

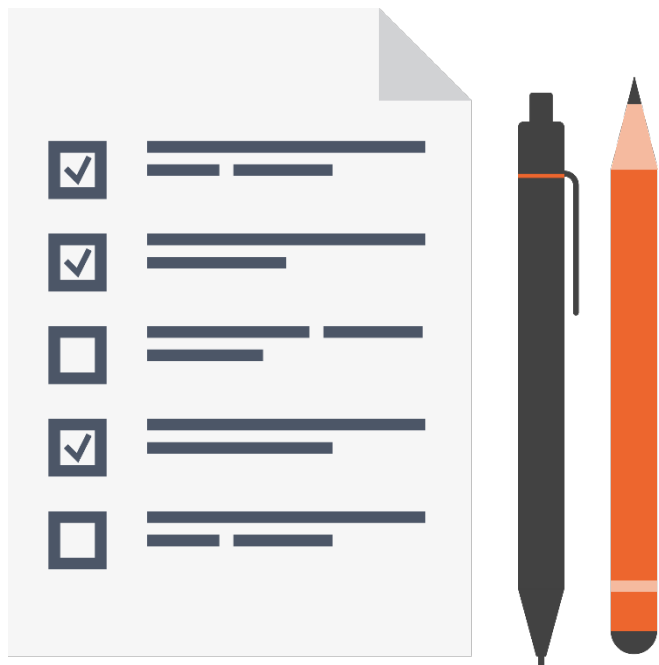
# SecureString



Stephen Haunts

@stephenhaunts | [www.stephenhaunts.com](http://www.stephenhaunts.com)

# Overview



- What is SecureString?
- Windows Data Protection API
- SecureString code demonstration

# Using SecureString for Sensitive Data

- System.String is not a secure solution
- System.String has the following problems
  - Several copies in memory
  - Not encrypted
  - Not mutable, old copied in memory
  - No effective way to clear out memory

# Using SecureString for Sensitive Data

- SecureString stored in encrypted memory
- SecureString implements IDisposable
- Create SecureString with a pointer to a char array

# Using SecureString for Sensitive Data

## SecureString

---

AppendChar()

InsertAt()

RemoveAt()

SetAt()

MakeReadOnly()

IsReadOnly()

Clear()

Dispose()

# Using SecureString for Sensitive Data

```
public static string ConvertToUnsecureString(SecureString securePassword){  
  
    var unmanagedString = IntPtr.Zero;  
    try {  
        unmanagedString =  
            Marshal.SecureStringToGlobalAllocUnicode(securePassword);  
        return Marshal.PtrToStringUni(unmanagedString);  
    }  
    finally {  
        Marshal.ZeroFreeGlobalAllocUnicode(unmanagedString);  
    }  
}
```

# Data Protection API (DPAPI)

- Operating system level data protection
- Provides confidentiality using encryption
- DPAPI is a password based data protection service
- Is part of Crypt32.dll
- DPAPI doesn't store your data for you
- DPAPI uses the user's logon credential
- Generates a master key
- Master keys have an expiry date

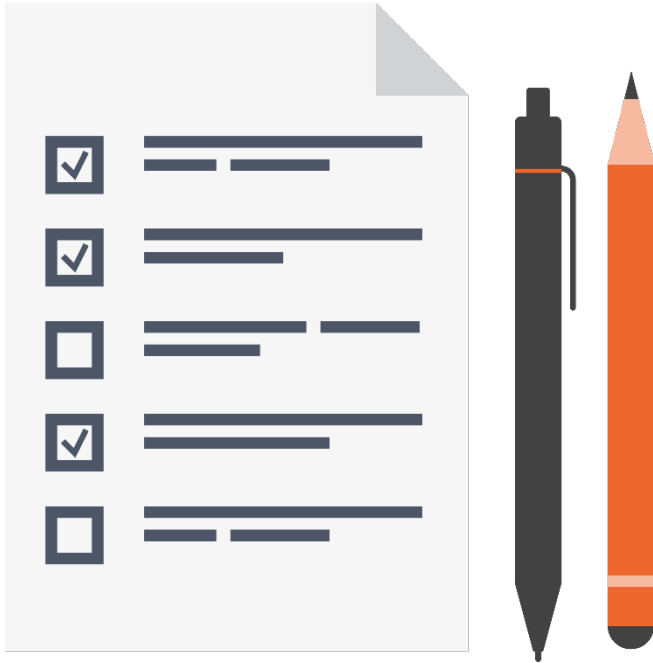
---

# Code Demonstration

Using SecureString



# Module Summary



- Use SecureString to store sensitive information
  - Passwords
  - Credit card numbers
- Uses the Data Protection API (DPAPI)
- Password based encryption