Vulnerabilidades e sugestões de melhorias encontradas no Teste Público de Segurança 2017 sobre o Ecossistema da Urna

Relatório técnico

Brasília, 12 de dezembro de 2017

Tribunal Superior Eleitoral
Secretaria de Tecnologia da Informação
Coordenadoria de Sistemas Eleitorais
Seção de Voto Informatizado

Introdução

O Teste Público de Segurança - TPS, iniciado em 2009 e já com quatro edições, é um dos marcos do processo de desenvolvimento dos sistemas eleitorais e do hardware da urna eletrônica. Ao longo dos últimos anos, a cada edição do TPS foi possível aprimorar os sistemas eleitorais que seriam utilizados nas eleições subsequentes, que passaram a contar com hardware e software mais seguros e robustos.

A edição de 2017 do TPS contou com um grande número de pesquisadores e profissionais altamente qualificados. E não por acaso, o brilhantismo do seu trabalho contribuiu para a descoberta do maior número de achados de software para uma única edição do TPS.

Este relatório tem por objetivo apresentar os planos de teste que foram executados pelos investigadores durante o evento. É feita uma breve descrição dos trabalhos apresentados, dos resultados obtidos e das falhas que deram causa ao sucesso dos achados.

Sempre que possível, os achados são colocados no contexto real de exploração da vulnerabilidade apresentada. Este relatório não tem o objetivo de desqualificar ou minimizar o trabalho dos investigadores, mas sim dar aos achados a dimensão adequada e evitar que sejam feitas especulações indevidas sobre o potencial de um ataque. Todos os achados do TPS são importantes e precisam ser devidamente tratados, pois afetam algumas das barreiras de segurança do processo eleitoral, direta ou indiretamente.

A análise feita neste relatório está limitada ao conjunto de software do Ecossistema da Urna e foi elaborado pela unidade técnica do Tribunal Superior Eleitoral - TSE responsável pelo seu desenvolvimento: a Seção de Voto Informatizado - Sevin. Destaca-se que o conjunto de software do Ecossistema da Urna é composto por todo o software executado pela urna eletrônica, conhecido como Uenux (composto por *bootloader*, kernel do Linux, drivers, bibliotecas e aplicativos), e pelo software de geração de mídias para a urna (Gedai-UE).

Ao final do relatório são apresentadas as ações levantadas pela Sevin que visam à mitigação dos achados do TPS 2017.

O objetivo deste documento é apresentar a visão da equipe técnica do TSE, tanto dos trabalhos realizados durante o TPS, quanto daquilo que precisa ser feito para mitigação das vulnerabilidades. Este documento não se sobrepõe ao relatório da Comissão Avaliadora do TPS, tampouco aos registros realizados pela equipe de acompanhamento e pelos próprios investigadores.

Na verdade, espera-se justamente a conjunção das visões do TSE, da Comissão Avaliadora, dos investigadores e da comunidade técnico-científica para a construção de sistemas eleitorais cada vez mais seguros.

Análise dos planos de teste executados

A seguir é feita uma análise dos trabalhos realizados durante o TPS 2017, tendo como referência os planos de teste apresentados pelos investigadores durante a fase de inscrição[1].

Grupo G1

O grupo liderado pelo Prof. Dr. Diego Aranha (Unicamp) contava também com a participação de Pedro Yossis Silva Barbosa (UFCG), Thiago Nunes Cardoso Carneiro (Hekima), Caio Lúders (UFPE) e Prof. Dr. Paulo Matias (UFSCar). Eles apresentaram quatro planos de teste na fase de inscrição, dos quais três focavam no conjunto de software do Ecossistema da Urna. Desses três planos originais, apenas um foi trabalhado até o fim pelos investigadores: **G1.1 - Captura de chaves criptográficas do Flash de Carga**. A proposta original consistia na decifração do sistema de arquivos de uma das partições da flash de carga - FC, com a finalidade de se obter acesso aos arquivos nela contidos, entre eles diversos arquivos de chaves de criptografia e assinatura.

Estritamente falando sobre o plano original, pode-se afirmar que houve sucesso parcial em sua execução, na medida em que não foi possível utilizar as chaves obtidas após a decifração bem sucedida do sistema de arquivos da FC. A utilização das chaves privadas não foi possível porque elas se encontram protegidas por mais um nível de criptografia. Nesse ponto é utilizado AES de 256 bits em modo CBC, cuja chave e vetor de inicialização encontram-se embarcadas no kernel do Linux.

Embora não tenha sido possível utilizar as chaves privadas, a decifração do sistema de arquivos abriu novas frentes de trabalho para o grupo, que em seguida apresentou novos planos de teste, deixando de lado os planos originais. A partir daí foram obtidos diversos achados, que serão detalhados a seguir.

A. Decifração do sistema de arquivos da flash de carga e de votação.

Utilizando software desenvolvido pelo próprio grupo, os investigadores foram capazes de decifrar e cifrar novamente o sistema de arquivos da urna. Isso foi possível com a utilização de informação obtida do ambiente de inspeção de código-fonte e de um dado fixo contido na mídia.

O sistema de arquivos utilizado é o Minix versão 1 com nomes de 30 caracteres. Foi criado um módulo de kernel derivado do original, no qual os blocos de dados dos arquivos são cifrados, mantendo os metadados em claro. É utilizado o algoritmo de criptografia AES de 256 bits em modo XTS para a criptografia do sistema de arquivos da urna.

A chave do sistema de arquivos é única e encontra-se embarcada no kernel do Linux. O vetor de inicialização é calculado a partir de um dado gravado em posição fixa de cada partição e do número do i-node de cada arquivo. Como o kernel do Linux encontra-se cifrado, a chave do sistema de arquivos encontra-se protegida dentro do seu binário. Além disso, a existência de uma única chave é uma propriedade necessária para viabilizar a troca de dados entre urnas, que

[1] http://www.tse.jus.br/hotsites/teste-publico-seguranca-2017/arquivos/tps-2017-relacao-dos-planos-de-teste-apos-recurso.pdf

é um recurso fundamental para os procedimentos de substituição de equipamentos defeituosos durante a votação ou para a recuperação de dados. Embora tenha passado por evoluções significativas ao longo dos anos, com total reescrita para 2017, a arquitetura do mecanismo de criptografia do sistema de arquivos é basicamente a mesma desde a sua implantação em 2009.

Como a chave do sistema de arquivos encontra-se embarcada no kernel do Linux, ela faz parte do seu código-fonte. Em consonância com o § 2º do Art. 66 da Lei 9.504 de 1997[2], todos os arquivos de código-fonte que continham chaves criptográficas foram retiradas do ambiente de inspeção de código-fonte. Por isso, o plano original previa um procedimento de engenharia reversa do kernel para obtenção da chave. Porém, por uma falha no procedimento de preparação do ambiente de inspeção do código-fonte, a chave de criptografia do sistema de arquivos permaneceu disponível nos computadores de visualização do código e foi vista pela equipe. O vazamento dessa chave no ambiente do TPS acelerou bastante os trabalhos da equipe de investigadores.

B. Alteração da biblioteca de log, com a modificação de texto fixo.

A partir da decifração do sistema de arquivos, os investigadores tiveram acesso às bibliotecas de link dinâmico da urna. Uma das bibliotecas usadas na infraestrutura de log foi modificada na FC e propagada para a flash interna - FI, sem apresentar qualquer tipo de alerta. Um dos textos que indicam a severidade do registro de log foi modificado de "INFO" para "XXXX". Dessa forma, o arquivo de log gravado pela urna ao final da votação continha "XXXX" no lugar de "INFO".

A infraestrutura de log da urna é baseada num *daemon* que recebe os eventos que devem ser registrados pelos demais aplicativos da urna. Esse *daemon* usa uma biblioteca de link dinâmico para as operações de escrita no arquivo de log.

Na urna eletrônica há dois mecanismos de assinatura digital que protegem a integridade e autenticidade de bibliotecas. O primeiro consiste na assinatura digital de seções de arquivos ELF. Trata-se de uma extensão do mecanismo de assinatura digital de módulo de kernel já disponível no Linux. Dessa forma, o kernel do Linux só coloca em execução os binários (módulos de kernel, bibliotecas de link dinâmico e aplicativos) que possuem assinatura digital válida. Esse mecanismo utiliza RSA de 4096 bits, na qual a chave pública é embarcada no kernel e a chave privada é destruída ao final do processo compilação e assinatura (lacração). Essa verificação é feita somente antes da execução de um binário.

Uma falha no mecanismo de validação de assinatura pelo kernel, introduzida durante o processo de atualização do Linux, permitiu que uma biblioteca modificada fosse executada normalmente. Embora o cálculo do RSA estivesse correto, a função que valida a assinatura retornava um inteiro com sinal com valor negativo para indicar falha de assinatura. Esse valor estava sendo atribuído incorretamente a uma variável do tipo inteiro sem sinal. Em seguida, outra função verificava por um valor negativo sobre a variável anterior para determinar a suspensão da execução do kernel. Esse bug tinha como efeito a execução de bibliotecas adulteradas.

[2] http://www.planalto.gov.br/ccivil_03/leis/L9504.htm

O segundo mecanismo de assinatura digital garante a integridade e autenticidade de qualquer arquivo na urna, independente da sua execução ou não. Trata-se de assinatura digital baseada em curvas elípticas de 256 bits desenvolvida pelo Cepesc/Abin. Essas assinaturas são geradas ao final do processo de lacração e somente a chave pública é incluída na urna (como arquivo numa das partições da FC e da FI). Cada assinatura é incluída num arquivo que contém uma lista de assinaturas de arquivos de um diretório. A verificação dessas assinaturas é feita por um *daemon*, de acordo com o solicitado pelo Software de Carga - SCUE[3] ou pelo Gerenciador de Aplicativos - GAP[4]. Caso seja encontrada uma assinatura inválida, o funcionamento da urna é interrompido.

Uma falha no conjunto de *scripts* do processo de lacração do TPS resultou na não inclusão da biblioteca de log na lista de assinaturas do seu respectivo diretório. Dessa forma, tanto o SCUE quanto o GAP não solicitaram que a assinatura digital dessa biblioteca fosse verificada. Com isso, o processo de carga foi concluído normalmente e o Software de Votação - VOTA foi executado sem qualquer indício preliminar de adulteração da biblioteca de log.

Num cenário real de uma eleição, a manipulação da biblioteca de log tal como realizada pelos investigadores poderia ser facilmente identificada. A execução do procedimento de conferência de hash detectaria a adulteração do biblioteca de log, que passaria a ter um hash diferente daquele publicado ao final da lacração[5]. O procedimento de verificação de assinatura digital com aplicativo próprio das entidades que auditam a lacração também detectaria falha de assinatura sobre a biblioteca de log.

Ademais, a presença de bug na validação de assinatura pelo kernel decorre do processo de atualização desse componente de software da urna. Embora esse mecanismo de assinatura esteja presente no software da urna desde 2009, com a atualização do kernel concluída em 2017, essa funcionalidade foi totalmente reescrita. Esse bug não encontra-se presente no kernel utilizado até 2016. Além disso, por se tratar de um novo arquivo[6], a biblioteca de log não foi incluída no processo de assinatura com o mecanismo desenvolvido pelo Cepesc/Abin — essa funcionalidade funciona corretamente. Em resumo, as falhas associadas à assinatura digital decorrem do atual estágio do desenvolvimento do software da urna.

C. Alteração da biblioteca de log, com a inclusão de código dos investigadores.

Como se mostrou possível alterar a biblioteca de log, os investigadores passaram para a tentativa de inserção de código nessa biblioteca. Então, o grupo obteve sucesso em inserir código capaz de fazer a leitura de teclas de um teclado padrão conectado a uma das portas USB da urna e

[3] Aplicativo executado pela urna para realização de seu processo de carga. Executado a partir da FC.

[4] Aplicativo executado pela urna para validações diversas e execução de outros aplicativos. Executado a partir da FI após a carga da urna.

[5] Procedimento executado com o apoio do aplicativo da urna chamado Verificação Pré e Pós Eleição - VPP.

[6] Em 2017 a infraestrutura de log da urna foi totalmente refeita, passando da manutenção de um arquivo binário com o registro dos eventos para um arquivo texto simples.

ecoar essas teclas no console do sistema operacional. Tratava-se de um laço infinito de chamada de sistema para leitura da entrada padrão e escrita desse conteúdo na saída padrão, ilustrada pela interação com o teclado — na prática, uma prova de conceito de que seria possível inserir código complexo feito pelos investigadores.

A falha na verificação de assinatura pelo kernel permitiu que a biblioteca de log com código adulterado fosse colocada em execução. Como o *daemon* de log não seguiu o seu fluxo de execução normal, isso impediu que o SCUE fosse executado e o console do sistema operacional permaneceu visível para a leitura daquilo que as aplicações escrevem na saída padrão. Finalmente, a manutenção do módulo HID Input no kernel (útil no ambiente de desenvolvimento, mas desnecessário no software lacrado) permitiu que um teclado externo funcionasse normalmente.

Nesse ponto, os investigadores também tentaram inserir código que fosse capaz de acesso ao sistema de arquivos de urna, para a potencial leitura ou modificação de arquivos e diretórios utilizados por outros processos. Também houve a tentativa de inclusão de um aplicativo de terminal na urna ou interpretador de comandos (*bash*, por exemplo). No entanto, as tentativas não foram bem sucedidas, entre outras razões, por esbarrarem no mecanismo de controle de acesso do Uenux, baseado no *Linux capabilities*.

D. Alteração da biblioteca de derivação de chaves criptográficas para execução de código dos investigadores.

A biblioteca utilizada para derivação de chaves criptográficas também estava sujeita às mesmas vulnerabilidades da biblioteca de log, relativas às suas assinaturas digitais. Essa biblioteca também foi incorporada ao software da urna em 2017 e é utilizada pelo VOTA.

A partir da possibilidade de execução de código dentro do espaço de memória do VOTA, os investigadores exploraram outras possibilidades.

1) Fixação da chave de criptografia do RDV.

A biblioteca de derivação de chaves é utilizada pelo VOTA para obtenção da chave de criptografia do RDV. Modificando essa biblioteca, os investigadores foram capazes de fazer o software utilizar uma chave deles no lugar daquela gerada exclusivamente pela urna, alterando a função de derivação de chaves para retornar um valor fixo. Com isso, o grupo foi capaz de decifrar o arquivo de RDV gravado na FI e na flash de votação - FV usando programa próprio.

O arquivo de RDV é mantido criptografado na FI e na FV com o algoritmo AES de 256 bits em modo CBC. A chave e o vetor de inicialização não estão no código-fonte e são derivadas a partir de uma informação contida exclusivamente no BIOS da urna.

O RDV é mantido criptografado na FI e na FV para impedir a sua leitura entre votações de diferentes eleitores. Embora os votos sejam gravados de forma aleatória entre os eleitores, caso fosse possível comparar duas versões do arquivo, antes e depois de uma votação, seria possível identificar um conjunto de votos recém inserido pela

diferença no conteúdo do arquivo entre esses dois momentos. Foi justamente essa a análise feita pelos investigadores.

Além da rastreabilidade da alteração da biblioteca de derivação de chaves através da conferência de hash e verificação externa de assinatura, conforme já explicado para o caso do log, é importante comentar também sobre a efetividade desse ataque do dia da eleição. Para que seja possível quebrar o anonimato de um conjunto de votos é necessário: a) desligar a urna antes da votação do eleitor alvo (aquele cujo anonimato deve ser quebrado); b) romper o lacre de acesso à FV e retirar a mídia; c) decifrar o sistema de arquivos e o RDV para então guardar uma cópia do arquivo; d) recolocar a FV e repôr o seu lacre; e) ligar a urna novamente para que o eleitor alvo faça o seu voto; f) desligar a urna novamente e repetir os passos b), c) e d); g) comparar as duas cópias do RDV, que terão como diferença o último conjunto de votos gravados, independente da sua ordem de gravação. Todos esses procedimentos precisariam ser feitos na seção eleitoral, que conta com quatro mesários, diversos eleitores, fiscais de partido e, no caso de celebridades, até da imprensa. Na hipótese de se mesclar os passos acima com o procedimento de contingência de urna ou de FV, vale destacar que esse procedimento não pode ser feito pelos mesários, que apenas podem requisitar esse trabalho a técnicos da Justiça Eleitoral. Os técnicos tentarão todos os procedimentos previstos em resolução específica do TSE e o ato será registrado na ata da seção e no log da urna.

Embora trate-se de um ataque muito difícil de ser colocado em prática, o cenário explorado pelos investigadores supera uma barreira de segurança criada justamente para mitigar essa possibilidade. Por isso, será devidamente tratada pela equipe técnica do TSE. Destaca-se também que o cenário explorado não possui paralelo com aquele observado no TPS 2012, quando foi atacado o mecanismo de embaralhamento dos votos. No TPS 2017 não foi explorado diretamente nenhum mecanismo de proteção dos votos no RDV, mas sim uma falha na verificação de assinatura digital da biblioteca associada a esses mecanismos.

Finalmente, embora tenha sido possível fazer a leitura do RDV, não foi possível modificar o conteúdo do arquivo. Isso porque o RDV está protegido por assinatura digital Cepesc/Abin e a sua validação funcionou perfeitamente.

2) Alteração de texto fixo de uma das telas do VOTA.

Dessa vez, a biblioteca de derivação de chaves foi alterada para modificar um campo de texto fixo numa telas do VOTA. Em particular foi alterado o texto “SEU VOTO PARA” para “VOTE EM 99” na tela de conferência dos dados do candidato (foto, número, nome…). Na prática, foi uma prova de conceito dos investigadores para demonstrar que conseguiam interferir de alguma forma na execução do VOTA e, sob esse aspecto, eles foram bem sucedidos.

Mais uma vez, esse cenário só foi possível devido à falha de assinatura digital da biblioteca, com o mesmo grau de rastreabilidade já comentado. Adicione ao fato somente

que qualquer eleitor seria capaz de ver a alteração e pedir algum tipo de intervenção na seção eleitoral nesse caso.

De qualquer forma, esse tipo de interferência não é aceitável e terá a sua causa raiz (assinatura digital da biblioteca) devidamente tratada.

3) **Tentativa (sem sucesso) de adulteração do voto.**

Aproveitando-se do fato de que a biblioteca de derivação de chaves tem acesso a todo o espaço de memória do VOTA, os investigadores tentaram modificar os votos dados pelo eleitor antes da sua gravação no RDV. Trata-se de uma expansão da técnica empregada para modificação de texto fixo numa tela do VOTA, porém, de dificuldade muito maior. Depois de três tentativas e esgotado o tempo do TPS, os investigadores não obtiveram sucesso, esbarrando em validações internas de consistência dos dados antes de sua gravação no arquivo pelo VOTA. Na última tentativa, os investigadores conseguiram "zerar" uma cédula de votação (estrutura de dados que armazena os votos em memória), o que provocou um erro de consistência e a suspensão do funcionamento do VOTA, sem que qualquer registro adulterado fosse gravado no RDV.

Grupo G4

O grupo liderado por Ivo Peixinho contava também com a participação de Fabio Caus Sicoli e Paulo Cesar Hermann Wanner, todos peritos da Polícia Federal. Eles apresentaram somente um plano de teste: **G4.1 - Extração de chave privada do Sistema Operacional da Urna Eletrônica**. Os investigadores procederam então com o emprego de técnicas de engenharia reversa para a obtenção da chave de criptografia do sistema de arquivos da urna, sendo bem sucedidos ao final do TPS.

Os investigadores foram capazes de alterar o código de *bootstrap* da MBR da FC para que fosse carregado o *bootloader* contido na mídia. Normalmente, o *bootstrap* da MBR contém código que faz o boot falhar quando a FC ou FI são utilizadas fora da urna. Alterada a MBR, foi possível iniciar o Uenux numa máquina virtual (no caso, QEMU e VirtualBox).

O boot do Uenux não foi completo, pois a inicialização falhou ao identificar o modelo da urna. No entanto, nesse ponto os investigadores foram capazes de fazer um *dump* da memória da máquina virtual e analisar o seu conteúdo. A partir daí foi possível localizar a chave de criptografia do sistema de arquivos, usando como referência o nome de um dos símbolos envolvidos no uso da chave.

Os investigadores usaram a chave vazada no ambiente de inspeção de código-fonte para validar o seu achado. Eles não fizeram nenhuma tentativa de ataque a outros mecanismos de segurança da urna a partir da decifração das mídias.

Evidentemente, há uma relação entre os trabalhos dos grupos G1 e G4. Mesmo sem o vazamento da chave no ambiente de inspeção de código-fonte, o sucesso na sua obtenção por processo de engenharia reversa indica que haveria algum grau de sucesso dentro daquilo que foi explorado pelo grupo G1 ao longo dos quatro dias de TPS. De qualquer forma, a possibilidade do

grupo G1 avançar sem engenharia reversa foi extremamente benéfica para o TPS, ao potencializar o número de achados.

Investigador I1

O investigador Cassio Goldschmidt apresentou somente um plano de testes durante a fase de inscrição: **I1.1 - Revisão de código e teste dinâmico de Geração das mídias para a preparação da urna eletrônica (GEDAI-UE)**. A partir da análise do código-fonte do aplicativo Gedai-UE, responsável pela geração das mídias da urna (FC, FI e MR), assim como do seu conjunto de arquivos de Makefile, o investigador fez as recomendações detalhadas a seguir. Destaca-se que não foi concretizada nenhuma tentativa de exploração de eventual vulnerabilidade que poderia ser mitigada com as recomendações apresentadas. Esse fato, contudo, não diminui a significância das recomendações fornecidas pelo investigador.

A) Uso de parâmetros de compilação do GCC associados a segurança.

Recomenda-se a adoção de diversos parâmetros (*flags*) disponíveis no GCC (compilador utilizado no conjunto de software do Ecossistema da Urna), que adicionam verificações automáticas de segurança no executável (proteção de pilha, por exemplo) e alertam sobre problemas de segurança comuns no código-fonte.

B) Validação do campo de comentário do cabeçalho de arquivos JPEG.

As fotos dos candidatos apresentadas pela urna estão no formato JPEG. Existem ataques de *buffer overflow* sobre o campo de comentários do cabeçalho do arquivo JPEG. Recomenda-se, portanto, que esse campo seja validado antes da abertura regular do arquivo.

Respostas aos achados do TPS

A edição de 2017 do TPS certamente foi aquela que trouxe o maior número de contribuições para o aperfeiçoamento do conjunto de software do Ecossistema da Urna. Todos os achados descritos anteriormente serão tratados até a conclusão do desenvolvimento do software que será utilizado nas Eleições 2018. Além da correção de bugs e implementação de melhorias em geral, uma série de procedimentos associados ao processo de desenvolvimento do software serão revistos, sobretudo quanto à rotina de testes.

As respostas receberam a seguinte classificação:

- **Curto prazo (C)**: conclusão até a primeira quinzena de janeiro.
- **Médio prazo (M)**: conclusão até o repetição dos testes executados no TPS 2017 (marco previsto no edital do TPS 2017, mas ainda sem data definida[7]).
- **Longo prazo (L)**: conclusão até a lacração das Eleições 2018.
- **Pós 2018 (P)**: conclusão após a lacração das Eleições 2018 (requer o descarte das urnas modelos UE2006/08); tratam-se de melhorias sobre as soluções de curto, médio e longo prazo.

Todas as ações dizem respeito ao software baseado naquele que foi apresentado no TPS 2017, cuja evolução resultará no software a ser utilizado nas Eleições 2018. Segue a lista de achados do TPS seguida das respectivas ações.

A) Chaves disponíveis no ambiente de inspeção de código:

1) Segregação das chaves contidas no código-fonte em *headers* separados (C).
2) Retirada das chaves contidas no código-fonte (M).

B) Bug na validação de assinaturas de bibliotecas pelo kernel:

1) Correção do bug (C).

C) Binários não incluídos em envelopes de assinatura Cepesc/Abin:

1) Correção dos *scripts* de assinatura usados no processo de lacração (C).
2) Criação de mecanismo unificado e centralizado que simplifique a adição de novos arquivos ao Uenux (M).
3) Minimização da quantidade de bibliotecas de link dinâmico no Uenux (M).
4) Implementação de mecanismo que garanta que a urna só contém os arquivos esperados e que aqueles que devem possuir assinatura estão devidamente assinados (L).

D) Acesso a chaves após engenharia reversa do *bootloader* e kernel:

1) Retirada das chaves contidas no código-fonte, baseado em mecanismo de derivação de chave mestre a partir de informação contida no BIOS da urna para criptografia de outras chaves (M).

[7] § 2º do Art. 37 do edital do TPS 2017, disponível em http://www.tse.jus.br/hotsites/teste-publico-seguranca-2017/arquivos/TPS-testes-publicos-seguranca-edital.pdf

2) Derivação de chave mestre a partir do MSD[8] (P).

E) Boot do Uenux no PC ou em máquina virtual:

1) Detecção de máquina virtual no *bootloader* e no kernel (M).
2) Validação do BIOS no *bootloader* e no kernel (M).
3) Validação do *bootstrap* do MBR pelo *bootloader* (L).

F) Uso de periférico não autorizado pelo Uenux:

1) Retirada do suporte a dispositivos não utilizados no kernel (USB HID Input) (C).

G) Ausência de *flags* de segurança do GCC:

1) Avaliação e adoção das *flags* (M).

H) Ausência de validação sobre o campo de comentário do cabeçalho do arquivo JPEG:

1) Exigir que o comentário tenha tamanho igual a zero (M).

Como parte das ações listadas acima, todos os processos de teste e verificação associados serão revistos e aprimorados.

A lista de ações apresentada não é definitiva e poderá sofrer alterações ao longo do processo de desenvolvimento. De qualquer forma, a sua implantação poderá ser auditada pelos mecanismos previstos na resolução de fiscalização das eleições, em especial durante os seis meses que antecedem a Cerimônia de Lacração e Assinatura Digital dos Sistemas Eleitorais e durante a cerimônia em si.

Os investigadores também serão convocados oportunamente para verificar as correções implementadas e executar novamente os seus planos de teste, com vistas a comprovar que as falhas foram tratadas.

[8] *Master Secure Device* - dispositivo de segurança embarcado na urna, responsável pelas verificações da cadeia de segurança do boot, e que possui a capacidade de geração e guarda segura de chaves criptográficas.

Conclusão

A edição do TPS de 2017 contou com a presença de pesquisadores e profissionais altamente qualificados em técnicas de criptografia, desenvolvimento de software seguro e engenharia reversa. Os achados dos investigadores provocarão correções e melhorias fundamentais para que o conjunto de software do Ecossistema da Urna esteja num patamar ainda mais elevado de segurança e robustez para as Eleições 2018.

As ações apresentadas aqui para a mitigação das falhas encontradas serão implementadas a tempo das Eleições 2018, incluindo até uma trilha de evolução após as próximas eleições. A equipe da Sevin entende, contudo, que as ações listadas neste documento podem não ser definitivas e está aberta a sugestões dos próprios investigadores e da comunidade técnico-científica em geral. Críticas e sugestões podem ser enviadas para sevin@tse.jus.br.