(In)segurança do voto eletrônico no Brasil

DIEGO F. ARANHA
MARCELO M. KARAM
ANDRÉ DE MIRANDA
FELIPE B. SCAREL

1. INTRODUÇÃO

■ O Brasil vem adotando crescente informatização de suas eleições desde o ano de 1996, culminando no cenário atual em que se vislumbra a instalação de dispositivos de identificação biométrica em todos os equipamentos de votação. Marcos importantes na história da iniciativa foram a realização das primeiras eleições completamente eletrônicas em 2000, a transferência da responsabilidade exclusiva do desenvolvimento de *software* para a audoridade eleitoral a partir de 2006 e a adoção de um sistema operacional auditável (*GNU/Linux*) a partir de 2008. Ao se estabilizar os componentes básicos do sistema eletrônico de votação e procedimentos relacionados, entende-se que a preocupação direta deve ser o incremento de segurança, para que seja possível executar eleições confiáveis que conservem o sigilo e a integridade das escolhas definidas pelo eleitor. Uma iniciativa louvável nesta direção é a realização desde 2009 de testes periódicos e públicos de segurança que permitem, ainda que com algumas restrições indesejáveis, a equipes de especialistas da academia e indústria avaliar de forma independente a segurança dos mecanismos adotados pelo sistema eletrônico de votação.

O objetivo geral deste artigo é formalizar as observações realizadas pela equipe de autores, enquanto participantes e vencedores da 2ª edição dos Testes Públicos de Segurança organizados pelo Tribunal Superior Eleitoral (TSE), tendo como motivação principal delinear as limitações do sistema eletrônico de votação adotado no Brasil e contribuir para a evolução do seu processo de segurança. Seguindo políticas padronizadas de divulgação de vulnerabilidades utilizadas

na área de Segurança da Informação, são apresentadas descrições suficientes das fragilidades e problemas de processo encontrados, acompanhadas de múltiplas sugestões de correção. Desta forma, a parte interessada encontra-se em posição adequada para implementar contramedidas efetivas. Em particular, este relatório versa sobre os principais problemas de projeto e/ou implementação de mecanismos de segurança detectados no *software* da urna eletrônica, mas pode-se observar de antemão que vários dos recursos implementados não representam realmente mecanismos de *segurança*, mas apenas de *ofuscação*, não resistindo a embustes montados por colaboradores internos ou atacantes persistentes. Como vários dos problemas encontrados resultam de falhas arquiteturais ou premissas inadequadas de projeto, é improvável que a intervenção pontual em algumas dessas questões resolva as causas fundamentais para a sua ocorrência. É imprescindível que se execute revisão crítica completa dos processos de desenvolvimento de *software* para que se estabeleçam boas práticas que tenham condições de evitar que novas vulnerabilidades sejam inseridas acidentalmente ou intencionalmente por agentes maliciosos internos ou externos.

Como o modelo de urna eletrônica adotado no Brasil depende exclusivamente da integridade do *software* para se atingir integridade dos resultados, os problemas discutidos aqui adquirem caráter crítico e exigem urgência na introdução de mecanismos que permitam a auditabilidade de resultados. Apenas com uma revisão de práticas e instalação de metodologia científica para avaliação contínua do sistema, é possível que o *software* da urna eletrônica satisfaça requisitos mínimos e plausíveis de segurança e transparência. É importante salientar ainda que o presente estudo trata apenas do *software* da urna eletrônica, não se manifestado a respeito dos aspectos físicos ou do *hardware* do equipamento. Esta decisão foi tomada respeitando-se os campos de especialidade dos autores. Ainda assim, também vale ressaltar que as observações coletadas referem-se apenas a uma pequena – ainda que estratégica – fração do código-fonte do *software* de votação, excluídos também outros componentes de *software* que constituem o sistema de votação do qual a urna faz parte, visto que as regras do evento, e o limite de tempo na participação dos investigadores, não permitiram uma avaliação mais detalhada.

2. DEFINIÇÕES PRELIMINARES

■ Sistemas de votação, eletrônicos ou não, precisam satisfazer alguns requisitos mínimos para serem úteis na prática. As duas principais propriedades de segurança de um sistema de votação referem-se ao anonimato e à destinação dos votos:

- *Sigilo*: os votos devem ser secretos, de maneira a prevenir sua venda e defender eleitores de coação por qualquer parte interessada;
- *Integridade*: os votos devem refletir a intenção dos eleitores individualmente, e sua apuração e totalização deve transferir a intenção coletiva dos eleitores para o resultado. Qualquer tentativa de violar a integridade de uma eleição deve ser detectável e corretamente atribuída.
- Em alguns sistemas, essas propriedades são alcançadas através da integração de *primitivas criptográficas*, que muitas vezes dependem de *dados aleatórios* para fornecer segurança. A *cifração* é uma transformação de *sigilo* que permite a leitura do conteúdo original apenas para os detentores de uma *chave criptográfica*. A *assinatura digital* é um mecanismo análogo à assinatura de punho que permite confirmar a *origem* e *autenticidade* de um documento ou programa. Outros requisitos de segurança interferem diretamente no sigilo do voto e integridade dos resultados, portanto é importante enumerar algumas propriedades relacionadas às duas principais:
- *Habilitação*: apenas eleitores legítimos podem votar, e uma única vez;
- *Equidade*: resultados não devem ser antecipados para influenciar os eleitores restantes;
- *Resistência à coação:* um eleitor não deve receber nenhum comprovante ou recibo que possa ser utilizado para provar suas escolhas, nem ser capaz de cooperar com outra parte para prová-las;
- *Verificação independente*: um eleitor deve ser capaz de verificar que seu voto foi registrado e contabilizado corretamente;
- *Independência de software:* erros não detectados no *software* não podem causar erros indetectáveis no resultado [1].
- Os equipamentos de votação podem ser classificados em diferentes modelos, organizados em níveis crescentes de transparência e decrescentes de dependência de *software* [2]:
- *Armazenamento eletrônico direto (DRE – Direct Recoding Electronic, 1***a** *Geração):* os votos são armazenados e contabilizados de maneira puramente eletrônica, impedindo assim qualquer possibilidade de recontagem ou de verificação independente dos resultados, pois a adulteração não detectada do *software* causa distorções indetectáveis nos resultados;
- *Voto impresso conferível pelo eleitor (VVPT – Voter Verified Paper Trail, 2***a** *Geração):* os votos são impressos para verificação independente pelo eleitor e apuração posterior, sem no entanto funcionarem como comprovantes de suas escolhas;

- *Verificabilidade fim-a-fim (E2E – End-to-end Verifiability, 3***a** *Geração):* os eleitores podem verificar que seus votos foram registrados e contabilizados corretamente e que todos os votos foram incluídos no resultado final.

2. VISÃO SUPERFICIAL

■ A urna eletrônica brasileira pode ser classificada como um modelo do tipo *DRE*, com registro puramente eletrônico dos votos. Em termos gerais, uma eleição utilizando o sistema eletrônico brasileiro de votação emprega as seguintes etapas de preparação:

- Lacração dos componentes de *software* e produção de mídias de carga;
- Instalação do *software* nas urnas eletrônicas a partir das mídias de carga;
- Distribuição das urnas às respectivas seções eleitorais.
- No dia determinado para realização das eleições, cada urna eletrônica deve executar uma sequência bem-definida de procedimentos:
- Impressão da *zerésima*, documento oficial que supostamente atesta que nenhum voto foi previamente computado para qualquer candidato;
- Abertura da votação pelo mesário responsável;
- Acesso dos eleitores à urna eletrônica para que suas escolhas sejam inseridas;
- Encerramento da votação, realizada também pelo mesário responsável;
- Emissão de vias do Boletim de Urna (BU) em papel, contendo a totalização parcial dos candidatos;
- Gravação autenticada dos chamados produtos públicos de votação, abrangendo principalmente as versões digitais do BU, arquivo de registro cronológico de eventos (LOG) e Registro Digital do Voto (RDV);
- Rompimento do lacre e retirada pelo mesário da Mídia de Resultados (MR) contendo os produtos públicos da eleição;
- Transmissão dos produtos públicos para o totalizador a partir de rede privada de comunicação.

O papel do totalizador consiste em combinar todas as totalizações parciais no resultado declarado como oficial.

3. FRAGILIDADES

■ O exame do código-fonte do *software* da urna eletrônica evidenciou um conjunto de fragilidades em componentes críticos do *software*. Cada fragilidade

apresentada aqui representa uma vulnerabilidade em potencial que permite a um agente interno ou externo formular uma metodologia de ataque. A presença de fragilidades, até mesmo em componentes críticos do *software*, atesta a presença de fragilidades no próprio processo de desenvolvimento de *software* utilizado.

3.1 No Software

A seguir, discutimos as fragilidades encontradas no *software*, algumas já anteriormente discutidas no Relatório elaborado pela Sociedade Brasileira de Computação em 2002 [3], ou na análise acadêmica do *software* de votação das máquinas utilizadas nos Estados Unidos e fabricadas pela Diebold [4], mesma companhia que fabrica o *hardware* das urnas brasileiras e produziu as versões iniciais do *software* de votação.

Proteção inadequada ao sigilo do voto

Desde a promulgação da lei eleitoral 9.504/97 [5], que oficializou a votação eletrônica com o modelo atual de urna DRE, o voto impresso verificável pelo eleitor foi instituído no Brasil pela primeira vez em 2002, através da lei 10.408/02 [6]. A finalidade desse recurso é permitir para todos os eleitores, agentes com maior interesse no processo democrático de votação, a possibilidade de verificação independente do seu voto. Sem verificação independente, a confiança é depositada apenas na habilidade dos partidos políticos em fiscalizar a confecção dos programas e na boa fé dos técnicos do TSE em produzir *software* correto [3], visto que depende apenas do *software* a contagem honesta dos votos. A proposta do voto impresso sugere então produzir uma versão materializada do voto, que pode ser conferida pelo eleitor, sem, no entanto, permitir que o próprio comprove suas escolhas para uma parte interessada qualquer. Após alegações de dificuldades operacionais e alto custo por parte do TSE, o voto impresso terminou descontinuado pela lei 10.740/03 [7]. Em seu lugar, adotou-se um substituto puramente digital.

O Registro Digital do Voto, ou RDV, foi introduzido para supostamente permitir a mesma capacidade de verificação independente dos resultados da urna. Por essa razão, é um documento público disponibilizado para os partidos após as eleições. Entretanto, enquanto o voto impresso permite de fato a verificação independente dos votos computados eletronicamente, o RDV é produzido pelo mesmo componente de *software* que produz o Boletim de Urna (BU) contendo os totais de cada candidato computados pela urna. Por essa razão, a possibilidade de adulteração do BU implica diretamente na possibilidade de adulteração do RDV,

o que significa que o RDV se qualifica apenas como informação redundante, tão passível de ataque quanto aquilo que tenta proteger. Além disso, o próprio projeto da urna não elimina completamente a possibilidade de se vincular a identidade do eleitor ao seu voto através de *software* adulterado [3], visto que ambos os equipamentos que coletam essas informações estão conectados eletronicamente.

O RDV consiste em uma tabela, separada por cargos em disputa eleitoral, que armazena de maneira desordenada os votos propriamente ditos inseridos pelos eleitores na urna eletrônica. O objetivo do embaralhamento dos votos é desassociar a ordem em que os votos foram inseridos da ordem em que foram armazenados. Durante os Testes Públicos de Segurança, a qualidade desse mecanismo foi extensamente analisada, visto que a ordem em que os votos são armazenados precisa atingir alto rigor de aleatoriedade, e apenas um profissional com algum treinamento básico na área de Segurança Computacional observaria que o mecanismo de embaralhamento é tão crítico para o sigilo do voto quanto a integridade do *software* é para a integridade dos resultados. Com apenas algumas buscas por funções conhecidamente inseguras na primeira hora de exame do código-fonte, observou-se que o mecanismo de embaralhamento foi projetado e implementado utilizando uma progressão de erros que terminou por permitir a sua reversão. A implementação utiliza um *gerador de números pseudo-aleatórios*, procedimento computacional que produz uma sequência de números aparentemente aleatórios, mas que pode ser unicamente determinada a partir de um pequeno parâmetro chamado *semente,* que precisa ser escolhido de forma verdadeiramente imprevisível. Para fins de segurança, a semente deve também ser mantida em segredo.

Na construção concretizada no *software*, observou-se escolha inadequada tanto do gerador, que não apresentava qualidade criptográfica, quanto da semente, que consistia em uma simples tomada de tempo com precisão de segundos executada na inicialização do sistema de votação. A semente era ainda tornada pública mediante registro em LOG e impressão em documento oficial, a zerésima. De posse da hora de emissão da zerésima, foi possível reproduzir a ordem de armazenamento de até 950 votos com exatidão e eficiência, sem probabilidades de erro ou alto custo computacional. Posteriormente, foi obtida a informação de que o LOG também público de eventos registra o instante de tempo em que cada eleitor confirmou seu voto [8]. Quando esse registro temporal é associado à lista de votos recuperados em ordem, fica também possível recuperar um voto específico inserido em um certo instante de tempo.

Agora suponha um atacante capaz de coagir eleitores e monitorar seu comportamento no dia de eleição. A recuperação dos votos em ordem permite que esse atacante tenha sucesso com certeza *matemática* em um conjunto de fraudes eleitorais, aqui denominadas por *voto de cabresto digital*, que consistem em inserir ou monitorar a posição dos eleitores coagidos na fila de votação e posterior recuperação dos votos correspondentes. Um horário de votação específico também determina a posição na ordem de votação que um certo eleitor confirmou seu voto. Examinando a posição correspondente na lista de votos recuperada em ordem do RDV revela diretamente quais foram as escolhas do eleitor. Este ataque de *quebra de sigilo direcionado* pode, além de violar o critério de voto secreto assegurado pela Constituição [9], causar constrangimento significativo para personalidades públicas (políticos, empresários, industriais, ministros). Note que o local e horário de votação destas personalidades é frequentemente noticiado pela imprensa no dia de eleição [10, 11].

Recomendação. *Eliminar o RDV e substitui-lo por um mecanismo que forneça a possibilidade real de verificação independente de resultados, como o voto impresso verificável pelo eleitor.*

Fonte inadequada de entropia

A coleta de informação imprevisível (entropia) tem caráter crítico para várias operações criptográficas que requerem dados aleatórios, como a geração de chaves criptográficas ou a alimentação de geradores pseudo-aleatórios, e em muitos casos é possível contornar completamente a técnica utilizada com ataques apenas na fonte de entropia. O *software* da urna eletrônica brasileira utilizava apenas a medida do tempo em resolução de segundos como fonte de entropia, mesmo tendo disponíveis fontes de melhor qualidade em *hardware*. Esta não é uma vulnerabilidade desconhecida em sistemas de votação ou *software* comercial. A máquina de votar utilizada nos Estados Unidos empregava técnicas igualmente inseguras [3]. Em 1995, calouros de doutorado da Universidade de Berkeley descobriram, sem acesso ao código-fonte, que a versão 1.1 do navegador Netscape apresentava exatamente a mesma vulnerabilidade [12] encontrada no *software* de votação da urna eletrônica.

Recomendação. *Para satisfazer o critério de aleatoriedade verdadeira, recomenda-se utilizar um gerador em* hardware *baseado em efeito físico bem estudado. Segundo especificação da urna eletrônica modelo 2009 [13], dois geradores com estas características já estão disponíveis no equipamento [14].*

Verificação insuficiente de integridade

A urna eletrônica conta com um mecanismo de verificação de integridade de *software* que tem como objetivo verificar se houve adulteração dos programas entre sua produção e sua execução propriamente dita no equipamento, mas toda a informação necessária para subverter esse mecanismo encontra-se armazenada nas próprias urnas eletrônicas, com dificuldades distintas para um ataque, dependendo da presença de um módulo customizado de segurança em *hardware*. Em urnas sem este recurso, o problema de verificação é reduzido a si próprio, sem fonte externa de confiança. Nesse caso, "*software* auto-verificável" [15] por assinatura digital equivale a confiar a autenticidade de uma assinatura de punho em um documento apenas ao testemunho do próprio "autor", que, assim, pode se passar por quem quiser. Em urnas com este recurso, o mecanismo pode também ser contornado, mas apenas com colaboração de um agente interno.

É importante ressaltar ainda que uma assinatura digital autêntica apenas atesta o processamento do conteúdo assinado em algum ponto no tempo e espaço no qual também estava presente a chave de assinatura. Mesmo que os mecanismos de verificação de integridade não sejam contornados, ainda não há qualquer garantia de que o conteúdo é de fato o desejado. Caso o *software* possua vulnerabilidades, a verificação de integridade tem o efeito colateral de garantir que as mesmas vulnerabilidades estarão presentes em todas as urnas. A versão do código observada pelos autores apresentava ainda como desativada a verificação de integridade de parte do *software* contido na urna, evidenciando as limitações intrínsecas da técnica. O Relatório da SBC já apresentava ceticismo explícito a respeito da possibilidade de auto-verificação de *software* através de técnicas criptográficas [3]. À esta preocupação, soma-se a observação de que garantir que o *software* sendo executado na urna eletrônica é exatamente o mesmo produzido pelo TSE não torna o *software* seguro, apenas confirma sua origem.

O problema de verificação de integridade de *software* é endêmico em sistemas de votação eletrônica. Este é um problema particularmente difícil de se resolver na prática. A mesma limitação nos controles de integridade também foi observada no *software* do equipamento utilizado nos Estados Unidos [4]. É por essa razão que se recomenda a instalação de mecanismos que forneçam capacidade de verificação independente de *software* dos resultados, para que os resultados da eleição não dependam unicamente da integridade do *software* [1].

Recomendação. *Transferir a pressão da verificação de integridade do* software *para a verificação independente dos resultados produzidos pelo* software.

Compartilhamento de chaves criptográficas

Todas as urnas eletrônicas em operação no país utilizam a mesma chave criptográfica para cifrar as partições protegidas nos cartões de memória. Utilizando a analogia clássica de um cadeado como abstração de técnica criptográfica, isto é equivalente a proteger meio milhão de cadeados com uma mesma chave, visto ser este o número aproximado de equipamentos em operação. O vazamento dessa chave criptográfica tem impacto devastador e revela ao atacante o conteúdo completo dos cartões de memória, incluindo aí o *software* de votação, os mecanismos de verificação de integridade implementados em *software* e a chave de assinatura dos produtos públicos de votação [16]. Esta última chave é compartilhada ainda por todas as urnas eletrônicas da mesma unidade federativa [17] e seu vazamento permite a uma atacante produzir um arquivo forjado (LOG, RDV ou BU) mas verificado como autêntico, em nome de uma urna escolhida arbitrariamente.

Observa-se que o módulo de segurança em *hardware* introduzido nas urnas eletrônicas possui capacidade ociosa para armazenamento seguro de chaves criptográficas [13]. Ou seja, o sigilo da chave privada e, consequentemente, a integridade dos boletins de urna com a totalização parcial dos votos, reside apenas na confidencialidade de um segredo compartilhado por meio milhão de equipamentos.

Recomendação. *Atribuir uma chave criptográfica distinta para cada equipamento, ou pelo menos, para cada cartão de memória utilizado para inseminar um conjunto reduzido de urnas eletrônicas.*

Presença de chaves no código-fonte

O compartilhamento da chave de cifração das mídias é agravado pela sua presença às claras no código-fonte do *software*. Utilizando a mesma analogia do cadeado, isto equivale a esconder a chave embaixo do tapete e confiar no segredo dessa localização como fonte de segurança. Ou seja, qualquer agente interno que possua acesso ao repositório onde é mantido o código-fonte também possui automaticamente acesso à chave criptográfica que protege as partições cifradas dos cartões de memória, podendo realizar o vazamento de impacto devastador mencionado anteriormente. Isto também significa que a chave de cifração precisa estar armazenada às claras dentro do próprio cartão de memória, qualificando este mecanismo como apenas de ofuscação ao invés de verdadeira segurança. Basta que um atacante conheça a posição em que é armazenada a chave de cifração, por análise da porção do *software* armazenada às claras nos cartões de memória, para que o vazamento da chave se torne possível até para agentes externos.

Recomendação. *Armazenar a chave de cifração no módulo de segurança em* hardware *ou, preferivelmente, em dispositivo criptográfico seguro externo ao ambiente da urna eletrônica.*

Escolha inadequada de algoritmos

Além da escolha absolutamente inadequada do algoritmo para geração de números pseudo-aleatórios, o *software* da urna eletrônica também utiliza uma *função de resumo criptográfico* para fins de assinatura digital e verificação de integridade com uso não recomendado desde 2006, quando se verificou que a mesma não fornecia a segurança esperada, ficando recomendada como prudente a migração *rápida* para funções mais seguras [19].

Recomendação. *Utilizar um gerador de números pseudo-aleatórios de qualidade criptográfica, como comentado anteriormente, e uma função de resumo criptográfico padronizada e segura.*

3.2 No processo de desenvolvimento

As fragilidades discutidas anteriormente são produto de um processo de desenvolvimento de *software* também frágil. Discutimos a seguir as fragilidades encontradas ou inferidas pelo contexto nesse processo de desenvolvimento.

Complexidade acentuada

Segurança advém de simplicidade, transparência e correta avaliação de premissas e condições de confiança. O volume de milhões de linhas de código-fonte empregado para se realizar eleições no Brasil elimina qualquer possibilidade de auditoria completa ou eficaz do *software*. Um volume de código dessa magnitude irá possuir, *inevitavelmente*, vulnerabilidades que podem ser exploradas. Por essa razão, a base de código deve ser completamente orientada em torno de um pequeno conjunto crítico de funcionalidades, das quais depende o funcionamento correto e seguro do equipamento. Como um valor de referência, os pesquisadores que realizaram a avaliação das máquinas de votar dos Estados Unidos em um intervalo de 60 dias concluíram que os milhares de código dedicados às camadas de aplicação daquele equipamento são de complexidade tal que não é possível tornar o *software* seguro [4].

Recomendação. *Reduzir o volume de código a partir de técnicas de engenharia de* software, e*vitar intervenções no código-fonte externo ao TSE e isolar as porções de código de sistema operacional e aplicação para facilitar a auditoria interna do* software.

Auditoria externa insuficiente

Os partidos possuem a prerrogativa legal de examinar o código-fonte do *software* da urna eletrônica, mas para isso precisam assinar um Acordo de Não-Divulgação (AND) que os impede de detalhar publicamente qualquer problema observado no código, mediante imposição legal. Desta forma, os fiscais de partidos são impedidos de prestar contas à sociedade sobre a qualidade do que é feito no *software*, enquanto agentes desonestos possuem toda a liberdade para tentar articular fraudes eleitorais, sem qualquer risco de vazamento dos detalhes das vulnerabilidades encontradas. Como a fiscalização por investigadores independentes é extremamente limitada, na prática nenhuma fiscalização efetiva é realizada sobre o *software* do sistema eletrônico de votação. Como afirma de forma contundente o Relatório SBC [3]:

"A segurança e corretude dos programas usados na urna baseia-se em confiar na boa fé dos técnicos do TSE. Repetimos: não há nenhuma razão para duvidar da boa fé destas pessoas. Mas isto fere as boas práticas de segurança."

Como a integridade dos resultados depende unicamente da integridade desse *software*, fica montado um cenário perfeito para fraudes que não deixam vestígios.

Recomendação. *Permitir a auditoria do código-fonte por qualquer cidadão brasileiro, especialista ou não, sem qualquer obstáculo legal.*

Formulação equivocada de modelo de atacante

O projeto de mecanismos de segurança utilizado preocupa-se exageradamente com atacantes externos e ignora o risco de atacantes internos. Em particular, como demonstra a própria posição oficial do TSE [18], a detecção de comportamento malicioso por agentes internos é reduzida a processos de auditoria também executados por humanos, obviamente internos. A questão da chave compartilhada de cifração é um exemplo perfeito deste fenômeno, visto que o armazenamento às claras desta mesma chave de cifração dentro da própria urna eletrônica evidencia que os mecanismos de segurança não são projetados para resistir a atacantes que dispõem de informação privilegiada.

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

Ausência de exercícios internos

Em reunião após a audiência pública para prestação de contas, realizada entre a equipe e vários membros dos setores responsáveis pelas fases de projeto, produção e logística da urna eletrônica, os autores ofereceram a possibilidade de

ministrar uma palestra técnica para detalhar todos os problemas de segurança encontrados no *software* e o raciocínio específico que os levou à detecção e exploração da vulnerabilidade no embaralhamento dos votos. A proposta foi bem recebida, por permitir aos interessados o entendimento exato de "como funciona a mente do atacante", nas palavras dos próprios membros do TSE. Não houve convite concreto posterior para tal, mas a leitura dos autores a partir dessa afirmação é de que não existe um time interno responsável por simular o atacante, exercitar metodologias de ataque e tentar derrotar os mecanismos de segurança.

Recomendação. *Instituir, treinar e orientar um time interno de atacantes simulados, prática recomendada para* software *de missão crítica [4]. Não faz sentido projetar mecanismos de segurança sem que existam tentativas simultâneas de subvertê-los.*

Falta de treinamento formal

As fragilidades discutidas aqui, presentes inclusive em mecanismos críticos de segurança, demonstram claramente que os membros da equipe de desenvolvimento de *software* do TSE não recebem treinamento suficiente para implementar *software* de segurança. A ausência de simulações internas que modelem satisfatoriamente atacantes plausíveis, por falta de entendimento sobre o modo de atuação de um atacante, também corrobora essa observação, visto que um profissional com treinamento adequado na área de segurança já naturalmente costuma se alternar entre os papéis de projetista e atacante por todo o tempo.

Recomendação. *Instituir uma política para treinamento especializado da equipe de desenvolvimento é fundamental para se incrementar a qualidade geral do* software. *Não é plausível esperar* software *seguro como resultado do trabalho de uma equipe de desenvolvimento sem treinamento.*

Disponibilização de dados críticos aos investigadores

As máquinas dedicadas por exibir o código-fonte na sala lacrada durante os Testes Públicos de Segurança pareciam ter vindo diretamente da equipe de desenvolvimento. A razão para tal é a disponibilização para todos os investigadores de informações críticas a respeito de nomes de usuário, senhas e o caminho na rede interna para servidores de versionamento do código da urna. Um atacante externo que consiga invadir a rede interna do TSE e esteja munido dessas informações consegue ainda realizar alterações maliciosas no código-fonte e efetivá-las sob a alcunha de um membro legítimo da equipe de desenvolvimento, transferindo completamente para um inocente a responsabilidade por seus atos.

Recomendação. *Sanitizar equipamentos disponibilizados para visitantes externos, para que os mesmos não revelem informações críticas.*

Ignorância da literatura relevante

A vulnerabilidade encontrada no embaralhamento dos votos é conhecida desde 1995 [12]. Além disso, várias fragilidades apresentadas nesse relatório já foram descritas em laudos técnicos de outros sistemas de votação [4], e mesmo em do próprio [3], os quais contrariam o bom senso e a especificação formal das técnicas utilizadas. A persistência desses problemas em uma base de código com 16 anos de história é injustificável e evidencia claramente que a equipe de desenvolvimento do TSE não acompanha de forma adequada os movimentos relevantes nas áreas de votação eletrônica e segurança computacional.

Recomendação. *Responsabilizar parte da equipe de desenvolvimento por estudar e disseminar avanços relevantes de caráter acadêmico ou prático para a segurança de sistemas.*

Falsa sensação de segurança

A repetição incessante de que a urna eletrônica brasileira é absolutamente segura e inviolável, mesmo que isso constitua até uma impossibilidade teórica, perturba o senso crítico dos membros da equipe de desenvolvimento, que terminam por suspender seus próprios mecanismos de auto-avaliação. O processo de desenvolvimento do *software* da urna eletrônica parece funcionar sob o efeito de *suspensão de descrença*, instalando uma falsa sensação de segurança generalizada. Este não é o ambiente ideal para se desenvolver soluções de segurança, especialmente quando as mesmas precisam satisfazer o requisito de missão crítica.

Recomendação. *Adequar o processo de desenvolvimento de* software *para que estimule a verificação mútua e crítica do trabalho realizado, com parâmetros realistas de avaliação.*

CONCLUSÕES E PERSPECTIVAS

Este artigo apresentou um conjunto de vulnerabilidades no *software* da urna eletrônica que permitiu a recuperação eficiente, exata e sem deixar vestígios dos votos em ordem registrados eletronicamente, derrotando o único mecanismo de proteção do sigilo do voto utilizado pelo *software* de votação. A necessidade de se instalar recursos para avaliação científica, independente e contínua do *software* torna-se evidente, havendo ampla disponibilidade de especialistas na academia e

indústria capazes de contribuir na direção do incremento real das propriedades de segurança na solução adotada para votação eletrônica no país.

Esse conjunto de fragilidades e vulnerabilidades termina apenas por fornecer evidências materiais para as preocupações já levantadas pelo Relatório SBC de 2002 [3]. Em particular, pode-se concluir que não houve incremento significativo nas propriedades de segurança fornecidas pelo *software* da urna eletrônica nos últimos 10 anos. Continuam preocupantes a proteção inadequada do sigilo do voto, a impossibilidade prática de auditoria completa ou minimamente eficaz do *software*, e a verificação insuficiente ou inócua de integridade do *software* de votação. Como estas três propriedades são atualmente críticas para garantir o anonimato e destinação correta dos votos computados, resta aos autores repetir as conclusões do Relatório SBC e defender a reintrodução do voto impresso nos termos apresentados em [3] como mecanismo simples de verificação de integridade dos resultados de eleições.

O voto impresso distribui a auditoria do *software* entre todos os eleitores, que se tornam responsáveis por conferir que seus votos foram registrados corretamente pela urna eletrônica, desde que apuração posterior seja realizada para verificar que a contagem dos votos impressos corresponde exatamente à totalização eletrônica parcial. Essa apuração pode ser realizada por amostragem, de forma a não haver impacto significativo na latência para divulgação dos resultados. Vale ressaltar que o voto impresso é para fins de conferência apenas no interior da seção eleitoral, e não pode servir de comprovante no ambiente externo à seção eleitoral, como determinava a legislação a respeito [20]. A proposta de voto impresso retornaria para o sistema brasileiro de votação nas eleições de 2014, mas infelizmente foi declarada inconstitucional sob alegações tecnicamente questionáveis.

Um movimento nesta direção acompanharia a tendência mundial vigente em sistemas de votação eletrônica. Com a adoção do voto impresso pela Índia, o Brasil permanece como o único país no mundo a adotar sistema de votação sem verificação independente de resultados. Acreditamos que por esse motivo, e dadas as fragilidades discutidas neste relatório, o *software* utilizado no sistema de votação eletrônica brasileiro não satisfaz requisitos mínimos e plausíveis de segurança e transparência.

AGRADECIMENTOS

Os autores gostariam de agradecer aos professores Pedro Rezende, Jeroen van de Graaf, Paulo Barreto, Francisco Rodríguez-Henríquez e Alex Halderman por discussões relevantes e comentários abrangentes em versões preliminares deste documento.

Diego Aranha · Bacharel em Ciência da Computação pela UnB (2005), Mestre (2007) e Doutor (2011) em Ciência da Computação pela Unicamp. Foi doutorando visitante por 1 ano na Universidade de Waterloo, Canadá, e Professor Adjunto por pouco mais de 2 anos no Departamento de Ciência da Computação da UnB. Hoje atua como Professor Doutor no Instituto de Computação da Unicamp. Tem experiência na área de Criptografia e Segurança Computacional, com ênfase em implementação eficiente de algoritmos criptográficos e projeto de primitivas criptográficas para fornecimento de anonimato computacional. Coordenou a primeira equipe de investigadores independentes capaz de detectar e explorar vulnerabilidades no software da urna eletrônica em testes controlados organizados pelo Tribunal Superior Eleitoral.

Marcelo Monte Karam · Graduado em 2003 como 3º Sargento de Comunicações do Exército Brasileiro, posteriormente transferido para a 1 Companhia de Guerra Eletrônica em Brasília, onde participou de atividades e cursos de guerra eletrônica e teve o primeiro contato com a segurança eletrônica e de sistemas. Formado em 2008 em Tecnologia da Segurança da Informação, tornou-se Analista de TI da UnB. Em 2009 foi convidado para coordenar a equipe de segurança de redes da universidade. Atualmente é pesquisador em atividade na área de respostas a incidentes computacionais e cursa mestrado profissionalizante em Computação Aplicada.

André de Miranda é Analista de Segurança da informação. Trabalhou em grandes projetos de segurança em Brasília em vários órgãos do governo, como Comando da Aeronáutica, Policia Federal, Caixa Econômica Federal, Universidade de Brasília, Presidência da República e Exército Brasileiro. Também atua como instrutor de cursos de segurança ofensiva e defensiva, Linux e redes de computadores.

Felipe Brant Scarel · Graduado em 2009 como Bacharel em Ciência da Computação pela Universidade de Brasília. Focado principalmente nas áreas de sistemas e redes de computadores, trabalhou em projetos ligados à UnB, Caixa Econômica Federal, Instituto de Tecnologia da Informação do Governo Federal, dentre outros. Atualmente trabalha como Analista de Segurança da Informação no Sistema de Cooperativas de Crédito do Brasil (SICOOB).

REFERÊNCIAS

[1] RIVEST, R. L.; WACK, J. P. On the notion of "software independence" in voting systems. Philosophical Transactions of The Royal Society A 366 (1881), 2008. Disponível em http://people.csail.mit.edu/ rivest/pubs.html#Riv08b

[2] REZENDE, P. Reforma Eleitoral e Informatização do Voto, 2012. Disponível em http://www.cic.unb.br/ docentes/pedro/trabs/Voto-eLima2012.html

[3] VAN DE GRAAF, J.; CUSTÓDIO, R. F.: Tecnologia Eleitoral e a Urna Eletrônica -- Relatório SBC 2002. Disponível em http://www.sbc.org.br/index.php?option=com_jdownloads&Itemid=195&task=view. download&catid=77&cid=107

[4] CALANDRINO, J. A.; FELDMAN, A. J.; HALDERMAN, J. A.; WAGNER, D.; Yu, H.; ZELLER, W. P.: Source Code Review of the Diebold Voting System, 2007. Disponível em https://jhalderm.com/pub/papers/diebold-ttbr07.pdf.

[5] PRESIDÊNCIA DA REPÚBLICA. Lei Nº 9.504, de 30 de Setembro de 1997. Disponível em http://www.planalto.gov.br/ccivil_03/leis/l9504.htm

[6] PRESIDÊNCIA DA REPÚBLICA. Lei Nº 10.408, de 10 de Janeiro de 2002. Disponível em http://www.planalto.gov.br/ccivil_03/leis/2002/L10408.htm

[7] PRESIDÊNCIA DA REPÚBLICA. Lei Nº 10.740, de 1º de Outubro de 2003. Disponível em http://www.planalto.gov.br/ccivil_03/leis/2003/l10.740.htm

[8] TRIBUNAL SUPERIOR ELEITORAL. Especificação do Arquivo e Registro de Log das Urnas Eletrônicas para as Eleições 2008, Versão 2. Disponível em http://www.tse.gov.br/internet/eleicoes/arquivos/ logs2008/EspecificacaoArquivoRegistroLogUrnasEletronicasEleicoes2008.pdf

[9] PRESIDÊNCIA DA REPÚBLICA. Lei N**o** 4.737, de 15 de Julho de 1965. Disponível em http://www.planalto.gov.br/ccivil_03/leis/l4737.htm

[10] AGÊNCIA DE NOTÍCIAS DA JUSTIÇA ELEITORAL. Presidente do TSE vota em trânsito na capital federal. Disponível em http://agencia.tse.jus.br/sadAdmAgencia/noticiaSearch.do?acao=get&id=1336461

[11] CORREIO BRAZILIENSE. Presidente do TSE, Ricardo Lewandowski, vota em trânsito no IESB. Disponível em http://www.correiobraziliense.com.br/app/noticia/especiais/eleicoes2010/2010/10/03/interna_eleicoes2010,216159/index.shtml

[12] GOLDBERG, I.; WAGNER, D.: Randomness and the Netscape Browser. Dr. Dobb's Journal, 1996.

[13] TRIBUNAL SUPERIOR ELEITORAL. Aquisição de Urnas Eletrônicas – UE2009 / PB – Projeto Básico. http://www.tse.jus.br/transparencia/arquivos/tse-projeto-basico-audiencia-publica-2009

[14] AMD. Design without compromise, 2007. Disponível em http://www.amd.com/us/Documents/ 33358e_lx_900_productb.pdf

[15] JANINO, G. D.; BALCÃO FILHO, A.; MONTES FILHO, A.; LIMA-MARQUES, M; DAHAB, R. Relatório do Comitê Multidisciplinar nomeado pela Portaria-TSE 192, 2009.

[16] TRIBUNAL SUPERIOR ELEITORAL. Eleições 2010 – Listagem de Hashs. Disponível em http://www.tse.jus.br/arquivos/tse-urna-eletronica-modelo-2009-eleicoes-2010-turno-1--e-2-atualizado-em-22-09-2010-991ue09

[17] TRIBUNAL SUPERIOR ELEITORAL. Sistema OKEY – 1º Turno, 2010. Disponível em http://www.tse.jus.br/ arquivos/tse-chaves-das-u.f.s-eleicoes-2010-turno-1-e-2-9910key

[18] Coluna Segurança Digital, por ROHR Altieres. Falha na urna brasileira ``reproduzia fielmente" erro de 1995, diz professor. http://g1.globo.com/platb/seguranca-digital/2012/05/28/falha-na-urna-brasileira-reproduzia-fielmente-erro-de-1995-diz-professor/

[19] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. NIST comments on Cryptanalytic Attacks on SHA-1, 2006. http://csrc.nist.gov/groups/ST/hash/statement.html

[20] PRESIDÊNCIA DA REPÚBLICA. Lei N**o** 12.034, de 29 de Setembro de 2009. Disponível em http://www.planalto.gov.br/ccivil_03/_ato2007-2010/2009/lei/l12034.htm