ID_USUARIO_ELEICAO descobertos, para obter o número do Comprovante de Votação de todos os usuários cadastrados na eleição. Isto posto, com a posse destas informações e sabendo a quem elas pertencem, basta que o atacante se autentique no sistema de votação e informe o número do Comprovante de Votação para ter acesso aos dados sigilosos do voto.

Hipótese 3: *Cadastro dos eleitores em ordem determinada*

Também pode ser verificado no código fonte que uma alternativa para o cadastro de eleitores em massa é o cadastro individual (*i.e.*, um eleitor por vez). Com esta funcionalidade, o usuário que esteja criando uma eleição (que não precisa necessariamente ser da DTI ou ter privilégios elevados) pode consultar o banco de dados do SIGAA e cadastrar os eleitores na sequência que lhe for conveniente. Desta forma, a estrutura de dados com a lista de eleitores, e consequentemente o número inteiro de identificação (ID_USUARIO_ELEICAO), serão gerados sequencialmente e na ordem em que foram cadastrados.

www.rara.unir.br

Tabela 3 – Exemplo de lista de eleitores homologados, com ID_USUARIO_ELEICAO recuperados (ordenada pelo ID_USUARIO_ELEICAO).

Usuário	ID no SIGAA	ID_USUARIO_ELEICAO
Frida Kahlo	5	663
Ada Lovelace	7	664
Nikola Tesla	11	665
Isaac Newton	27	666
Noam Chomsky	29	667
Hedy Lamarr	31	668
Alan Turing	71	669

Assim, na hipótese deste usuário querer atacar o sigilo do voto do sistema, será necessário apenas que ele vote e recupere o seu ID_USUARIO_ELEICAO a partir do próprio Comprovante de Votação (de maneira similar à discutida na hipótese anterior). Em seguida, será necessário que ele calcule o ID_USUARIO_ELEICAO dos demais eleitores com base na ordem de cadastro que ele tenha adotado.

Tabela 4 – Exemplo de lista de eleitores homologados, com Comprovante de Votação calculados (ordenada pelo ID_USUARIO_ELEICAO).

Usuário	ID_USUARIO_ELEICAO	Comprovante de Votação
Frida Kahlo	663	8757150decbd89b0f5442ca3db4d0e0e
Ada Lovelace	664	2291d2ec3b3048d1a6f86c2c4591b7e0
Nikola Tesla	665	84117275be999ff55a987b9381e01f96
Isaac Newton	666	fae0b27c451c728867a567e8c1bb4e53
Noam Chomsky	667	b5dc4e5d9b495d0196f61d45b26ef33e
Hedy Lamarr	668	192fc044e74dffea144f9ac5dc9f3395
Alan Turing	669	5c04925674920eb58467fb52ce4ef728

Por fim, bastará utilizar a função de hash MD5, passando como parâmetro de entrada os ID_USUARIO_ELEICAO descobertos, para obter o número do Comprovante de Votação de todos usuários cadastrados na eleição. Com a posse destas informações e sabendo a quem

elas pertencem, no- vamente basta que o atacante se autentique no sistema de votação e informe o número do Comprovante de Votação para ter acesso aos dados sigilosos do voto.

Recomendação: *Além das recomendações para proteção do Comprovante de Votação descritas na Seção 3.1.2, é imprescindível aprimorar a funcionalidade de "auditoria" e limitar o acesso dos usuários somente aos dados que lhes pertençam.*

3.1.5 Possibilidade de Coerção

Segundo Karlof, Sastry e Wagner (2005), outro requisito de segurança que deve ser considerado no desenvolvimento de sistemas de votação eletrônica é a Resistência à Coerção. Segundo este princípio, um eleitor não deve ter a capacidade de provar a terceiros fora do local de votação como ele votou (KARLOF; SASTRY; WAGNER, 2005), evitando, assim, os chamados "votos de cabresto".

Apesar de não ser interessante para todos cenários de votação eletrônica, uma vez que concorre com o princípio da verificabilidade (OLIVEIRA JUNIOR, 2017), cabe aos responsáveis pelo pleito eleitoral ponderar sobre a sua necessidade. Vale frisar que, no sistema de votação eletrônica da UNIR, a verificabilidade é implementada pela funcionalidade de "auditoria", conforme explicado nas seções anteriores.

Recomendação: *Ponderar sobre a necessidade de resistência à coerção para, então, revisar a forma de verificação do voto (funcionalidade de "auditoria").*

3.1.6 Pergunta de Segurança Ineficaz

Em simulações realizadas no ambiente de testes verificou-se que o sistema de votação busca implementar uma "camada extra" de proteção por meio de uma pergunta de segurança. Mais especificamente, a pergunta realizada ao eleitor é "Qual o nome da sua mãe?".

Apesar de parecer uma estratégia interessante, na prática este tipo de procedimento não é muito eficaz para adicionar segurança ao sistema. Isto ocorre porque a informação extra requisitada pertence à mesma classe de mecanismo de segurança das senhas (especificamente, autenticação baseada em algo que o usuário sabe) (BRANQUINHO *et al.*,

2014; ANDRADE *et al.*, 2016). Logo, se o atacante possui conhecimento suficiente para se autenticar usando as credenciais da vítima, é concebível que ela também tenha acesso ao nome de sua mãe (e.g., por meio de bases públicas de registros na Internet, incluindo redes sociais).

No caso do sistema implementado, a segurança deste procedimento é particularmente frágil devido à falta de um limite de tentativas de votação/resolução da pergunta de segurança. Mais precisamente, por não estabelecer este limite, o atacante pode realizar quantas tentativas forem necessárias para responder corretamente a pergunta. Contudo, o atacante necessita de no máximo 5 (cinco) tentativas, uma vez que o sistema fornece apenas esta quantidade de alternativas (escolhidas "aleatoriamente") e, logicamente, o nome correto estará entre aqueles que aparecerem em todas as tentativas (ou, comumente, será o único que sempre aparecerá). Consequentemente, basta observar os nomes que aparecem nas alternativas para sobrepujar a segurança deste procedimento.

www.rara.unir.br

Recomendação: *Criar um mecanismo eficiente para limitar o número de vezes que um determinado usuário pode tentar votar/responder a pergunta de segurança. Se optar por utilizar outro método de autenticação, preferencialmente adotar um mecanismo baseado "no que o usuário possui" (como um aparelho celular).*

3.2 CENÁRIO 2

Neste cenário foi adotado o modelo adversarial descrito na Seção 2.2, com acesso a dados e detalhes do sistema. Frisa-se que, conforme informações da Diretoria de Tecnologia da Informação (DTI), este não foi o modelo adversarial considerado durante o desenvolvimento do sistema de votação eletrônica pela Internet da UNIR. Contudo, os autores deste relatório consideram pertinente elencar os aspectos que seguem, para fundamentar as discussões da Comissão Eleitoral para Reitor; principalmente porque a literatura especializada indica que um número substancial de falhas de segurança são causadas por este tipo de atacante (ZADELHOFF, 2016; WRIT, 2018; VERIZON, 2019).

De qualquer modo, as fragilidades e vulnerabilidades explicadas nesta seção podem ser relevadas caso este tipo de atacante não seja considerada uma preocupação no sistema em questão.

3.2.1 Formulação Equivocada do Projeto de Segurança

Todo o projeto dos mecanismos de segurança utilizados (banco de dados, repositório de códigos, ambiente de desenvolvimento e demais artefatos) preocupa-se com atacantes externos e ignora totalmente o risco de atacantes internos.

Na realidade, conjectura-se que este trabalho de auditoria seja o primeiro a ter sido realizado por agentes externos à DTI, havendo indícios de que a detecção de comportamento malicioso por agentes internos esteja reduzida a processos de auditoria interna, também executados por humanos.

A adoção desta abordagem resulta em um relaxamento das preocupações com as rotinas ligadas ao desenvolvimento e manutenção do sistema, que por sua vez tornam-se vulnerabilidades que podem ser exploradas por agentes internos insatisfeitos ou com motivações ilícitas.

Alguns exemplos destas possibilidades são explanados nas próximas seções.

www.rara.unir.br

Recomendação: *Adotar mecanismos de segurança que resistam a agentes externos e, particularmente, a agentes internos que os conhecem em seus mínimos detalhes.*

3.2.2 Falta de Rastreabilidade

O conceito de rastreabilidade é amplo. A NBR ISO 9001 define rastreabilidade como a capacidade de rastrear o histórico, uso ou localização de uma entidade por meio de informação documentada (ABNT, 2015). Sob o enfoque de segurança da informação, o termo engloba as rotinas necessárias para encontrar todos os passos de um processo, desde a origem até o fim.

Complementarmente, tem-se que a rastreabilidade não é um fim em si mesma, pois é uma ferramenta que, em algumas circunstâncias, será utilizada para buscar informação ou garantir a veracidade de alguma ação quando isso for necessário (ROCHA, 2012). Além disso, sabe-se que a rastreabilidade possibilita medidas de vigilância, isolamento, mitigação e repreensão de atividades delituosas.

No contexto de votação eletrônica, rastreabilidade pode ser vista como o armazenamento e documentação de toda e qualquer ação executada pelo sistema, que pode ser utilizada para apuração de uma eventual anormalidade.

Apesar de ser de extrema importância, em análise ao código fonte, fora as rotinas padrões utilizadas pela linguagem de programação e banco de dados, não foi possível identificar qualquer mecanismo de rastreabilidade sendo implementado pelo sistema de votação da UNIR.

Recomendação: *Desenvolver e implementar mecanismo de rastreabilidade para apurar e combater ações ilícitas cometidas por usuários externos e, principalmente, usuários internos com acesso privilegiado ao sistema.*

www.rara.unir.br

3.2.3 Comprometimento do Sigilo do Voto

O armazenamento às claras das credenciais para acesso ao banco de dados (conforme verificado no código fonte da Classe CONNECTIONJDBC), atrelado ao fato de que a tabela de armazenamento do voto (cédula do voto, chamada de ELEICAO.VOTO) relaciona um determinado voto a seu eleitor por meio da chave estrangeira ELEICAO.VOTO.ID_USUARIO_ELEICAO, evidenciam que os mecanismos de segurança não são projetados para resistir a atacantes que disponham de informações privilegiadas (SANS, 2011).

Isto acontece porque este armazenamento inseguro, aliado ao conhecimento do modelo conceitual do banco de dados, possibilitam que alguém com acesso a estas informações realize consultas para recuperar o nome de todos eleitores e os códigos identificadores de seus candidatos. O resultado é, assim, o comprometimento do sigilo de todos os votos.

Para tornar essa discussão mais concreta, seguem abaixo alguns exemplos de consultas que podem ser realizadas para consultar o nome de todos eleitores e códigos identificadores de seus candidatos (Figura 1), para posteriormente recuperar o nome dos candidatos (Figura 2).

Recomendação: *Retirar as credenciais de acesso ao banco de dados do código fonte e, também, revisar a forma de armazenamento do voto, retirando informações que correlacionem as cédulas de votação a seus eleitores. Adotar também as sugestões apresentadas na Seção 3.2.2.*

Figura 1 – Exemplo de código SQL para consultar o nome de todos os eleitores e códigos identificadores de seus candidatos no sistema de votação eletrônica implementado pela UNIR.

```
   SELECT nome, eleicao.voto.id_candidato
FROMeleicao.usuario_sie
   WHERE      eleicao.usuario_sie.id_usuario = eleicao.usuario_eleicao.id_usuario
   AND        eleicao.voto.id_usuario_eleicao = MD5(eleicao.usuario_eleicao.id_usuario_eleicao) AND
      eleicao.voto.valido = true
```


Figura 2 – Exemplo de código SQL para consultar o nome dos candidatos no sistema de votação eletrônica implementado pela UNIR.


```
   SELECT nome, eleicao.voto.id_candidato
FROMeleicao.usuario_sie
   WHERE      eleicao.usuario_sie.id_usuario = eleicao.usuario_eleicao.id_usuario
   AND        eleicao.usuario_eleicao.id_usuario_eleicao = eleicao.candidato.id_usuario_eleicao
```

3.2.4 Possibilidade de Adulteração e Criação de Voto

Além da possibilidade de comprometimento do sigilo do voto discutida anteriormente, a proteção equivocada das informações descritas na Seção 3.2.3 também pode ocasionar um problema ainda maior, o da adulteração ou criação de voto.

Isto ocorre porque as mesmas credenciais que podem realizar consultas ao banco de dados também podem ser utilizadas para modificá-lo. O resultado seria, assim, o comprometimento da integridade dos votos.

Seguem abaixo alguns exemplos de comandos que podem ser executados para criar novos votos (Figura 3) ou adulterar/atualizar votos já depositados (Figura 4).

Figura 3 – Exemplo de código SQL para criar um novo voto no sistema de votação eletrônica implementado pela UNIR, onde "?" é um valor que pode variar de acordo com a necessidade do atacante.

```
INSERT INTO eleicao.voto
                                                    (ip, mac, data_voto, valor, id_candidato,
                                    id_situacao_voto, id_eleicao_cadeira, id_usuario_eleicao, id_tipo_eleitor)

VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)
```

Figura 4 – Exemplo de código SQL para adulterar/atualizar o candidato de um voto depositado no sistema de votação eletrônica implementado pela UNIR, onde "?" é um valor que pode variar de acordo com a necessidade do atacante.

```
                                    UPDATE
                    eleicao.voto SET id_candidato = ? WHERE
                     id_usuario_eleicao = ?
```


www.rara.unir.br

Recomendação: *Adotar sugestões apresentadas nas Seções 3.2.2 e 3.2.3.*

3.3 CENÁRIO 3

Neste cenário foi adotado o modelo adversarial descrito na Seção 2.3, com acesso diferenciado e/ou conhecimento de ferramentas especializadas.

3.3.1 Autenticação Vulnerável a Ataques Automatizados

Conforme discutido na Seção 3.1.3, o sistema de votação da UNIR não limita a quantidade de tentativas de acesso em sua página de autenticação/login. Desta forma, além de estar vulnerável a ataques de "força bruta" proferidos por usuários comuns (conforme discutido na seção supracitada), a página de autenticação também fica vulnerável a aplicações e bots que implementem ataques automatizados (TEDESCO, 2017). Neste contexto, destacam-se as ferramentas Hydra (HAUSER, 2020) e Burp (PORTSWIGGER, 2020) pela sua popularidade e interface amigável. Contudo, um usuário malicioso pode implementar sua própria aplicação ou script de ataque.

Para se proteger deste tipo de ameaça, podem ser usados alguns frameworks implementando soluções nativas, bastando que o programador as habilite. Além disso, se a página de autenticação/login estiver sob forte ameaça desse tipo de ataque, uma opção paliativa é usar um captcha (como o reCaptcha do Google). Por outro lado, se o ataque de força bruta estiver generalizado e vindo de muitos endereços IPs diferentes (ataque distribuído), talvez uma boa opção seja utilizar um serviço de segurança que trabalhe mitigando-os como um firewall ainda no DNS (como a Cloudflare ou nginx).

Recomendação: *Criar um mecanismo eficiente para limitar o número de vezes que um determinado usuário pode tentar se autenticar.*

3.3.2 Negação de Serviço

Seja na autenticação dos usuários (conforme discutido nas Seções 3.1.3 e 3.3.1), na resposta de perguntas no processo de votação (conforme discutido na Seção 3.1.6), ou na funcionalidade de "auditar"/verificar o voto (conforme discutido na Seção 3.1.4), a falta de limitação de acessos por usuário pode abrir brechas para ataques de negação de serviço ao sistema de votação da UNIR.

www.rara.unir.br

A negação de serviço (também conhecido como ataque de *DoS*, um acrônimo em inglês para *Denial of Service*) impede ou inibe o uso ou gerenciamento normal de instalações de comunicação ou serviços de rede específicos (BROWN; STALLINGS, 2017). Nesse tipo de ataque, é definido um alvo (por exemplo, o sistema de votação eletrônica pela Internet da UNIR), para receber uma infinidade de mensagens/solicitações pela rede. Não se trata de uma invasão do sistema, mas sim da sua sobrecarga, que pode ser realizada de apenas uma máquina ou de forma distribuída.

Em linhas gerais, utilizando troca de mensagens de rede, este ataque pode ser realizado de duas formas: (I) forçar o alvo do ataque a reinicializar ou consumir todos os recursos (*e.g.*, memória ou processamento) de forma que ele não possa mais fornecer seu serviço; (II) obstruir a rede de comunicação entre os utilizadores (legítimos ou não) e o sistema sob ataque de forma a não se comunicarem adequadamente.

Recomendação: *Além de adotar as sugestões apresentadas nas Seções 3.1.3, 3.1.4, 3.1.6 e 3.3.1, recomenda-se que seja utilizado algum serviço de segurança que mitigue ataques de negação de serviço, preferencialmente ainda no nível do DNS.*

3.3.3 Vulnerabilidades Herdadas

Além da escolha inadequada de algoritmos discutida na Seção 3.1.2, por funcionar de forma integrada ao SIGAA e utilizar diversas ferramentas e bibliotecas de produção, o sistema de votação eletrônica pela Internet da UNIR, em especial o seu módulo de autenticação, pode apresentar vulnerabilidades herdadas.

Desta forma, é essencial que a equipe de desenvolvimento realize estudos bibliográficos constantes a fim de levantar o estado da arte das vulnerabilidades das ferramentas utilizadas.

Batista *et al.* (2016) é um bom exemplo de trabalho que realizou este tipo de estudo. Nele, realizaram-se testes de penetração e verificou-se que o SIGAA precisa (BATISTA *et al.*, 2016): "ativar *plugins Anti-XSS*, além de uma programação preventiva baseada em recomendações da OWASP. Também se verificou a necessidade de boas práticas de codificação que tratem os erros somente do lado do servidor, evitando dar informações sobre a tecnologia empregada, assim como de higienização de código e verificações constantes. O sistema também não deve habilitar o recurso autocompletar em campos de autenticação[...], além de melhorar no quesito gerenciamento de sessão para evitar a janela de oportunidade existente quando o usuário logado se ausenta do computador por períodos prolongados de tempo."

www.rara.unir.br

Recomendação: *Realizar estudos bibliográficos constantes a fim de verificar as fragilidades das ferramentas utilizadas, bem como implementar as medidas de segurança necessárias para corrigir vulnerabilidades herdadas destas ferramentas.*

3.3.4 Compartilhamento de Credenciais de Autenticação

Em diversos departamentos e setores da Universidade Federal de Rondônia criou-se o hábito dos chefes ou responsáveis pela unidade compartilharem suas credenciais de acesso

ao sistemas institucionais com técnicos administrativos, estagiários e outros colaboradores que possam auxiliá-los nas tarefas cotidianas.

Apesar de não parecer uma ameaça direta ao sistema de votação, este tipo de prática aliada a (1) outras fragilidades do sistema, como falta de rastreabilidade (descrita na Seção 3.2.2) e pergunta de segurança ineficaz (descrita na Seção 3.1.6), e (2) a característica de que é possível realizar diversos votos no mesmo pleito eleitoral, ficando como válido somente o último, permite que (3) um colaborador malicioso vote como se fosse outra pessoa.

Recomendação: *Além de adotar as sugestões para proteção de senhas apresentadas na Seção 3.1.2, recomenda-se implementar um mecanismo de autenticação baseado em duas etapas a fim de reduzir o risco desse tipo de prática.*

www.rara.unir.br

3.3.5 Vulnerabilidade nos Certificados Digitais Utilizados

Para proteger suas comunicações em sistemas web, a Universidade Federal de Rondônia utiliza o Certificado Digital Corporativo da Infraestrutura de Chaves Públicas para Ensino e Pesquisa – ICPEdu, emitido pela Rede Nacional de Ensino e Pesquisa – RNP (RNP, 2020).

Todavia, recentemente foi descoberta uma vulnerabilidade que afeta mais de 270 certificados brasileiros (SLEEVI, 2020), incluindo diversos certificados do tipo utilizado pela UNIR.

Recomendação: *Verificar se o certificado digital que será utilizado pelo sistema de votação durante o pleito eleitoral não encontra-se na lista de certificados vulneráveis. Caso encontre-se, providenciar um novo certificado para proteger a comunicação do sistema de votação.*

4 CONSIDERAÇÕES FINAIS

Este relatório apresentou um conjunto de fragilidades e vulnerabilidades que evidenciam falhas de segurança no sistema de votação eletrônica pela Internet implementado

e adotado pela UNIR (ou sistema de consulta à comunidade, como vem sendo chamado em algumas comunicações oficiais). As consequências dessas falhas foram discutidas ao longo do texto sob diferentes modelos conceituais de adversário. Em particular, dependendo do modelo adversarial, mostrou-se possível que um atacante explore fragilidades do sistema para autenticar-se de maneira indevida, fragilize a disponibilidade do sistema de votação, e até comprometa o sigilo do voto.

Desta forma, é essencial que a Comissão Eleitoral defina o tipo de adversário que será considerado no pleito eleitoral para Reitor 2020, para que a Diretoria de Tecnologia da Informação promova as alterações seguindo as recomendações apresentadas.

Contudo, além das correções nas primitivas de segurança, espera-se que a equipe de desenvolvimento da Universidade Federal de Rondônia fique atenta a algumas fragilidades comuns no processo de desenvolvimento de *softwares* destinados à votação. Complexidade acentuada, auditoria externa insuficiente, ausência de análise estática de código, ausência de exercícios internos, falta de treinamento formal, disponibilização de dados críticos aos investigadores, desconhecimento da literatura relevante, e falsa sensação de segurança são exemplos de erros comuns que levam ao desenvolvimento de soluções frágeis (ARANHA *et al.*, 2013; CALANDRINO *et al.*, 2007).

www.rara.unir.br

Além disso, torna-se evidente a necessidade de se utilizar recursos para avaliação científica, independente e contínua das soluções de segurança adotadas pelo Fundação Universidade Federal de Rondônia. Isto é ainda mais relevante havendo disponibilidade de especialistas internos e externos, bem como cursos de computação e suas tecnologias, capazes de contribuir na direção do incremento real das propriedades de segurança destas soluções. Neste sentido, recomenda-se que a DTI busque estabelecer vínculos com estes parceiros a fim de aprimorar o processo de desenvolvimento de soluções como a de votação eletrônica.

AGRADECIMENTOS

Os autores gostariam de agradecer ao Prof. Dr. Diego F. Aranha da Aarhus University e ao Prof. Dr. Marcos A. Simplicio Jr da Universidade de São Paulo por revisarem este relatório, e pelas sempre interessantes discussões a respeito de criptografia, segurança da informação e principalmente votação eletrônica.

REFERÊNCIAS

ABNT. **ABNT NBR ISO/IEC 27001:2013 – Tecnologia da informação – Técnicas de segurança – Sistemas de gestão da segurança da informação – Requisitos.** Rio de Janeiro/RJ, Brasil, 2013.

_____. **ABNT NBR ISO 9001:2015 – Sistema de Gestão da Qualidade – Requisitos**. Rio de Janeiro/RJ, Brasil, 2015.

ADIDA, B.; MARNEFFE, O. de; PEREIRA, O. **Helios Election System**. 2020. Disponível em: <https://heliosvoting.org>. (Acesso em: 6 de julho de 2020).

ANDRADE, E. R.; SIMPLICIO JR, M. A.; BARRETO, P. S. L. M.; SANTOS, P. C. F. d. Lyra2: efficient password hashing with high security against time-memory trade-offs. **IEEE Transactions on Computers**, PP, n. 99, 2016. ISSN 0018-9340. See also <http://eprint.iacr.org/2015/136>.

ARANHA, D. F. *et al*. **Vulnerabilidades no software da urna eletrônica brasileira**. 2013. 40 p. Disponível em: <https://sites.google.com/site/dfaranha/projects/relatorio-urna.pdf?attredirects=0>. (Acesso em: 6 de julho de 2020).

BATISTA, R. R.; SANTOS, C. G. dos; ARAÚJO, S. G. L.; ARAÚJO, W. J. de. Teste de invasão no sigaa da ufpb. **GESTÃO. Org**, Universidade Federal de Pernambuco, v. 14, n. 5, p. 247–254, 2016.

BIRYUKOV, A.; DINU, D.; KHOVRATOVICH, D. **Argon2: the memory-hard function for password hashing and other applications**. v1.3 of argon2. Luxembourg, 2016. <https://github.com/P-H-C/phc-winner-argon2/blob/master/argon2-specs.pdf>.

BRANQUINHO, M.; SEIDL, J.; MORAES, L. de; BRANQUINHO, T.; AZEVEDO, J. de. **Segurança de Automação Industrial e SCADA**. Elsevier Editora Ltda., 2014. ISBN 9788535277876.

BROWN, L.; STALLINGS, W. **Segurança de Computadores: Princípios e Práticas**. Elsevier Editora Ltda, 2017. ISBN 9788535264500.

BRUNAZO, A.; CARVALHO, M.; TEIXEIRA, M.; JR, M. S.; FERNANDES, C. Auditoria especial no sistema eleitoral 2014. In: **Anais do XI Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg'15) – II Workshop de Tecnologia Eleitoral (WTE)**. Porto Alegre/RS, Brazil: Sociedade Brasileira de Computação (SBC), 2015. v. 13, p. 511–522. Disponível: <http://sbseg2015.univali.br/anais/wte.html>. Ver também: <http://www.brunazo.eng.br/voto-e/arquivos/RelatorioAuditoriaEleicao2014-PSDB.pdf>.

CALANDRINO, J. A.; FELDMAN, A. J.; HALDERMAN, J. A.; WAGNER, D.; YU, H.; ZELLER, W. P. Source code review of the diebold voting system. **University of California, Berkeley under contract to the California Secretary of State**, 2007.

www.rara.unir.br

CHAKRABARTI, S.; SINGBAL, M. Password-based authentication: Preventing dictionary attacks. **Computer**, v. 40, n. 6, p. 68–74, june 2007. ISSN 0018-9162.

CONSUN/UNIR. **Ato Nº 11, de 17 de junho de 2020 – CONSUN/UNIR**. 2020. Disponível em: <https://sei.unir.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_ externo=0>. Código Verificador: 0440163. Código CRC: 2D4DA0FE. (Acesso em: 26 de junho de 2020).

_____. **Resolução Nº 213, de 8 de junho de 2020 – CONSUN/UNIR**. 2020. Disponível em: <https: //sei.unir.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_exter no=0>. Código Verificador: 0434887. Código CRC: 030C7184. (Acesso em: 8 de julho de 2020).

CONTINI, S. **Method to Protect Passwords in Databases for Web Applications**. 2015. Cryptology ePrint Archive, Report 2015/387. <https://eprint.iacr.org/2015/387>.

DTI/UNIR. **Despacho DTI/UNIR, de 24 de abril de 2020**. 2020. Disponível em: <https: //sei.unir.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_exter no=0>. Código Verificador: 0411579. Código CRC: 4BE93254. (Acesso em: 26 de junho de 2020).

FACEBOOK. **Quiz – Facebook**. 2020. Disponível em: <https://pt.quizur.com/type/trivia>. (Acesso em: 7 de julho de 2020).

FLORENCIO, D.; HERLEY, C. A Large-scale Study of Web Password Habits. In: **Proceedings of the 16th International Conference on World Wide Web**. New York, NY, USA: ACM, 2007. p. 657–666. ISBN 978-1-59593-654-7.

GILMAN, E.; BARTH, D. **Zero Trust Networks**. O'Reilly Media, Incorporated, 2017.

GRAAF, J. van de. **O mito da urna: desvendando a (in)segurança da urna eletrônica (Versão 1)**. Creative Commons, 2017.

GRITZALIS, D. **Secure Electronic Voting**. Springer US, 2012. (Advances in Information Security). ISBN 9781461502395.

HAO, F.; RYAN, P. **Real-World Electronic Voting: Design, Analysis and Deployment**. CRC Press, 2016. (Series in Security, Privacy and Trust). ISBN 9781315354118.

HAUSER, V. **Hydra THC**. 2020. <https://github.com/vanhauser-thc/thc-hydra>.

HERLEY, C.; OORSCHOT, P. V.; PATRICK, A. Passwords: If We're So Smart, Why Are We Still Using Them? In: **Financial Cryptography and Data Security**. Berlin, Germany: Springer Berlin Heidelberg, 2009.

KALISKI, B. **PKCS#5: Password-Based Cryptography Specification version 2.0 (RFC 2898)**. RSA Laboratories. Cambridge, MA, USA, 2000. <http://tools.ietf.org/html/rfc2898>.

www.rara.unir.br

KARLOF, C.; SASTRY, N.; WAGNER, D. A. Cryptographic voting protocols: A systems perspective. In: **USENIX Security Symposium**. 2005. v. 12, p. 39.

KRIMMER, R.; VOLKAMER, M.; BARRAT, J.; BENALOH, J.; GOODMAN, N.; RYAN, P.; TEAGUE, V. **Electronic Voting: First International Joint Conference, E-Vote-ID 2016, Bregenz, Austria, October 18-21, 2016, Proceedings**. [S.l.]: Springer International Publishing, 2017. (Lecture Notes in Computer Science). ISBN 9783319522401.

NIST. **NIST Interagency or Internal Reports 7896: Third-Round Report of the SHA-3 Cryptographic Hash Algorithm Competition**. 2012.

_____. **FIPS 180-4: Secure hash standard**. 2015.

_____. **FIPS 186-5: Digital Signature Standard (DSS)**. 2019.

_____. **NIST Special Publication 800-131A Rev.2 - Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths.** 2019.

OLIVEIRA JUNIOR, N. d. **Sistema de eleição seguro o suficiente**. Trabalho de Conclusão de Curso, 2017.

PHC. **Password Hashing Competition**. 2013. <https://password-hashing.net/>.

PORTSWIGGER. **Burp Suite Community Edition**. 2020. <https://portswigger.net/burp/communitydownload>.

PROVOS, N.; MAZIÈRES, D. A future-adaptable password scheme. In: **Proc. of the FREENIX track: 1999 USENIX annual technical conference**. Monterey, California, USA: USENIX, 1999.

RIVEST, R.; DUSSE, S. **The MD5 message-digest algorithm**. MIT Laboratory for Computer Science, 1992.

RNP. **Conheça o Certificado Corporativo da ICPEdu**. 2020. Disponível em: <https://www.rnp.br/servicos/alunos-e-professores/identidade-e-seguranca/icpedu>. (Acesso em: 14 de julho de 2020).

ROCHA, R. U. G. d. **Fluxo da informação no sistema de rastreabilidade em uma empresa do segmento eletrônico**. Tese (Doutorado), 2012.

SANS. **CWE/SANS TOP 25 Most Dangerous Software Errors**. 2011. Disponível em: <https://www.sans.org/top25-software-errors>. (Acesso em: 16 de julho de 2020).

SÊMOLA, M. **Gestão da segurança da informação: uma visão executiva**. 2. ed. Rio de Janeiro: Elsevier, 2014. ISBN 978-85-352-7178-2.

SILVA, M. C. da. **Voto eletrônico: é mais seguro votar assim?** Editora Insular, 2002.

SLEEVI, R. **SECURITY RELEVANT FOR CAs: The curious case of the Dangerous Delegated Responder Cert**. 2020. Disponível em: <https://www.mail-archive.com/dev-security-policy@lists. mozilla.org/msg13493.html>. (Acesso em: 14 de julho de 2020).

www.rara.unir.br

SPRINGALL, D.; FINKENAUER, T.; DURUMERIC, Z.; KITCAT, J.; HURSTI, H.; MACALPINE, M.; HALDERMAN, J. A. **Independent Report on E-voting in Estonia**. 2014. Disponível em: <https://estoniaevoting.org>. (Acesso em: 6 de julho de 2020).

TEDESCO, K. **Checklist de segurança para autenticação**. 2017. <https://www.treinaweb.com.br/ blog/checklist-de-seguranca-para-autenticacao/>. (Acesso em: 9 de julho de 2020).

TRF4. **Sistema Eletrônico de Informações (SEI) – UNIR**. 2020. Disponível em: <https: //sei.unir.br/>. (Acesso em: 6 de julho de 2020).

UFRN. **Sistema Integrado de Gestão de Atividades Acadêmicas (SIGAA) – UNIR**. 2020. Disponível em: <https://sigaa.unir.br/>. (Acesso em: 6 de julho de 2020).

UNIR. **SIC – Portal de "Acesso à Informação" da UNIR**. 2016. <http://www.sic.unir.br>.

_____. **Sistema de Ordem de Serviço – SOS**. 2020. Disponível em: <http://sistemas.unir.br/sos/>. (Acesso em: 6 de julho de 2020).

VERIZON. **2019 Data Breach Investigations Report**. 2019. Disponível em: <https://enterprise.verizon.com/resources/executivebriefs/2019-dbir-executive-brief.pdf>. (Acesso em: 16 de julho de 2020).

VOXVOTE. **VoxVote free and easy Mobile Voting tool for ANY speaker or teacher**. 2020. Disponível em: <https://www.voxvote.com>. (Acesso em: 7 de julho de 2020).

WANG, X.; YU, H. How to Break MD5 and Other Hash Functions. In:_____. **24th Annual International Conference on the Theory and Applications of Cryptographic Techniques**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.

WAZLAWICK, R. S. **Metodologia de pesquisa para ciência da computação**. Rio de Janeiro: Elsevier, 2008. 184 p. ISSN 978-85-352-3522-7.

WRIT. **Workshop on Research for Insider Threats – Welcome page**. 2018. Disponível em: <https://www.ieee-security.org/TC/SPW2018/WRIT/>. (Acesso em: 16 de julho de 2020).

ZADELHOFF, M. van. **The Biggest Cybersecurity Threats Are Inside Your Company**. 2016. Disponível em: <https://hbr.org/2016/09/the-biggest-cybersecurity-threats-are-inside-your-company>. (Acesso em: 16 de julho de 2020).

www.rara.unir.br