User Account Management and Windows Firewall Configuration

Olaf Hubert Bykowski, bykowola@fit.cvut.cz

October 14, 2024

1 User Account Management

Setting up and managing user accounts is crucial for system security. Below is a concise guide emphasizing best practices for password security.

1.1 Creating and Managing User Accounts in Windows

User accounts can be added and removed using the **Settings** app [1].

1.1.1 Adding a User Account

1. **Open Settings:** Click on the *Start* menu and select the *Settings* app (gear icon).
2. **Navigate to Accounts:** In the Settings window, click on *Accounts*.
3. **Access Other Users:** Select *Other users* from the menu.
4. **Add a New User:** Under the *Add other user* section, click on *Add account*.
5. **Enter Account Information:**
   - If the person has a Microsoft account, enter their email address.
   - If they do not have a Microsoft account, you can create one using their email address.
   - To create a local account, select *I don't have this person's sign-in information* and then choose *Add a user without a Microsoft account*.
6. **Complete Setup:** Follow the on-screen instructions to finish setting up the account.

1.1.2 Removing a User Account

1. **Open Settings:** Click on the *Start* menu and select *Settings*.
2. **Navigate to Accounts:** Click on *Accounts*.
3. **Access Other Users:** Select *Other users*.
4. **Select Account to Remove:** In the menu, click on the account you wish to remove.
5. **Remove Account:** Click on *Remove* and confirm the action.

1.1.3 Best Practices for Passwords

- Use passwords at least 16 characters long.
- Include uppercase letters, lowercase letters, numbers, and special characters [2].
- Avoid common or easily guessable passwords.
- Change passwords regularly.
- Enforce password policies via Group Policy Editor.

1.2 PowerShell Script for User Account Creation

The following script automates user account creation while enforcing password policies, including checks for uppercase and lowercase letters.

Listing 1: create_user.ps1

```
# Check if the script is running with administrative privileges
$myWindowsID = [System.Security.Principal.WindowsIdentity]::GetCurrent()
$myWindowsPrincipal = New-Object System.Security.Principal.WindowsPrincipal($myWindowsID)
$adminRole = [System.Security.Principal.WindowsBuiltInRole]::Administrator

if (!$myWindowsPrincipal.IsInRole($adminRole)) {
    # Relaunch the script as administrator
    $newProcess = New-Object System.Diagnostics.ProcessStartInfo "PowerShell"
    $newProcess.Arguments = $myInvocation.MyCommand.Definition
    $newProcess.Verb = "runas"
    [System.Diagnostics.Process]::Start($newProcess)
    exit
}

# Get the username and password from the user
$username = Read-Host "What would you like your user to be named?"
$SecurePassword = Read-Host "What password would you like to give the user?" -AsSecureString

# Convert the SecureString to a plain text string
$BSTR = [System.Runtime.InteropServices.Marshal]::SecureStringToBSTR($SecurePassword)
$UnsecurePassword = [System.Runtime.InteropServices.Marshal]::PtrToStringAuto($BSTR)

# Password validation checks
if ($UnsecurePassword.Length -lt 16) {
    Write-Host "Error: The password needs to be at least 16 characters long." -BackgroundColor White -ForegroundColor Red
    [Runtime.InteropServices.Marshal]::ZeroFreeBSTR($BSTR)
    return
} elseif ($UnsecurePassword -notmatch ’[ˆa-zA-Z0-9]’) {
    Write-Host "Error: The password must contain at least one special character." -BackgroundColor White -ForegroundColor Red
    [Runtime.InteropServices.Marshal]::ZeroFreeBSTR($BSTR)
    return
} elseif ($UnsecurePassword -notmatch ’[A-Z]’) {
    Write-Host "Error: The password must contain at least one uppercase letter." -BackgroundColor White -ForegroundColor Red
    [Runtime.InteropServices.Marshal]::ZeroFreeBSTR($BSTR)
    return
} elseif ($UnsecurePassword -notmatch ’[a-z]’) {
    Write-Host "Error: The password must contain at least one lowercase letter." -BackgroundColor White -ForegroundColor Red
    [Runtime.InteropServices.Marshal]::ZeroFreeBSTR($BSTR)
    return
} elseif ($UnsecurePassword -notmatch ’[0-9]’) {
    Write-Host "Error: The password must contain at least one number." -BackgroundColor White -ForegroundColor Red
    [Runtime.InteropServices.Marshal]::ZeroFreeBSTR($BSTR)
    return
}
```

```

# Free the BSTR memory
[Runtime.InteropServices.Marshal]::ZeroFreeBSTR($BSTR)

# Create the user with the given parameters
New-LocalUser -Name $username -Password (ConvertTo-SecureString $SecurePassword -
    AsPlainText -Force)
Write-Host "User '$username' created successfully." -ForegroundColor Green

# Wait for user input before closing
$null = $Host.UI.RawUI.ReadKey("NoEcho,IncludeKeyDown")
```

Note: The script uses regular expressions for password validation [3] and includes a check for administrative privileges [4].

2 Windows Firewall

2.1 Importance in Network Security

Windows Firewall is a critical component in network security, acting as a barrier between your computer and potential threats from the internet. It monitors network traffic and decides whether to allow or block specific traffic based on defined security rules [5].

2.1.1 Key Functions and Benefits

- **Traffic Monitoring:** Keeps track of network communications and blocks unauthorized access.
- **Threat Prevention:** Protects against malware, viruses, and hacking attempts by filtering incoming traffic.
- **Application Control:** Allows or blocks programs from accessing network resources, reducing the risk of data breaches.
- **Customizable Rules:** Enables administrators to define specific rules tailored to their network's needs.

2.2 Configuring Windows Firewall for a Small Business Network

For a small business with internet-accessible email and web servers:

1. **Open Firewall Settings:** Access *Windows Defender Firewall* and select *Advanced settings*.
2. **Create Inbound Rules:**
    - **Web Server:** Allow TCP ports 80 (HTTP) and 443 (HTTPS).
    - **Email Server:** Allow TCP ports 25 (SMTP), 143 (IMAP), and 993 (IMAPS).
3. **Apply Rules:** Ensure rules apply to appropriate profiles (Domain, Private, Public).
4. **Test Configuration:** Verify access to services from an external network.

2.3 PowerShell Script for Firewall Rules

This script automates the addition of necessary firewall rules.

Listing 2: configure_firewall.ps1

```powershell
# Check if the script is running with administrative privileges
$myWindowsID = [System.Security.Principal.WindowsIdentity]::GetCurrent()
$myWindowsPrincipal = New-Object System.Security.Principal.WindowsPrincipal($myWindowsID)
$adminRole = [System.Security.Principal.WindowsBuiltInRole]::Administrator

if (!$myWindowsPrincipal.IsInRole($adminRole)) {
    # Relaunch the script as administrator
    $newProcess = New-Object System.Diagnostics.ProcessStartInfo "PowerShell"
    $newProcess.Arguments = $myInvocation.MyCommand.Definition
    $newProcess.Verb = "runas"
    [System.Diagnostics.Process]::Start($newProcess)
    exit
}

# Define ports and create rules
$ports = @(25, 80, 143, 443, 993)
foreach ($port in $ports) {
    New-NetFirewallRule -DisplayName "Allow Port $port" -Direction Inbound -LocalPort $port -Protocol TCP -Action Allow
    Write-Host "Rule added for port $port." -ForegroundColor Green
}
Write-Host "Firewall configuration complete." -ForegroundColor Green
```

References

[1] Microsoft Support, *Manage user accounts in Windows*, Microsoft, Available at: https://support.microsoft.com/en-us/windows/manage-user-accounts-in-windows-104dc19f-6430-4b49-6a2b-e4dbd1dcdf32

[2] Microsoft Support, *Create and use strong passwords*, Microsoft, Available at: https://support.microsoft.com/en-us/windows/create-and-use-strong-passwords-c5cebb49-8c53-4f5e-2bc4-fe357ca048eb

[3] Microsoft Docs, *about_Regular_Expressions*, Microsoft, Available at: https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_regular_expressions

[4] Server Fault, *Gaining administrator privileges in PowerShell*, Available at: https://serverfault.com/questions/11879/gaining-administrator-privileges-in-powershell

[5] Microsoft Docs, *Windows Firewall overview*, Microsoft, Available at: https://learn.microsoft.com/en-us/windows/security/operating-system-security/network-security/windows-firewall/