Universidade Federal de Pernambuco - UFPE
Centro de Informática - CIn
Graduação em Ciência da Computação

Predição de números pseudo-aleatórios gerados em .NET por System.Random

Proposta de Trabalho de Graduação

Aluno: Lucas Miranda Lin (lml@cin.ufpe.br)
Orientador: Paulo André da Silva Goncalves(pasg@cin.ufpe.br)
Área: Segurança da informação

Recife, 2021

Sumário

Contexto

C# é uma das linguagens mais populares do mundo, em quarto lugar segundo índices recentes [1], usada em desenvolvimento de jogos, aplicações para desktop, aplicações web, APIs REST, IDEs, dentre outros. O gerador de números pseudoaleatórios (PRNG) padrão dessa linguagem, a classe System.Random, é inseguro segundo sua própria documentação [2]. Assim como em várias outras linguagens [3][4][5], optou-se por um gerador padrão simples e de rápida execução, porém inseguro.

A plataforma .NET oferece outros PRNGs apropriados e recomendados para situações nas quais a previsibilidade do gerador é inaceitável [6][7]. Porém, estas alternativas são relativamente obscuras e de usabilidade programática inferior a System.Random. Isso abre brecha para que desenvolvedores desatentos, sem instrução de segurança, ou outrora indispostos deslizem e façam uso de System.Random em sistemas críticos de segurança, deixando suas aplicações expostas a ataques que explorem a previsibilidade deste gerador.

Objetivos

O algoritmo de geração pseudoaleatória implementado por System.Random é conhecido, e prever os números por este gerados é trivial [8]. Porém, não encontramos em via pública um ataque de predição apropriado a um caso de uso específico, o qual nomeamos de *descartável*. Neste caso de uso, uma nova instância da classe é criada para gerar cada número, e em seguida descartada.

Este trabalho objetiva desenvolver uma ferramenta que seja capaz de prever números gerados por System.Random em casos de uso *descartáveis*, além de uma aplicação-alvo exemplar para fins de testes e demonstração. A ferramenta poderá ser utilizada para verificar se sistemas rodando na plataforma .NET estão vulneráveis ao respectivo ataque, e ressalta a necessidade de evitar o uso de System.Random em contextos sensíveis de segurança.

Metodologia

O código e o comportamento da classe System.Random serão analisados para o desenvolvimento de um ataque preditor. A ferramenta preditora será uma CLT (Command Line Tool) escrita em Python; e a aplicação vulnerável exemplo será escrita em C#, com a vulnerabilidade presente especificamente em um mecanismo 2FA. Testes automatizados serão escritos para verificar a efetividade da ferramenta contra a aplicação exemplar sob várias condições e configurações.

Cronograma

Atividade	Maio	Junho	Julho	Agosto
Pesquisa sobre o tema				
Desenvolvimento do projeto				
Elaboração da monografia				
Defesa				

Possíveis Avaliadores

Sugerimos como possíveis avaliadores:

- José Augusto Suruagy Monteiro (suruagy@cin.ufpe.br)
- Paulo G.S. da Fonseca (paguso@cin.ufpe.br)

Referências

1. **PYPL PopularitY of Programming Language**. Disponível em https://pypl.github.io/PYPL.html. Acesso em 2 de junho de 2021.
2. **Random Class**. Disponível em https://docs.microsoft.com/en-us/dotnet/api/system.random. Acesso em 2 de junho de 2021.
3. **Random (Java Platform SE 8)**. Disponível em https://docs.oracle.com/javase/8/docs/api/java/util/Random.html. Acesso em 2 de junho de 2021.
4. **random — Generate pseudo-random numbers — Python 3.9.5 documentation**. Disponível em https://docs.python.org/3/library/random.html. Acesso em 2 de junho de 2021.
5. **rand - cpp reference**. Disponível em https://en.cppreference.com/w/c/numeric/random/rand. Acesso em 2 de junho de 2021.
6. **RNGCryptoServiceProvider Class**. Disponível em https://docs.microsoft.com/en-us/dotnet/api/system.security.cryptography.rngcryptoserviceprovider. Acesso em 2 de junho de 2021.
7. **RandomNumberGenerator Class**. Disponível em https://docs.microsoft.com/en-us/dotnet/api/system.security.cryptography.randomnumbergenerator. Acesso em 2 de junho de 2021.
8. SPORICI, Dan. **C# Predict the Random Number Generator of .NET**. Disponível em https://codingvision.net/c-predict-random-number-generator-net. Acesso em 2 de junho de 2021.