

Мова Асемблера
Mova Asemblera
x64 & arm64

Niezwykłe anomalie
w językach C#/IL

Podstawy PowerShell dla użytkowników
systemów Microsoft Windows

Technologia łańcucha bloków
(ang. blockchain) z przykładami
w języku C#

Dawid Farbaniec



0x3e41

ethical.blue Magazine to czasopismo edukacyjne oraz pracownia artystyczna w klimacie postapokaliptycznym.

Dawid Farbaniec
magazine@ethical.blue
Rymanów Zdrój, Subcarpathia, Poland

Wszystkie materiały zawarte w tym czasopiśmie są chronione prawem autorskim. Kopiowanie i rozpowszechnianie opublikowanych tu materiałów bez zgody autora jest surowo zabronione. Wszystkie opublikowane tutaj materiały (w tym kody źródłowe i programy) mają charakter informacyjny i powstały wyłącznie w celach edukacyjnych. Autor niniejszego czasopisma nie ponosi odpowiedzialności za nielegalne wykorzystanie udostępnionych tu materiałów. Czytelnik niniejszego magazynu oświadcza, że wykorzystuje udostępnione materiały na własne ryzyko. Wszystkie znaki towarowe i zarejestrowane nazwy zostały użyte wyłącznie w celach informacyjnych i należą wyłącznie do ich prawnych właścicieli. Czcionki Agency FB (learn.microsoft.com/en-us/typography/font-list/agency-fb), Archivo ([3fonts.google.com/specimen/Archivo/license](https://fonts.google.com/specimen/Archivo/license)), Open Sans (fonts.google.com/specimen/Open+Sans/license), Papyrus (learn.microsoft.com/en-us/typography/font-list/papyrus) oraz Tangerine (fonts.google.com/specimen/Tangerine/license) zostały użyte zgodnie z licencją dostarczoną od dystrybutora. Autor tego magazynu, w momencie tworzenia materiału nie działa w imieniu firm, których technologie lub produkty opisuje — za wyjątkiem, gdzie zostało to wyraźnie oznaczone. Produkty i rozwiązania opisywane w tekstach są wybierane losowo. Autor nie otrzymał żadnych pieniędzy za opisanie tych produktów lub rozwiązań — za wyjątkiem, gdzie jest to wyraźnie zaznaczone w tekście.

Teksty

Pierwszy dzień w strefie i spotkanie 3°	4
Otchłań zapomnienia. Udokumentowana anomalia ze strefy 3OHA	5
Elfickie ostrze. Podstawy PowerShell dla użytkowników systemu Microsoft Windows	6
Anomaly.IL.ethical.blue.LNK.Z!3050	18
Dziwna mgła	20
Wrak pojazdu	23
Skrypt C2.E!undead	25
Nieskończone podziemia	27
Zapomniana wiedza. Asembler x86-64 (x64) w systemach Microsoft Windows	29
Fotografia zjawiska paranormalnego ze strefy 3OHA	78
Niezwykłe anomalie w językach C#/IL dla platformy .NET	80
Śpiący motyl nocny	83
Czarna korozja. Anomalia	103
Nieznane rozkazy. Asembler arm64 w systemach Microsoft Windows	105
Żywiołek czasu	133
Leśny sad. Asembler arm32/arm64 na urządzeniach Raspberry Pi	134
Niezidentyfikowany obiekt latający	142
Technologia łańcucha bloków (ang. blockchain) z przykładami w języku C#	143
Klucz do skarbcza z cyfrowym złotem	152
Polana za klasztorem (poza strefą 3OHA)	155

Wykaz literatury

158

Zjawisko paranormalne
ethical.blue / Art

David Farbaniec





Niektórzy podejrzewają, że fotografia jest fałszyfikatem. Wielu jednak twierdzi, że to nieznana siła nagle zaczęła trawić metalowe zamknięcie, pozwalając niesłusznie skazanym wydostać się z więzienia.



Vulnerable.C#/IL!Random.A

```
/*- Vulnerable.C#/IL!Random.A  -*\
/*- ethical.blue Magazine      -*/

using System.Security.Cryptography;

byte[] buffer = [0x00];

var randomGenerator =
    RandomNumberGenerator.Create();
randomGenerator.GetBytes(buffer);
T($"0x{buffer[0]:X2}");
buffer[0] = 0x00; // avoid leak

#region Vulnerable (Predictable)
    int seed = 0xBAD;
    var random = new Random(seed);
    random.NextBytes(buffer);
    T($"0x{buffer[0]:X2}");
#endregion

static void T(string szT) =>
    typeof(Console).Assembly
        .GetType("\x53\x79\x73"
            + "\x74\x65\x6D\x2E\x43"
            + "\x6F\x6E\x73\x6F\x6C\x65")?
        .GetMethod("\x57\x72\x69\x74"
            + "\x65\x4C\x69\x6E\x65",
            [typeof(string)])?
        .Invoke(null, [szT]);
```

Konsola debugowania

```
0xEA
0x43
```

Aplikacje mogą korzystać z losowych wartości w funkcjach kryptograficznych i różnych mechanizmach zabezpieczających. Z tego powodu bardzo ważne jest, aby generowane wartości były jak najmniej możliwe do przewidzenia.

Krytycznym błędem jest używanie ziarna (ang. seed), które da się przewidzieć lub uzyskać w inny sposób. Podatność tego typu powoduje, że losowe wartości nie są wystarczająco bezpieczne kryptograficznie.

W celu złagodzenia zagrożenia należy korzystać ze specjalnych klas kryptograficznych, takich jak m.in. `System.Security.Cryptography.RandomNumberGenerator`, zamiast zwykłego `System.Random`.

Vulnerable.C#/IL!NaN.eq.A

```
/*- Vulnerable.C#/IL!NaN.eq.A  -*\
/*- ethical.blue Magazine      -*/

T("0.00 / 0.00 == 0.00 / 0.00");
T("${0.00 / 0.00 == 0.00 / 0.00}");

T(Environment.NewLine);

T("float.NaN == float.NaN");
T("${float.NaN == float.NaN}");

T(Environment.NewLine);

T("float.IsNaN(float.NaN)");
T("${float.IsNaN(float.NaN)}");

static void T(string szT) =>
typeof(Console).Assembly
.GetType("\x53\x79\x73"
+ "\x74\x65\x6D\x2E\x43"
+ "\x6F\x6E\x73\x6F\x6C\x65")?
.GetMethod("\x57\x72\x69\x74"
+ "\x65\x4C\x69\x6E\x65",
[typeof(string)])?
.Invoke(null, [szT]);
```

Konsola debugowania

```
0.00 / 0.00 == 0.00 / 0.00
False
```

```
float.NaN == float.NaN
False
```

```
float.IsNaN(float.NaN)
True
```

NaN (ang. Not a Number) to wartość, która nie jest równa żadnej wartości, a nawet samej sobie.

W celu weryfikacji czy w wyniku powstał NaN należy używać specjalnej metody o nazwie IsNaN.



**Kup pełne czasopismo na ethical.blue
i dołącz do odkrywców strefy 3OHA!**

Polana za klasztorem (poza strefą 30HA)

ethical.blue / Art

Dawid Farbaniec



**Kup pełne czasopismo na ethical.blue
i dołącz do odkrywców strefy 3OHA!**

Wykaz literatury

- [1] Advanced Micro Devices, Inc. (AMD), *AMD64 Architecture Programmer's Manual*, 2024.
- [2] Advanced Micro Devices, Inc. (AMD), *System V Application Binary Interface*, 2025.
- [3] Arm Limited, *Arm Architecture Reference Manual for A-profile architecture*, 2025.
- [4] Arm Limited, *ELF for the Arm 64-bit Architecture (AArch64)*, 2025.
- [5] Ecma International, *Common Language Infrastructure (CLI)*, 2012.
- [6] Ecma International, *C# Language Specification*, 2023.
- [7] Ethereum Foundation, <https://ethereum.org/>, 2013.
- [8] Intel Corporation, *Intel Advanced Vector Extensions 10.2 Architecture Specification*, 2025.
- [9] Intel Corporation, *Intel Architecture Instruction Set Extensions and Future Features*, 2025.
- [10] Intel Corporation, *The Intel 64 and IA-32 Architectures Software Developer's Manual*, 2025.
- [11] ISO/IEC, *Information technology. Common Language Infrastructure (CLI)*, 2012.
- [12] ISO/IEC, *Information technology. Programming languages. C#*, 2018.
- [13] ISO/IEC, *Programming languages. Avoiding vulnerabilities in programming languages*, 2024.
- [14] Microsoft Corporation, *Windows PowerShell Language Specification Version 3.0*, 2012.
- [15] National Institute of Standards and Technology (NIST), *Blockchain Technology Overview*, 2018.
- [16] National Institute of Standards and Technology (NIST), *Report on Post-Quantum Cryptography*, 2016.
- [17] National Institute of Standards and Technology (NIST), *SHA-3 Standard*, 2015.
- [18] National Institute of Standards and Technology (NIST), *Specifications for Secure Hash Standard*, 2015.