SHUBHKEY
SOFTWARE HUB OF KEY IT SERVICES

MICROSOFT DEVELOPERS COMMUNITY
MDCA
AHMEDABAD

4th DOT NET DAY

Dot net Library : Beyond Imagination & Noteworthy

SHUBHKEY
SOFTWARE HUB OF KEY IT SERVICES

MDCA
AHMEDABAD

SPEAKER

TOPIC

Dot net Library : Beyond Imagination & Noteworthy

Sandeep Kumar Das

CEO & Founder

Shubhkey Infotech LLP

Who ?

- ✓A CEO – Technopreneur
- ✓A Mentor – By Deeds
- ✓A Speaker – By Passion
- ✓A Developer – By Heart
- ✓20+ Years of Experience

TECHNOLOGY EXECUTIVE COUNCIL

Microsoft has a $20 billion hacking plan, but cybersecurity has a big spending problem

PUBLISHED WED, SEP 8 2021•10:04 AM EDT | UPDATED WED, SEP 8 2021•1:29 PM EDT

Eric Rosenbaum
@ERPROSE

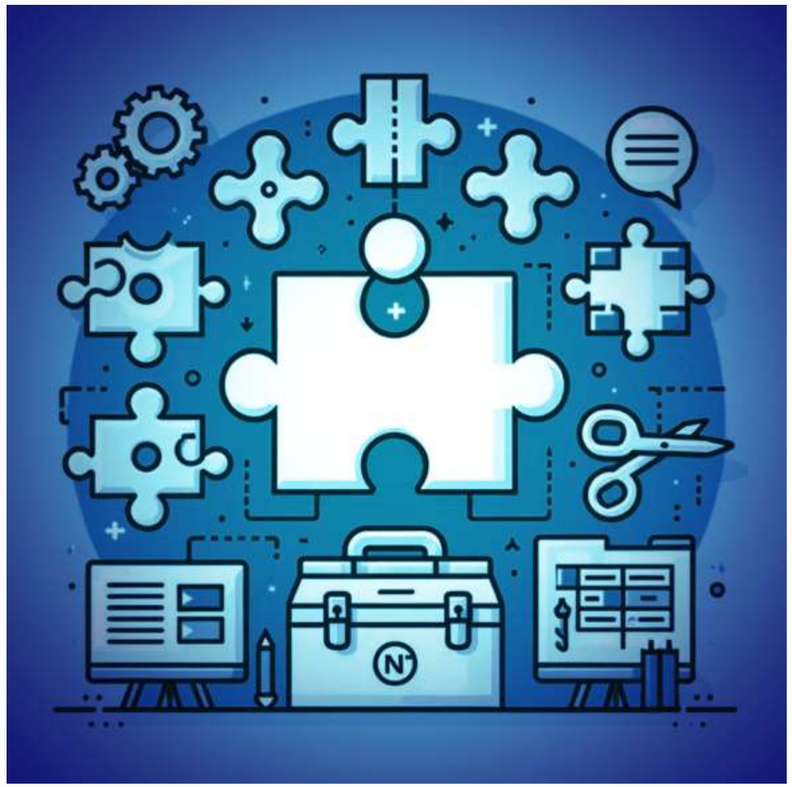
SHARE

KEY POINTS

- Microsoft is quadrupling its cybersecurity investment to $20 billion over the next five years.
- One of the reasons for the big investment cited by Microsoft president Brad Smith in a CNBC interview this week speaks to a Catch-22 in the cyber arms race: the increased spending in recent years by public and private enterprises hasn't resulted in better protection against criminal hackers.
- The shortage of workers skilled in cybersecurity is one of the factors that has led to a situation in which companies are paying for products that in many cases they aren't even using.

RELATED

Versatility and Flexibility of Dot Net Library



```
using System.Runtime.InteropServices;

using var watcher = new System.IO.FileSystemWatcher(@"D:\USGeographical");

watcher.NotifyFilter = NotifyFilters.Attributes
                     | NotifyFilters.CreationTime
                     | NotifyFilters.DirectoryName
                     | NotifyFilters.FileName
                     | NotifyFilters.LastAccess
                     | NotifyFilters.LastWrite
                     | NotifyFilters.Security
                     | NotifyFilters.Size;

watcher.Changed += OnChanged;
watcher.Created += OnCreated;
watcher.Deleted += OnDeleted;
watcher.Renamed += OnRenamed;
watcher.Error += OnError;

//watcher.Filter = "*.txt";
watcher.IncludeSubdirectories = true;
watcher.EnableRaisingEvents = true;
```

Versatility and Flexibility of Dot Net Library

```
string str = "";
for (int i = 0; i < 100; i++)
{
    Guid sequentialGuid = SequentialGuidGenerator.NewSequentialId();
    Console.WriteLine(sequentialGuid);
    str = str + sequentialGuid.ToString() + "\n";
}
Console.WriteLine(str);
```

```
1 reference
public static class SequentialGuidGenerator
{
    [DllImport("rpcrt4.dll", SetLastError = true)]
    1 reference
    private static extern int UuidCreateSequential(out Guid guid);

    1 reference
    public static Guid NewSequentialId()
    {
        Guid guid;
        UuidCreateSequential(out guid);
        return guid;
    }
}
```

ImageSharp

A library for processing images in .NET applications. ImageSharp is fully managed and cross-platform, providing extensive functionality for image manipulation.

ImageSharp Middleware for serving images via a url based API

It is designed to be flexible, extensible, and easy to use, offering various functionalities for image manipulation, such as resizing, cropping, filtering, and more.

ML.Net

Microsoft's open-source and cross-platform machine learning framework of Dot Net Library

ML.NET enables developers to implement machine learning tasks, including complex algorithms for predictions, anomaly detection, natural language processing, and more, directly within .NET applications.

SignalR

A library for adding real-time web functionality to applications. SignalR enables bi-directional communication between server and client. It's widely used for real-time chat applications, live updates, and interactive web features.

- WebSockets:
- Server-Sent Events (SSE):
- Long Polling:

Open Source Power

Akka.NET

An open-source toolkit and runtime for building highly concurrent, distributed, and fault-tolerant event-driven applications on .NET. It's inspired by the Akka framework from the Java Virtual Machine.

Akka.NET utilizes the actor model. Actors are objects that encapsulate state and behavior, communicate through messages, and run concurrently, making it easier to reason about complex concurrent processes.

Autofac

A popular dependency injection library. It's known for its advanced features and flexibility in managing object lifetimes and dependencies in .NET applications.

Microsoft does have its own dependency injection framework, which is included in .NET Core and later versions of the .NET framework, but Autofac remains a popular choice for developers seeking additional features or specific behaviors not covered by Microsoft's built-in dependency injection support.

Hangfire

Fire-and-Forget Jobs

Fire-and-forget jobs are executed **only once** and almost **immediately** after creation.

```
var jobId = BackgroundJob.Enqueue(
    () => Console.WriteLine("Fire-and-forget!"));
```

Delayed Jobs

Delayed jobs are executed **only once** too, but not immediately, after a certain **time interval**.

```
var jobId = BackgroundJob.Schedule(
    () => Console.WriteLine("Delayed!"),
    TimeSpan.FromDays(7));
```

Recurring Jobs

Recurring jobs fire **many times** on the specified **CRON schedule**.

```
RecurringJob.AddOrUpdate(
    "myrecurringjob",
    () => Console.WriteLine("Recurring!"),
    Cron.Daily);
```

Continuations

Continuations are executed when its parent job **has been finished**.

```
BackgroundJob.ContinueJobWith(
    jobId,
    () => Console.WriteLine("Continuation!"));
```

Batches Pro

Batch is a group of background jobs that is **created atomically** and considered as a single entity.

```
var batchId = BatchJob.StartNew(x =>
{
    x.Enqueue(() => Console.WriteLine("Job 1"));
    x.Enqueue(() => Console.WriteLine("Job 2"));
});
```

Batch Continuations Pro

Batch continuation is fired **when all** background jobs in a parent batch **finished**.

```
BatchJob.ContinueBatchWith(batchId, x =>
{
    x.Enqueue(() => Console.WriteLine("Last Job"));
});
```

A library for background job processing. Hangfire allows you to run background tasks in .NET applications, and it's known for its ease of use and reliability.

Using Hangfire, developers can significantly reduce the complexity of background job processing in .NET applications, improving the reliability, scalability, and maintainability of their applications.

- Email : sandeep.das@shubhkey.com
- Mobile : **+91 9825843993**

SHUBHKEY
SOFTWARE HUB OF KEY IT SERVICES

MDCA
AHMEDABAD

Questions ? & Answers !

SPONSORS

SHUBHKEY
SOFTWARE HUB OF KEY IT SERVICES

THANK YOU

Stay Tuned For Next Dot Net Day

MICROSOFT DEVELOPERS COMMUNITY
MDCA
AHMEDABAD

SHUBHKEY
SOFTWARE HUB OF KEY IT SERVICES

MDCA
AHMEDABAD

You can find event photos on Memorylens AI app.