Testes Públicos de Segurança do Sistema Eletrônico de Votação

Eleições 2012

Brasília, março de 2012

Relatório dos resultados da realização dos Testes Públicos

Grupo 04

Representando a Universidade de Taubaté - UNITAU

Luís Fernando de Almeida – Doutor em Metodologia e Técnicas da Computação – UNESP
Bárbara Maximino da Fonseca Reis – Graduada em Engenharia da Computação – UNITAU
João Cristiano Monteiro Silva – Graduado em Engenharia da Computação – UNITAU
Luís Felipe Feres Santos – Graduado em Engenharia da Computação
Rafael Kudaka de Oliveira – Graduado em Sistemas de Informação – UNITAU

Plano de Teste G4PT1

Injeção de código e violação da rotina de aleatoriedade

Conteúdo deste relatório

Grpo 4-1

TESTES PÚBLICOS DE SEGURANÇA DO SISTEMA ELETRÔNICO DE VOTAÇÃO ELEIÇÕES 2012

Plano de Teste do Sistema Eletrônico de Votação

1 Informações gerais

Título do plano de teste	Injeção de Código e Violação da Rotina de Aleatoriedade
Instituição proponente (se aplicável)	Universidade de Taubaté
Responsável	nome: Luis Fernando de Almeida e-mail: luis.almeida@unitau.br telefone (do autor ou responsável): (12) 3625-4256, (12) 3629-5982, (12) 8113-5754
Sistemas afetados	**Software:** *Software* de votação usado nas seções eleitorais. **Hardware:** ☐ Microterminal ☒ Terminal do eleitor ☐ Lacres ☐ Mídias **Procedimentos:** ☐ Carga da urna ☒ Votação
Duração estimada do teste (em minutos)	300
Extensão do ataque	☐ Urna ou seção eleitoral ☐ Local de votação ☐ Zona eleitoral ☐ Município ☐ Unidade da Federação ☒ País
Conhecimentos necessários	Arquitetura do Sistema Operacional, Compilador Utilizado para geração do Sistema da Urna, Linguagem de Programação.

Observações:
- O teste a ser realizado deve, obrigatoriamente, ser reproduzível.
- Este plano deverá ter no máximo dez páginas em formato A4 ou Carta.

2 Reservado ao Tribunal Superior Eleitoral (TSE)

Protocolo	Data
	Resultado ☐ Aprovado ☐ Aprovado com ressalvas ☐ Reprovado

3 Detalhamento do teste

3.1 Resumo do teste

Considerando que o processo de compilação do programa da Urna Eletrônica faz uso de tecnologia Open Source, o presente teste propõe a simulação de injeção de código a partir do compilador g++ ou alteração das bibliotecas padrões a fim de mapear a rotina de gravação dos votos do eleitor. Outro aspecto seria o mapeamento da rotina aleatória, permitindo o mapeamento do eleitor.

3.2 Fundamentação

Considerando a abordagem realizada pelo NIST SP800-90A, para realização e determinação de números randômicos, o teste procura mapear a função aleatória para inserção de votos e confrontá-la com a norma citada.

Como a distribuição utilizada é de domínio público, poderia ser levada em consideração a alteração do repositório da mesma, incluindo código malicioso a partir do compilador, que alteraria o software final utilizado na Urna Eletrônica. Desse modo, o atacante poderia associar o voto ao eleitor, comprometendo o sigilo do voto.

3.3 Precondições para o teste

Recursos Humanos:

- Especialista na Arquitetura do Compilador;
- Domínio da Estrutura do Sistema Operacional;
- Especialista em Desenvolvimento.

Recursos Materiais:

- Distribuição Linux utilizada na etapa de compilação do software da Urna, com o compilador apropriado;
- Acesso ao código fonte do programa da Urna;

3.4 Escopo – Superfície de Ataque

Atingir o programa da Urna, propriamente o módulo "*Vota*", e assim possibilitar a quebra do sigilo do voto. Para que isso seja possível, deve-se considerar um olheiro identificando a sequência de votantes e um agente interno que permita a manipulação da arquitetura do Sistema Operacional.

3.5 Janela de atuação simulada do atacante

O teste proposto considera um agente interno malicioso, com a capacidade de alterar as rotinas do compilador e gerar a brecha no módulo "*Vota*". Após essa etapa, a necessidade de outro agente malicioso identificando os eleitores, para em seguida associar os votos realizados.

3.6 Pontos de intervenção

Alterar a estrutura do compilador de forma imperceptível durante o processo de compilação final. Não revelar as alterações durante a fase de teste da Urna. Não ser identificado pelos processos de gestão de informação aplicados pela equipe de desenvolvimento.

3.7 Passos a serem realizados e material necessário

1. Alteração da estrutura do Sistema Operacional, possibilitando a inclusão de bibliotecas maliciosas e a estrutura do compilador (60 minutos);
2. Realizar a instalação do compilador e bibliotecas no ambiente de compilação final (30 minutos);
3. Compilar o código fonte da Urna no ambiente alterado (60 minutos);
4. Carregar o software na Urna Eletrônica (30 minutos);
5. Simular o ambiente de votação, anotando a sequência de votantes (45 minutos);
6. Gerar o boletim com a quantidade de eleitores cadastrados, quantidade de votantes e a tabela de votos (30 minutos);
7. Mapear a rotina de aleatoriedade, possibilitando a associação de votos (45 minutos);
8. Fim.

Material necessária:

- Distribuição Linux, compatível a utilizada durante a etapa final de compilação;
- Acesso administrativo ao componentes do Sistema Operacional.
- Ferramentas de compilação e depuradores;
- Ambiente de desenvolvimento (Eclipse ou NetBeans), com suporte a desenvolvimento em linguagem C/C++;

3.8 Possíveis resultados e impacto

Resultado Esperado:

- Rompimento do sigilo do voto.

Extensão do Ataque:

- Pais.

3.9 Rastreabilidade

Para estimar as chances de sucesso, o atacante deve considerar como é realizada a gestão da informação no processo de desenvolvimento e o quanto sofisticado esse processo está. Para passar imperceptível, na fase de teste de software o algoritmo injetado não deve realizar nenhum processo.

Chances de sucesso: 40%
Detectar o ataque: 60%

3.10 Solução proposta

Para mitigar o ataque proposto, deve-se adotar um plano de validação e conformidade de segurança na distribuição Linux utilizada e o compilador do mesmo. Uma possibilidade seria verificar se o checksum oferecido pelo integrador do sistema seja o mesmo que foi utilizado no processo de compilação. Também simular testes de aleatoriedade, considerando o período de votação, para identificar se a rotina aleatória está desempenhando o papel esperado.

TESTES PÚBLICOS DE SEGURANÇA DO SISTEMA ELETRÔNICO DE VOTAÇÃO ELEIÇÕES 2012

Formulário de Acompanhamento dos Testes Públicos

Dados do Grupo de Investigadores			
G4PT1	Coordenador:	Luís Fernando de Almeida	
	Investigador 1:	Barbara Maximino F. Reis	
	Investigador 2:	João Cristiano Monteiro Silva	
	Investigador 3:	Luís Felipe Féres Santos	
	Investigador 4:	Rafael Kudaka de Oliveira	

Informações do Acompanhamento					
Data:	23 / 03 / 12	Hora de Início:	- : -	Hora de Término:	- : -
Resp. Acomp.:	Fausto Filho			Rubrica:	

Dados do Teste		
Titulo do teste:	Injeção de código e violação da rotina de aleatoriedade	
Inicio do teste (Data/Hora):	- / - / -	- : -
Termino do teste (Data/Hora):	- / - / -	- : -
Criterio de Parada:	NÃO HÁ.	

Relaxamento nos mecanismos e procedimentos de segurança
NÃO HÁ.

Etapas Propostas para o Teste		
Etapa	Descrição	Status
1	NÃO HÁ	
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		

Acompanhamento dos Procedimentos

Hora	Procedimentos realizados durante o teste
—:—	plano não realizado em virtude do indeferimento
:	por parte da Comissão.
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	
:	

Conclusões sobre o teste

Futuras Possibilidades

Informações Adicionais