

(IN)SEGURANÇA DO VOTO ELETRÔNICO NO BRASIL

Diego F. Aranha, Pedro Barbosa, Thiago Cardoso, Caio Lüders, Paulo Matias

Unicamp, UFCG, Hekima, UFPE, UFSCar

Propriedades de segurança

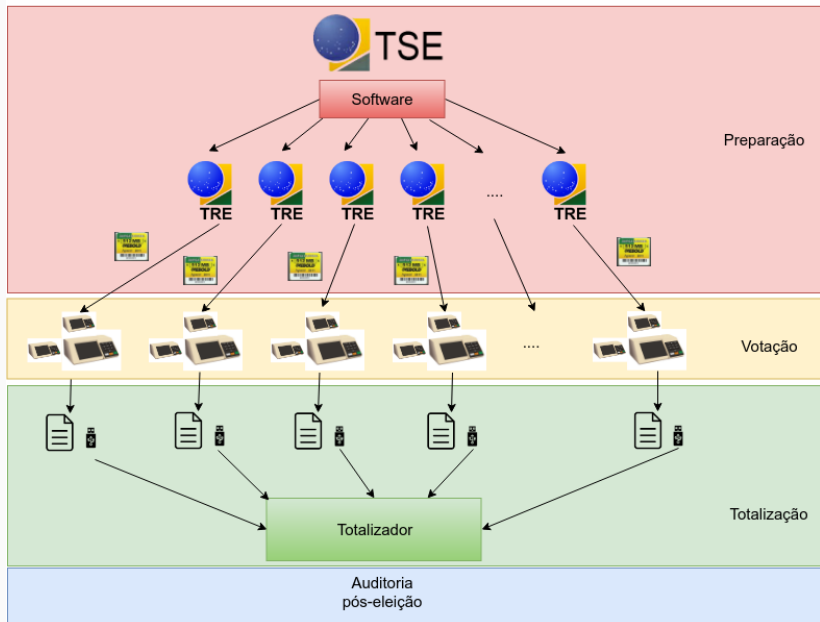
Não importando a tecnologia empregada, um sistema de votação precisa satisfazer algumas propriedades:

1. *Autenticação dos eleitores*: apenas eleitores autorizados podem votar
2. *Sigilo do voto*: voto deve ser secreto
3. *Integridade dos resultados*: resultado é justo
4. *Possibilidade de auditoria*: idealmente, sem especialização

Importante: em um sistema puramente eletrônico de votação, **todas as propriedades** são responsabilidade da tecnologia.

- 1996 : Urnas eletrônicas em 30% das seções eleitorais
- 2000 : Primeiras eleições inteiramente eletrônicas
- 2002 : Primeira experiência com voto impresso
- 2006 : TSE passa a ser responsável pelo *software*
- 2008 : Migração para GNU/Linux
- 2009 : I Testes Públicos de Segurança (quebra de sigilo do voto)
- 2012 : II TPS (quebra de sigilo do voto)
- 2016 : III TPS (quebra na integridade de resultados)
- 2017 : IV TPS (quebra na integridade de *software*)

Organização do sistema:
preparação, votação, totalização



Preparação

1. Confeção do *software* de votação no TSE
2. Transmissão do *software* de votação para TRES
3. Gravação do *software* de votação em cartões de memória *flash*
4. Distribuição dos cartões de memória
5. Instalação nas urnas eletrônicas (carga)



Transparência

- Exame por fiscais de partido, OAB, MPU, SBC
- Testes Públicos de Segurança

1. Impressão da zerésima
2. Sessão de votação (identificação biométrica, interação com urna)
3. Impressão do Boletim de Urna (BU)
4. Gravação digital na Mídia de Resultados (MR) de BU eletrônico, Registro Digital do Voto (RDV), etc.

Transparência

- Zerésima
- Votação paralela
- Registro Digital do Voto?

Totalização

1. Transmissão dos resultados parciais
2. Combinação dos resultados parciais
3. Divulgação do resultado final
4. Publicação dos BUs eletrônicos

Transparência

- Conferência entre BUs físico e eletrônico
- Totalização paralela

Limitações de transparência:
o que pode dar errado?

Limitações na preparação

Fiscalização

- Complexidade do *software* de votação ($> 10^6$ linhas)
- Falta de treinamento formal dos fiscais
- Filiação ou indicação partidária
- Termo de Confidencialidade

Testes Públicos de Segurança

- Formato burocrático (8 tipos de formulários)
- Escopo e duração dos testes
- Condições de trabalho poucos realistas, não modelam atacante hábil
- Conflito de interesse intrínseco
- Termo de Confidencialidade (em 2016)

Zerésima e Votação paralela

- Não previnem *software* desonesto
- Simulação × Realidade (caso da Volkswagen)
- Tamanho e qualidade da amostra

Limitações na votação

Zerésima e Votação paralela

- Não previnem *software* desonesto
- Simulação × Realidade (caso da Volkswagen)
- Tamanho e qualidade da amostra

Exemplo de comportamento nunca detectado na votação paralela:

```
If (voto == 99999) {  
    ativar_comportamento_malicioso();  
}
```

Importante: Assumir versão ofuscada escondida na base de código!

Limitações na totalização

Conferência

- Limitação na emissão de BUs impressos
- Restrições logísticas (custo e cobertura)
- *Projeto Você Fiscal* (2014 e 2016)
- Tamanho e qualidade da amostra

Importante: É provavelmente a fase mais transparente do processo eleitoral eletrônico, especialmente após introdução do código QR.



E uma auditoria pós-eleição?

- Primeira realizada em 2014, relatório inconclusivo (“*não permite a plena auditagem*”)
- Meses para entrega de todos os arquivos e documentos
- Conflito de interesse com o TSE e interesses partidários
- Influência da situação política

Imprensa e audiência do TSE

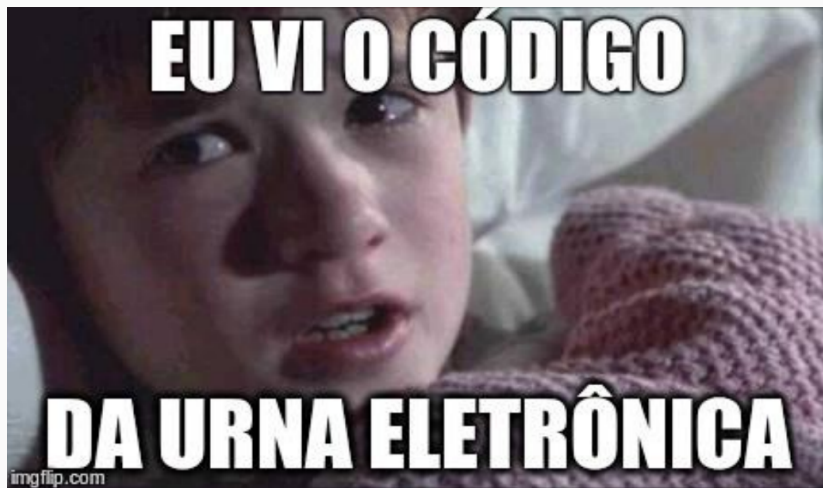
“Auditoria conclui que não houve fraude na eleição de 2014”

Conclusão: Sistema brasileiro eletrônico de votação **não é auditável.**

Mas o *software* é 100% seguro!
Somos líder em tecnologia
eleitoral... certo?



memecrunch.com



- Vulnerabilidade trivial no **sigilo do voto**
- Compartilhamento e armazenamento inseguro de **segredos** criptográficos
- Verificação **insuficiente** de integridade
- Processo de desenvolvimento **inseguro**
- Modelo adversarial **inadequado**
- Cultura interna **sem transparência**

Conclusão: falhas graves tecnológicas e procedimentais!

Governador Senador Presidente

71	31	37
	BRANCO	
13		
71	NULO	
		BRANCO
		37

Semente **secreta e aleatória** para embaralhar RDV:
`srand(time(NULL))`

Inst. Federal de Educação Ciência e Tecnologia do Rio Grande do Sul Campus Bento Gonçalves	
Zerésima	
Eleição do IFRS (28/06/2011)	
Município	88888
Bento Gonçalves	
Zona Eleitoral	0008
Seção Eleitoral	0021
Eleitores aptos	0083
Código identificação UE	01105161
Data	28/06/2011
Hora	08:32:08
RESUMO DA CORRESPONDENCIA	
588.653	

File 1/1: lew.jpg
File name: lew.jpg
File size: 47009 Bytes
MIME type: image/jpeg
Image size: 276 x 360
Camera make: Canon
Camera model: Canon EOS-1Ds Mark III
Image timestamp: 2010:10:03 11:20:37



Código de autenticação de BU para digitação manual:

-----PRESIDENTE-----		
Nome do candidato	Nro cand Votos	
DILMA	13	0124
AÉCIO NEVES	45	0037

Total de votos Nominais	0161	
Branco	0004	
Nulos	0021	
Total Apurado	0186	
Código Verificador: 94316		
=====		
Código de identificação da carga		
856.562.403.165.702.654.890.929		
Ver: 4.12.0.0 - Rio Sao Francisco		
ASSINATURAS:		

Montada pelo time *ELT*, que participa de competições nacionais e internacionais de *Capture The Flag* (CTF). Muitas das habilidades são equivalentes (curiosidade, ferramental, raciocínio adversarial):

- Pedro: *Assembly* e criptografia
- Thiago: *Web* e exploração
- Caio: *Tecnologias Web*
- Paulo: Engenharia Reversa e exploração
- Diego: ecossistema da urna, formato dos testes, criptografia

Diversidade: Todos os membros contribuíram com uma idéia fundamental em certo ponto que permitiu o avanço da equipe.

Edital de abertura especificava que investigadores não teriam acesso a chaves criptográficas.

Interpretação do TSE

Apagar as chaves criptográficas do código, “*para aumentar o desafio*”!

Edital de abertura especificava que investigadores não teriam acesso a chaves criptográficas.

Interpretação do TSE

Apagar as chaves criptográficas do código, “*para aumentar o desafio*”!

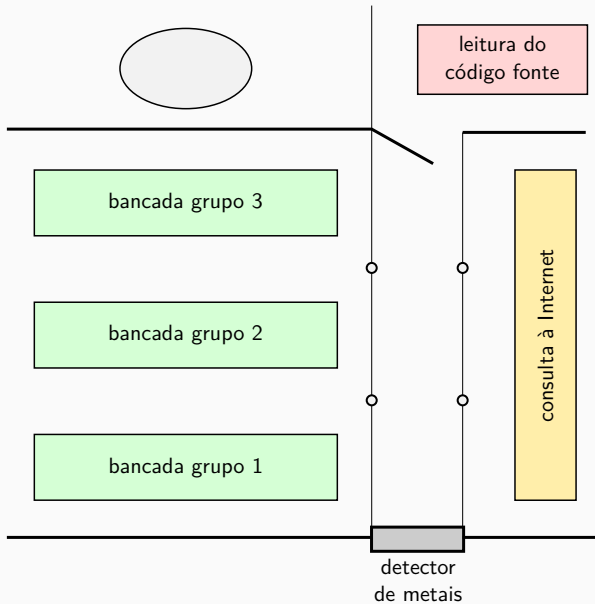
Falha operacional: Felizmente, não apagaram todas. Encontramos chave em claro na versão 3.18 do *kernel*. :)

Equipe fez uma análise inicialmente superficial do sistema, depois aprofundada nos **mecanismos criptográficos**, que concentram **risco**.

Submetemos vários planos de teste, todos **aprovados**:

- *Captura de chaves criptográficas do flash de carga*
- *Execução remota de código na plataforma web*
- *Tentativa de violação do sigilo do voto*
- *Inserção de dispositivo USB malicioso*

Pela restrição de tempo, nos concentramos no primeiro ataque e suas consequências.



Praticamente dedicado a **preencher formulários**, reconhecer o ambiente, solicitar computadores e começar a instalar Kali Linux nas máquinas.

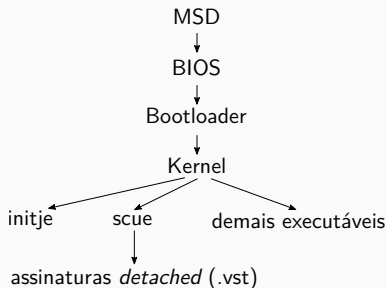
Os cartões de memória da urna eletrônica são cifrados com AES-256 em modo XTS, que exige duas chaves criptográficas, compartilhadas entre **todas as urnas**.

Progresso: Descobrimos durante a inspeção de código uma **presente no código-fonte** e a outra armazenada **às claras** no cartão de memória.

→ *Script* Python+OpenSSL em máquina de inspeção de código para decifrar um *stub* da partição cifrada que encontramos por lá.

→ Reimplementamos decifração de cartões de memória com `pycrypto` nas máquinas de teste, copiando a chave criptográfica **memorizada** alguns *bytes* por vez.

Após decifrar o cartão inteiro, estudamos a verificação de integridade:



Problema: Recebemos a visita de observadores internacionais, que atrapalharam bastante. :(

Progresso: Encontramos duas bibliotecas (`libapilog.so` e `libhkdf.so`) sem assinaturas digitais.

→ Injetamos trechos de código simples nessas bibliotecas para verificar se versões adulteradas eram devidamente instaladas na carga.

→ Alteramos todas as chamadas de uma das bibliotecas para imprimir **FRAUDE!** no terminal, o que aconteceu. :)

Exploramos ataques contra o sistema utilizando as funcionalidades fornecidas pelas bibliotecas:

- `libapilog.so`: conseguimos adulterar o registro de *log* substituindo `INFO` por `XXXX`.
- `libhkdf.so`: conseguimos adulterar a biblioteca para zerar chave criptográfica derivada para cifrar RDV.

→ Implementamos ainda um programa para receber e imprimir comandos de um teclado **acoplado à urna**. :)

Progresso: Cifrar o RDV com uma chave conhecida permite **violar sigilo de um voto específico**.

Após conseguir desempacotar o programa de votação **vota** com UPX, percebemos que estava ligado com as duas bibliotecas.

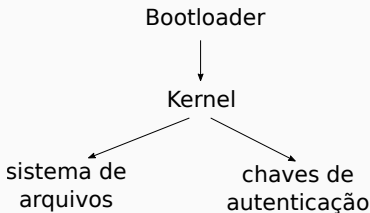
Progresso: Injetamos código para alterar a versão do *software* e o **conteúdo da tela** em tempo de execução:

→ Passamos as últimas horas trabalhando em interferir com a contagem de votos, e chegamos a produzir **erro de consistência por cédula vazia**.

Observação: Conseguimos controle sobre o software de votação para **injeção de código arbitrário**.

Problema: Tivemos que preparar uma demonstração dos resultados parciais, o que tomou mais tempo. :(

Progresso: Peritos da Polícia Federal **recuperam** a chave de cifração diretamente do *bootloader*, mostrando que acesso a código-fonte não é obrigatório.



Observação: acesso a uma única chave de encriptação fornece poder desproporcional a um atacante. **Sempre questionar premissas!**

SEU VOTO PARA

Presidente

Número:

Nome: Natação

Partido: PEsp

Presidente

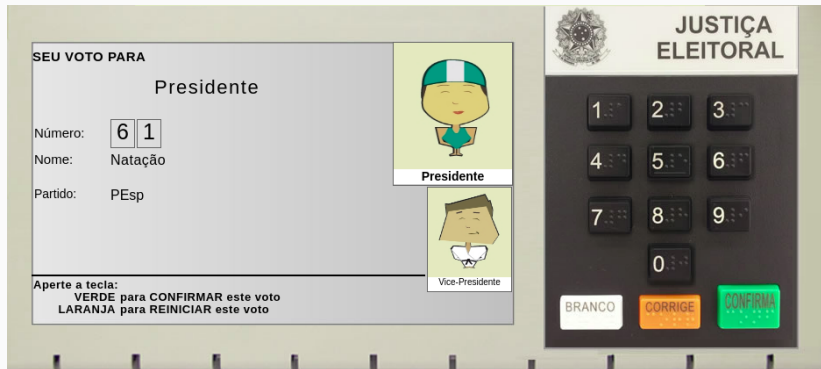
Vice-Presidente

Aperte a tecla:
VERDE para **CONFIRMAR** este voto
LARANJA para **REINICIAR** este voto

JUSTIÇA ELEITORAL

1 2 3
4 5 6
7 8 9
0

BRANCO CORRIGE CONFIRMA



VOTE 99


Presidente


Número:

Nome: Darth Vader

Partido: Dark Side

Aperte a tecla:
VERDE para CONFIRMAR este voto
LARANJA para REINICIAR este voto

 Presidente

 Vice-Presidente

JUSTIÇA ELEITORAL

1 2 3
4 5 6
7 8 9
0

BRANCO CORRIGE CONFIRMA

Solução independente de *software*

Figura 1: Máquina de votar utilizada no México com trilha de auditoria em papel (VVPAT)



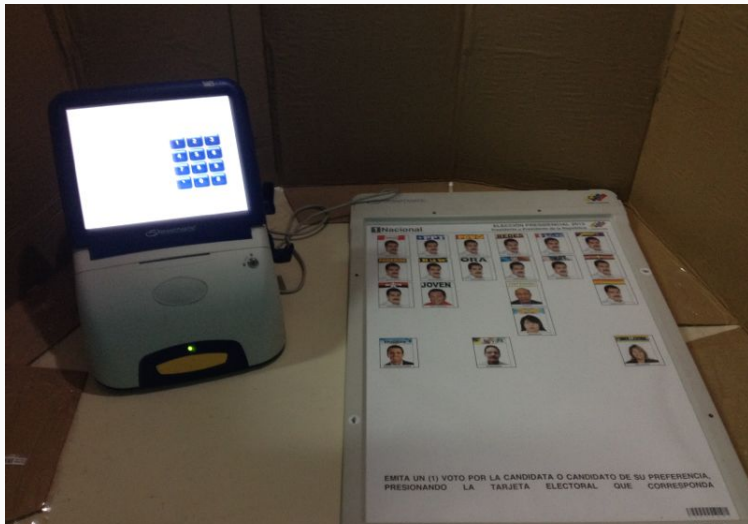
Solução independente de software

Figura 2: Máquina de votar utilizada na Índia com trilha de auditoria em papel (VVPAT)



Solução independente de software

Figura 3: Máquina de votar utilizada na Venezuela com trilha de auditoria em papel (VVPAT)

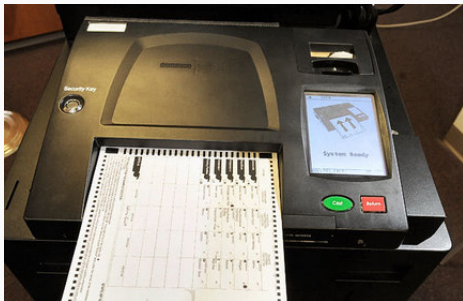


Solução independente de software

Figura 4: Sistema de votação utilizado na Argentina (E2E?).



Figura 5: Máquinas utilizadas nos Estados Unidos como scanners de votos.



Violadas ao menos duas barreiras de segurança:

- *Cifração das mídias* (erro de projeto)
- *Autenticação de bibliotecas compartilhadas* (falha procedimental e erro de projeto)

Crítico: Projeto depende fundamentalmente da ofuscação do *bootloader* para impedir ataques.

Observações adicionais:

- Tempo dos testes **não é comparável** a ataque real;
- Não é factível seguir **estritamente** o protocolo formal dos testes, usamos muitos atalhos.

... que persistem:

1. *Software* **secreto** por mais de 20 anos
2. *Software* **demonstravelmente** inseguro em múltiplas ocasiões
3. Ausência de **recontagem**
4. Ausência de auditoria **efetiva**
5. **Conflitos de interesse** em todo lugar
6. **Ataques internos** completamente ignorados

O que fazer?

1. Voto impresso

Implementar registro físico e anônimo do voto, conferível pelo eleitor, para auditoria/recontagem.

2. Código aberto

Publicar código-fonte do software é desejável para ampliar a capacidade de auditoria, mas insuficiente. Favor não esquecer de mover chaves criptográficas para lugar seguro. :)

3. Controle social

Ampliar mecanismos de transparência para que sociedade possa exercer maior controle social sobre o sistema, financiado por recursos públicos.

Perguntas?

D. F. Aranha

dfaranha@ic.unicamp.br

@dfaranha

1. Diego F. Aranha, Marcelo M. Karam, André de Miranda, Felipe Scarel. **Software vulnerabilities in the Brazilian voting machine.** In: *Design, Development, and Use of Secure Electronic Voting Systems*, 149-175, IGI Global, 2014.
2. Diego F. Aranha, Helder Ribeiro, André Luiz Ogando Paraense. **Crowdsourced integrity verification of election results: an experience from Brazilian elections.** *Annals of Telecommunications*, 71(7), 287-297, 2016.
3. Diego F. Aranha, Pedro Y. S. Barbosa, Thiago N. C. Cardoso, Caio Lüders de Araújo, Paulo Matias. **The Return of Software Vulnerabilities in the Brazilian Voting Machine**, Relatório Técnico, 2018.