# Hacking the Brazilian Voting System

## Hackers 2 Hackers Conference

Diego Aranha (Unicamp), Pedro Barbosa (UFCG),
Thiago Cardoso (Hekima), Caio Lüders (UFPE),
**Paulo Matias (UFSCar)**

20 de outubro de 2018

# 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

# Um breve histórico

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

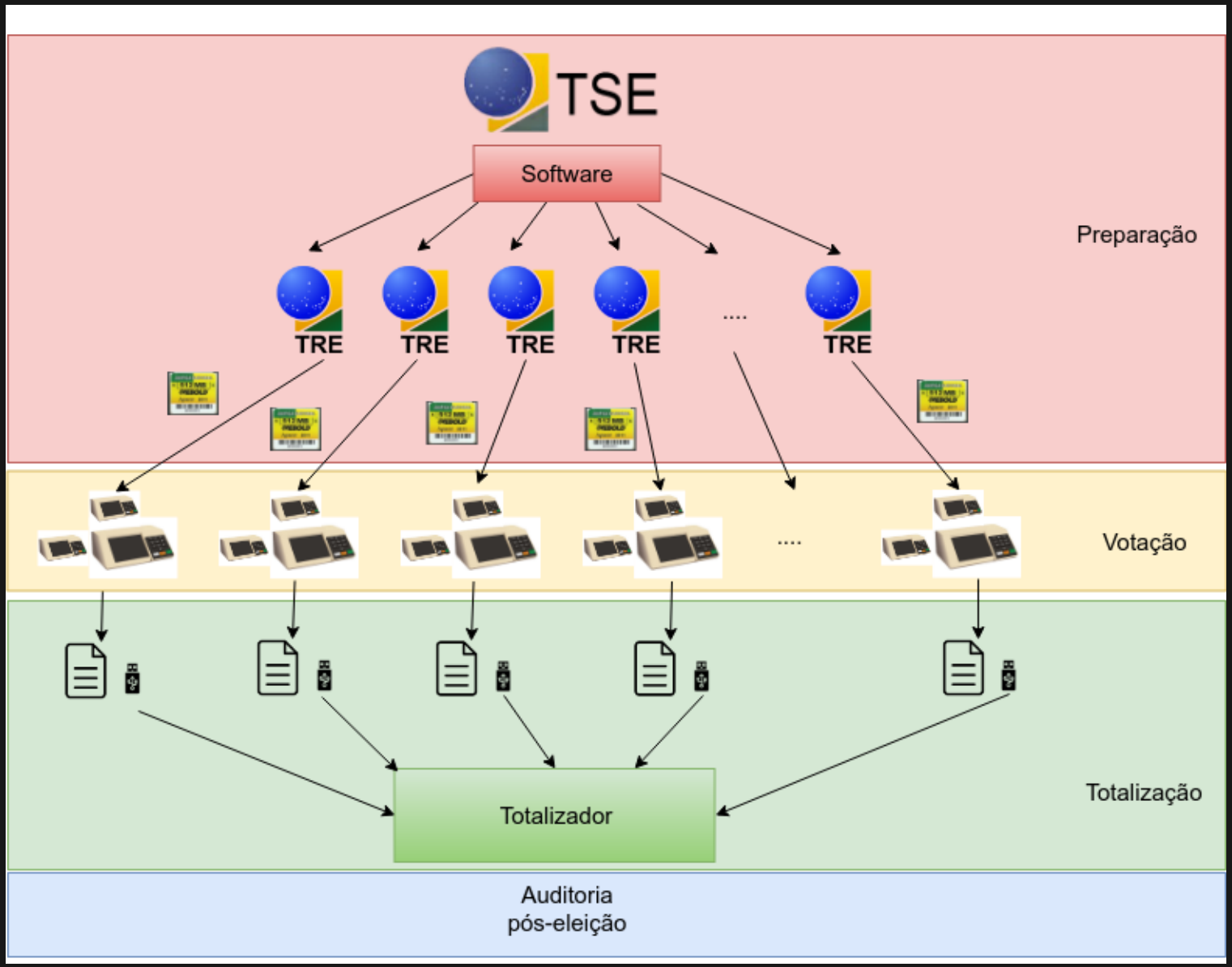
2009 : I Testes Públicos de Segurança (quebra de sigilo do voto)

2012 : II TPS (quebra de sigilo do voto)

2016 : III TPS (quebra na integridade de resultados)

2017 : IV TPS (quebra na integridade de *software*)

# Processo de votação brasileiro



# Preparação

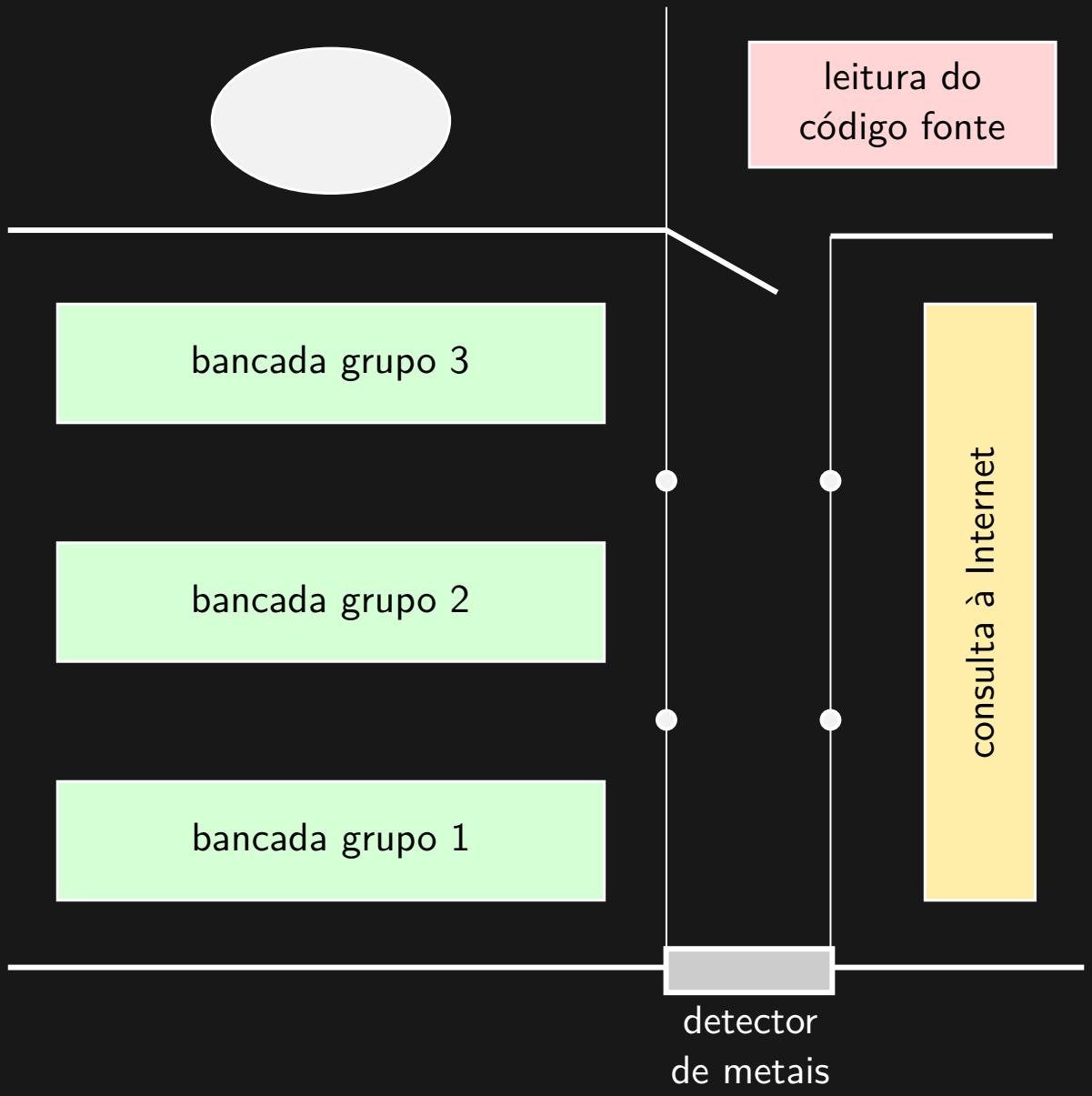
1. Confecção do *software* de votação no TSE
2. Transmissão do *software* de votação para TREs
3. Gravação do *software* de votação em cartões de memória *flash*
4. Distribuição dos cartões de memória
5. Instalação nas urnas eletrônicas (carga)

# Instalação (carga) nas urnas

# Como funciona o TPS?

- Fase de inspeção dos códigos fonte
- Submetemos **planos de teste**
- Os planos de teste são analisados e aprovados pelo TSE
- Executamos os planos de teste em uma bancada com computador e urna eletrônica

# Planta do ambiente

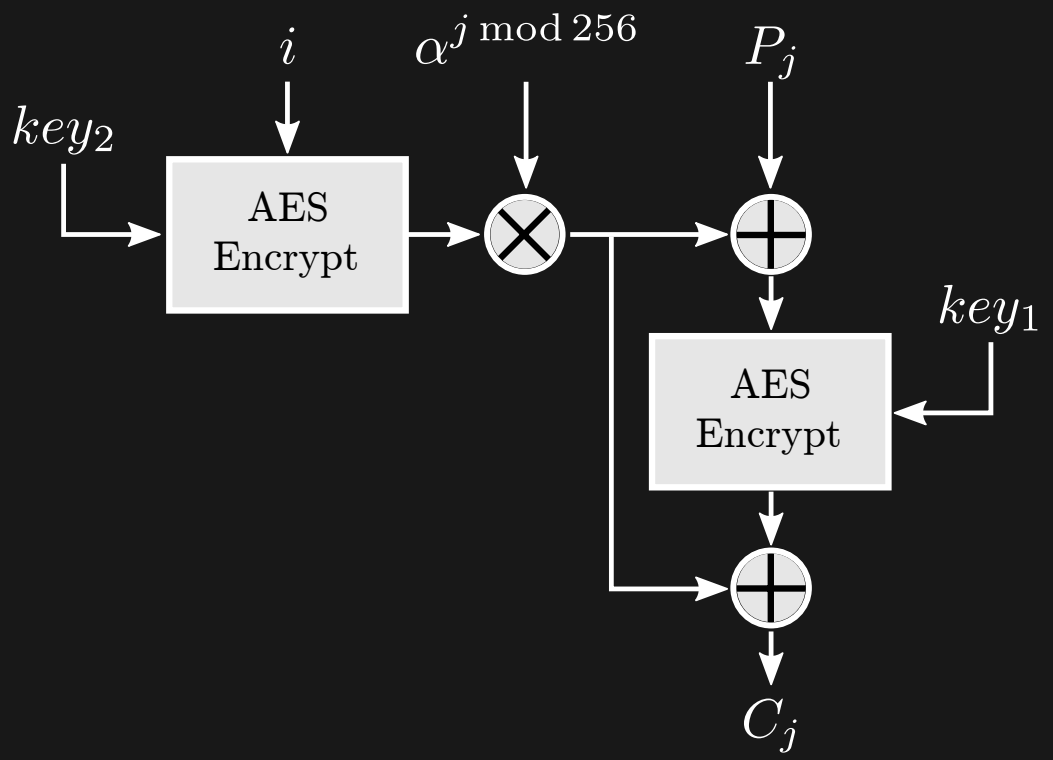


# Dificuldades

- Formato burocrático (8 tipos de formulários)
- Escopo e duração dos testes
- Entrada de software (em DVD-ROM) ou material impresso apenas após análise e aprovação de *solicitação de material*
- Regras aplicam-se mesmo para material discriminado nos planos de teste previamente aprovados
- Proibido transitar com anotações entre ambiente de leitura de código fonte e bancada de testes
- Problemas para habilitar virtualização nos computadores fornecidos
- Necessidade de realizar apresentações de resultados parciais

# Inspeção de código

- Encontramos chave da mídia de instalação em claro no código fonte do kernel 3.18



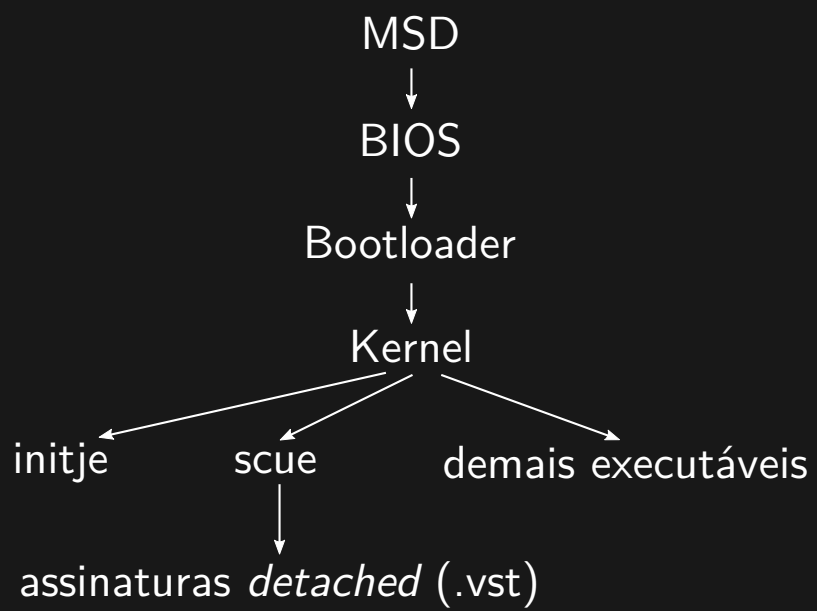
# Primeiro dia

- Preencher formulários, solicitar computadores, inspeção de código, configuração do ambiente...

- Fizemos script Python+OpenSSL em uma máquina de inspeção de código e conseguimos decifrar um stub da partição cifrada que encontramos por lá

# Segundo dia

- Reimplementamos o script de decifrar o cartão de memória com pycrypto nas máquinas de teste
- Estudamos a verificação de integridade do software:



# Terceiro dia

- Encontramos duas bibliotecas (libapilog.so e libhkdf.so) sem assinaturas digitais.
- Alteramos todas as funções de uma das bibliotecas para imprimir FRAUDE! no terminal, o que aconteceu :-)

# Terceiro dia

- Encontramos duas bibliotecas (libapilog.so e libhkdf.so) sem assinaturas digitais.
- Alteramos todas as funções de uma das bibliotecas para imprimir FRAUDE! no terminal, o que aconteceu :-)

- *Onde está o VOTA?*

# Quarto dia

- libapilog.so: adulteramos o registro de log, substituindo INFO por XXXX
- libhkdf.so: adulteramos a biblioteca para zerar a chave criptográfica derivada para cifrar o RDV e violar o sigilo de um voto específico
- Programa para interagir com um teclado USB conectado à urna

# Quarto dia

- libapilog.so: adulteramos o registro de log, substituindo INFO por XXXX
- libhkdf.so: adulteramos a biblioteca para zerar a chave criptográfica derivada para cifrar o RDV e violar o sigilo de um voto específico
- Programa para interagir com um teclado USB conectado à urna

- *Onde está o VOTA?*

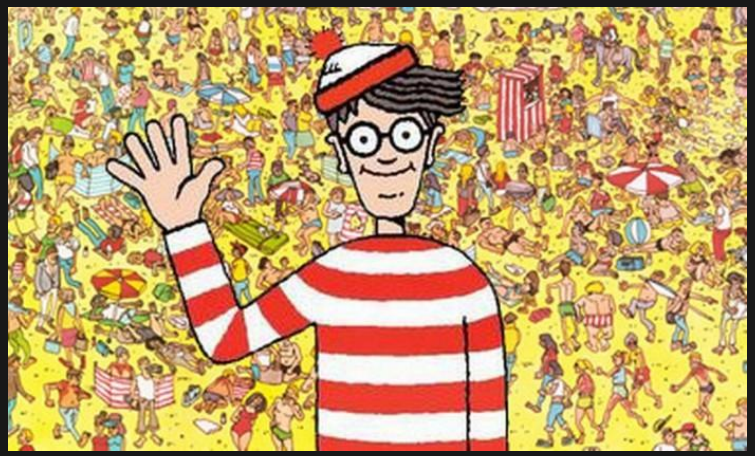
# Quinto e último dia

- Peritos da Polícia Federal inicializam carga da urna em uma máquina virtual e recuperam a chave da mídia de instalação
  $\Rightarrow$ basta acesso a um cartão para montar nosso ataque!

# Quinto e último dia

- Peritos da Polícia Federal inicializam carga da urna em uma máquina virtual e recuperam a chave da mídia de instalação
  $\Rightarrow$ basta acesso a um cartão para montar nosso ataque!

- *Achamos o VOTA!*

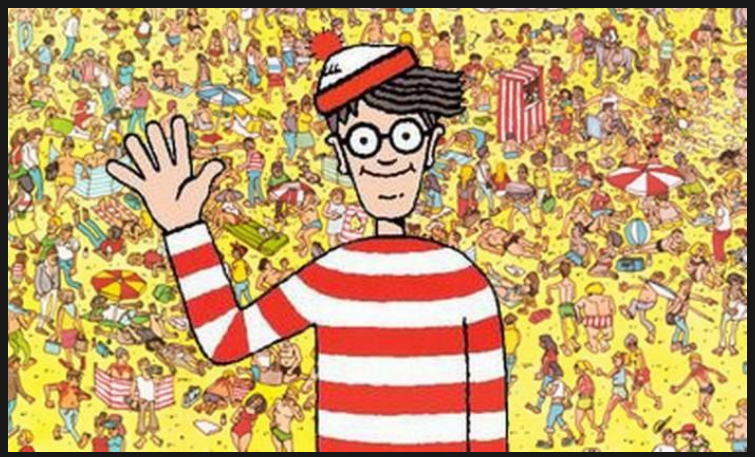


- Estava na terceira partição, e ninguém do time prestou atenção :-(

# Quinto e último dia

- Peritos da Polícia Federal inicializam carga da urna em uma máquina virtual e recuperam a chave da mídia de instalação
  $\Rightarrow$ basta acesso a um cartão para montar nosso ataque!

- *Achamos o VOTA!*



- Estava na terceira partição, e ninguém do time prestou atenção :-(

- Mas corre que dá tempo!

# Quinto e último dia

- Desempacotamos o VOTA (UPX)
- Percebemos que o VOTA estava linkado com as duas bibliotecas sem assinaturas
- **Execução arbitrária de código no espaço de memória do software de votação**
- Era suficiente? Não para os leigos...
  - Urna que faz boca de urna
  - Infecção do método `AdicionaVoto`

# Contramedidas e como contorná-las

- Usaram trecho da Flash da BIOS como chave para cifrar a mídia
  - Dumpar a flash da BIOS (uma única vez, de qualquer urna do país)
- Habilitaram PIE na compilação
  - Olhar a pilha e calcular endereços
- Incluíram todos os arquivos no .vst
  - Desabilitar verificação no SCUE
- Linkaram libapilog, libhkdf, *etc.* estaticamente
  - Infectar libc
- Kernel agora verifica assinatura de bibliotecas
  - Achar outro bug na cadeia de confiança
  - Achar bug na leitura de arquivos de dados do cartão
  - Dispositivo USB malicioso (não amplificável?)

# Decifração da mídia não é grave?!

- Chave privada dentro da mídia, compartilhada entre todas as urnas
- Problema reportado no relatório do TPS 2012
- Permitiria gerar produtos públicos falsos para qualquer seção
  - Mídia de Resultados *fake*
  - Boletim de Urna *fake*
- Cuidados na transmissão? Não amplificável?

"*Um sistema eleitoral é independente do software se uma modificação ou erro não-detectado no seu software não pode causar uma modificação ou erro indetectável no resultado da apuração*" – Ronald Rivest