Автоматическое подписание SignTool в Windows 10

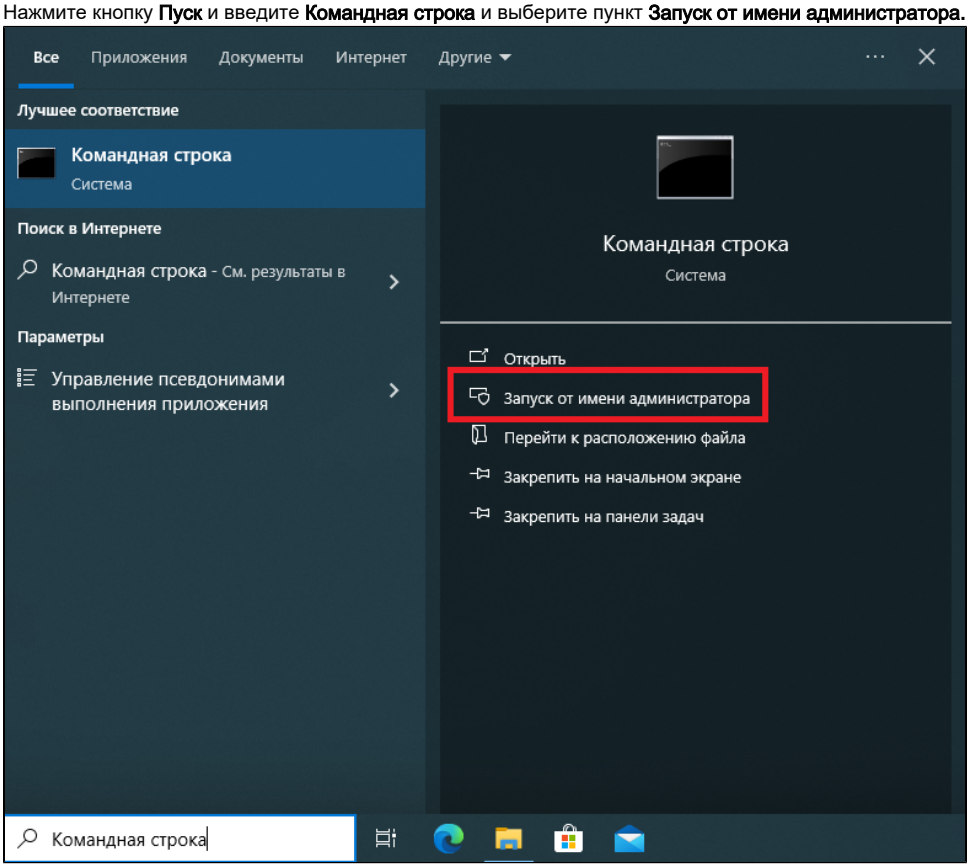
Для работы SignTool необходимо установить пакет разработчика Windows SDK: https://developer.microsoft.com/ru-ru/windows/downloads/sdk-archive/
Данная инструкция описывает сценарий автоматического подписания файлов с помощью SignTool и Рутокен ЭЦП. При настройке кэширования PIN кода, можно настроить автоматическое подписания без входа в сеанс пользователя Windows 10.
Проверка данного сценария проводилась на виртуальной машине Windows 10 с использованием Рутокен ЭЦП 3.0 3220. Проброс Рутокена на виртуальную машину не привёл к каким-либо ошибкам и трудностям.

- Импорт сертификата на Рутокен
- Изменение Криптопровайдера
- Регистрация сертификатов
- Настройка кеширования PIN кода
- Создание скрипта для подписания документов
- Добавление Скрипта в Службы Windows
- Настройка службы
- Подключение по SSH
- Итог

Импорт сертификата на Рутокен

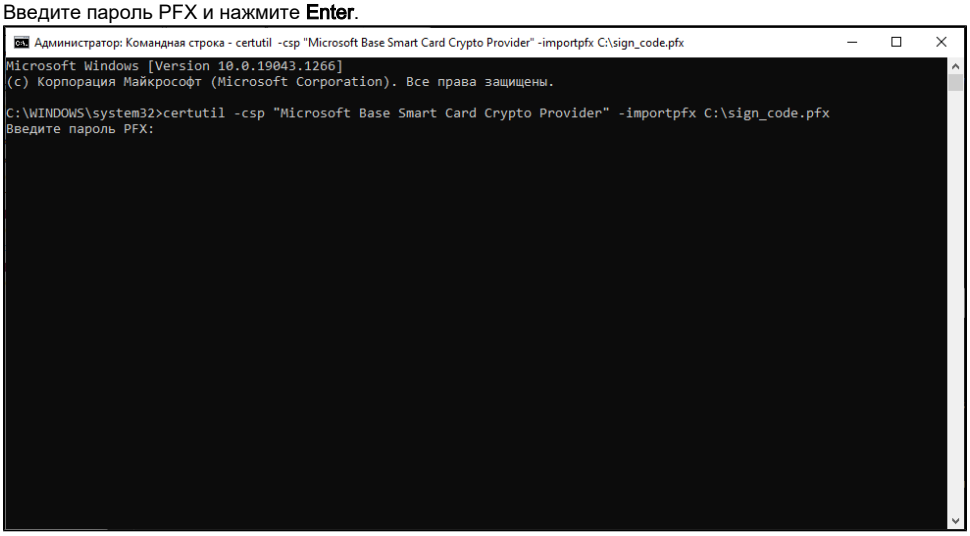
Для импорта сертификата на Рутокен необходимо выполнить следующие действия

1. Нажмите кнопку **Пуск** и введите **Командная строка** и выберите пункт **Запуск от имени администратора.**

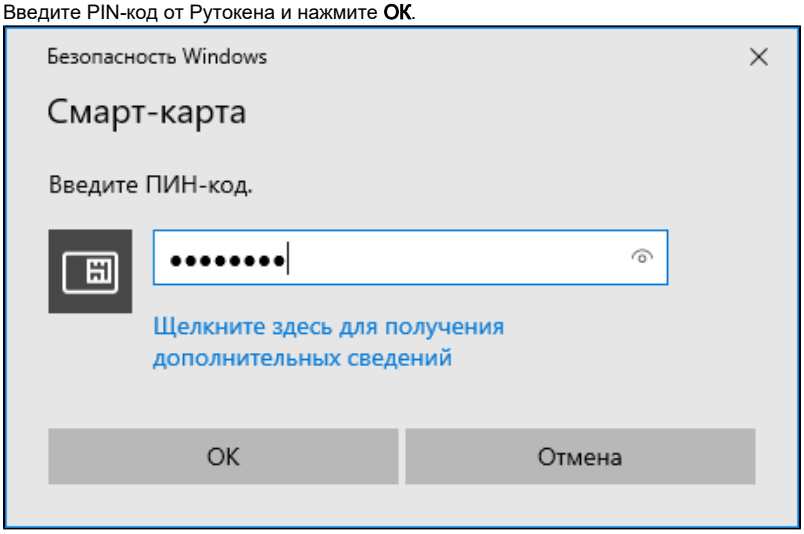


2. Если необходимо введите имя и пароль администратора компьютера.
3. Подключите Рутокен к компьютеру.
4. В командной строке наберите следующую команду: ***certutil -csp "Microsoft Base Smart Card Crypto Provider" -importpfx*** **C:\sign_code.pfx** и нажмите **Enter**.

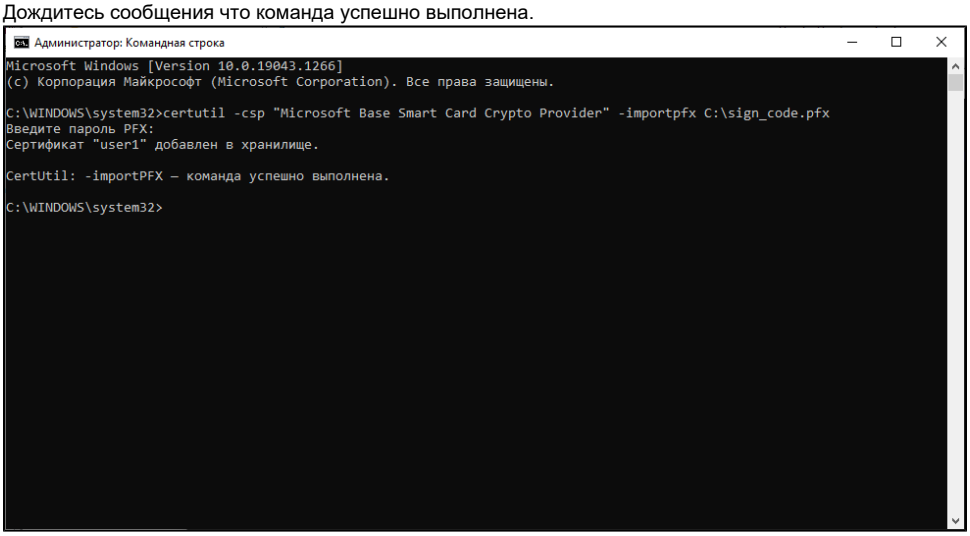
5. Введите пароль PFX и нажмите **Enter**.



6. Введите PIN-код от Рутокена и нажмите **OK**.



7. Дождитесь сообщения что команда успешно выполнена.

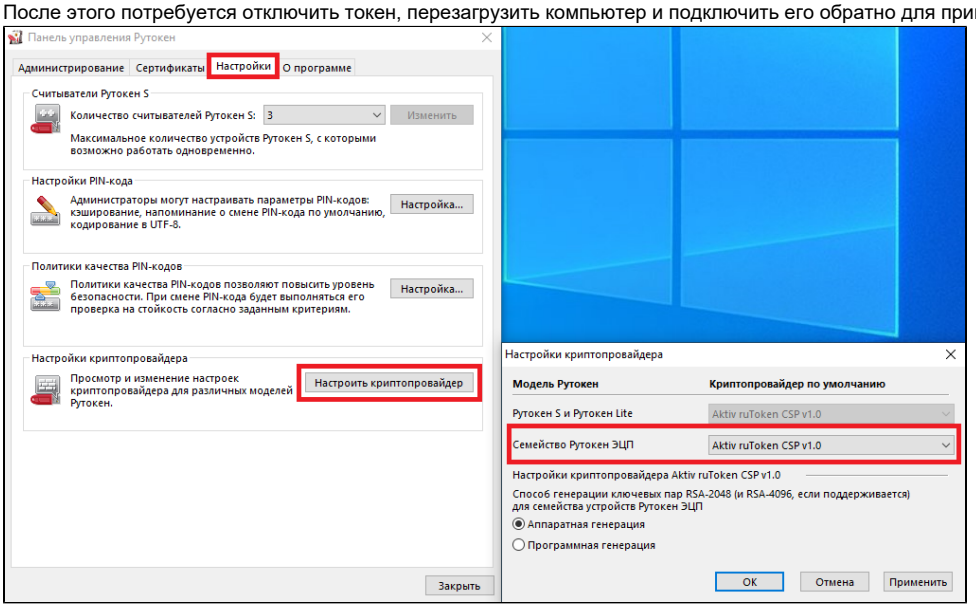


Изменение Криптопровайдера

Для корректной работы SignTool при закешированном PIN коде от Рутокена, необходимо изменить параметры криптопровайдера.

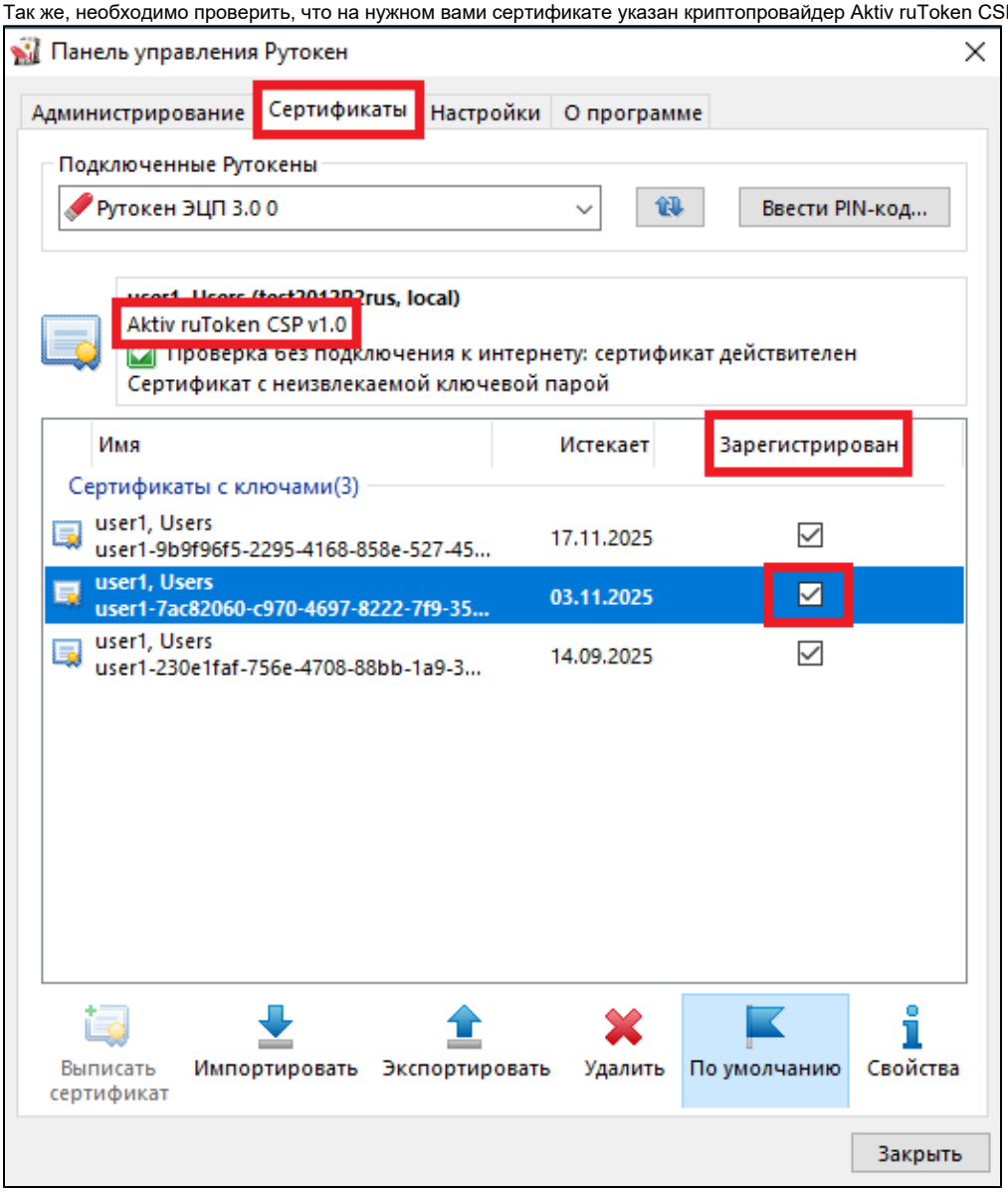
1. Зайти в Панель Управления Рутокен и перейти в раздел Настройки.

2. В разделе Настройки криптопровайдера необходимо нажать на "Настроить криптопровайдер".
3. В пункте "Семейство Рутокен ЭЦП", необходимо выбрать Aktiv ruToken CSP v1.0.
4. После этого потребуется отключить токен, перезагрузить компьютер и подключить его обратно для применения настроек.



Регистрация сертификатов

Для корректной работы SignTool, необходимо зарегистрировать сертификат.
Для этого в Панеле Управления Рутокен, в разделе Сертификаты необходимо поставить галочку напротив нужного сертификата для подписи.
Так же, необходимо проверить, что на нужном вами сертификате указан криптопровайдер Aktiv ruToken CSP v1.0.



Подробнее с работой в Панели Управления Рутокен можно ознакомиться в статье Начало работы с устройствами Рутокен.

Настройка кеширования PIN кода

Для включения кеширования пин-кода необходимо обратиться в службу нашей технической поддержки.

Создание скрипта для подписания документов

Ниже приведён пример скрипта Power Shell для подписания файлов помещённых в заранее созданную папку через SignTool.
Скрипт автоматически перемещает файл C:\test.exe в папку C:\sign и подписывает его.
После чего, скрипт переходит в статус ожидания. Далее можно вручную переносить файлы в папку sign, скрипт автоматически их подпишет.
Можно переносить и несколько файлов одновременно, скрипт подпишет их все по очереди. **Главное, чтобы в названии файла не было пробелов.**

Обратите внимание, что путь к signtool может отличаться в зависимости от версии установленного Windows SDK (в примере ниже, этот путь - signtoolPath = "C:\Program Files (x86)\Windows Kits\10\bin**10.0.19041.0**\x64\signtool.exe").
Так же, в самой команде подписания sign, необходимо указывать отпечаток вашего сертификата (в примере ниже, это - sign /debug /sha1 **f90be6d**

6ba25c388a384189ba5cd7975a3a04389 /v /td SHA256 $filePath).
Более подробно о том, как узнать отпечаток сертификата, можно прочитать в статье "Подпись файлов в Windows с помощью сертификата на Рутокен".

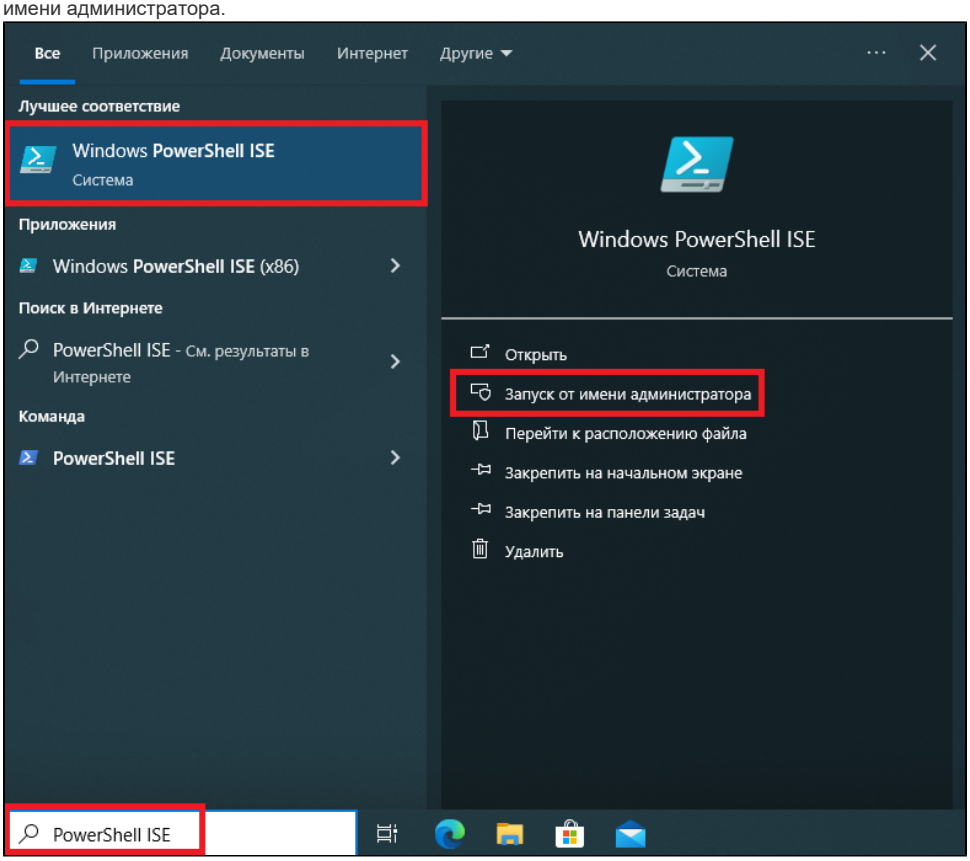
```
# Path and filter settings
$path = "C:\sign"
$filter = "*.*"



# Ensure the path exists
if (!(Test-Path $path)) {
    Write-Host "Path '$path' does not exist!"
    return
}



# The script block called when files are created
$action = {
    $signtoolPath = "C:\Program Files (x86)\Windows Kits\10\bin\10.0.19041.0\x64\signtool.exe"
    $filePath = $Event.SourceEventArgs.FullPath
    $arguments = "sign /debug /sha1 f90be6d6ba25c388a384189ba5cd7975a3a04389 /v /td SHA256 $filePath"
    Write-Host "Signing file '$filePath'"
    #Start-Process -FilePath $signtoolPath -ArgumentList $arguments -NoNewWindow
    Invoke-Expression "& '$signtoolPath' $arguments"
}



$sourceIdentifier = "FileCreated"



# Unregister the event if it is already registered
try {
    $existingEvent = Get-EventSubscriber -SourceIdentifier $sourceIdentifier -ErrorAction Stop
    if ($null -ne $existingEvent) {
        Unregister-Event -SourceIdentifier $sourceIdentifier
    }
}
catch {
    Write-Host "Event not found. Registering the event."
}



# Create the FileSystemWatcher
$fsw = New-Object IO.FileSystemWatcher $path, $filter
$fsw.EnableRaisingEvents = $true



$job = Register-ObjectEvent $fsw Created -SourceIdentifier $sourceIdentifier -Action $action



# Validate if the event is actually registered
if (Get-EventSubscriber | Where-Object { $_.SourceIdentifier -eq $sourceIdentifier })
{
    Write-Host "Event has been registered successfully."
}
else
{
    Write-Host "Failed to register event."
}
```

```
Write-Host "Script is now monitoring $path."



# Copy test.exe to the sign folder
$testExePath = "C:\test.exe"
if (Test-Path $testExePath) {
    Copy-Item $testExePath -Destination $path
    Write-Host "test.exe copied to $path."
} else {
    Write-Host "Could not find $testExePath."
}



# Prevent the console from closing immediately
do {
    Start-Sleep -Seconds 1
} while ($true)
```

Добавление Скрипта в Службы Windows

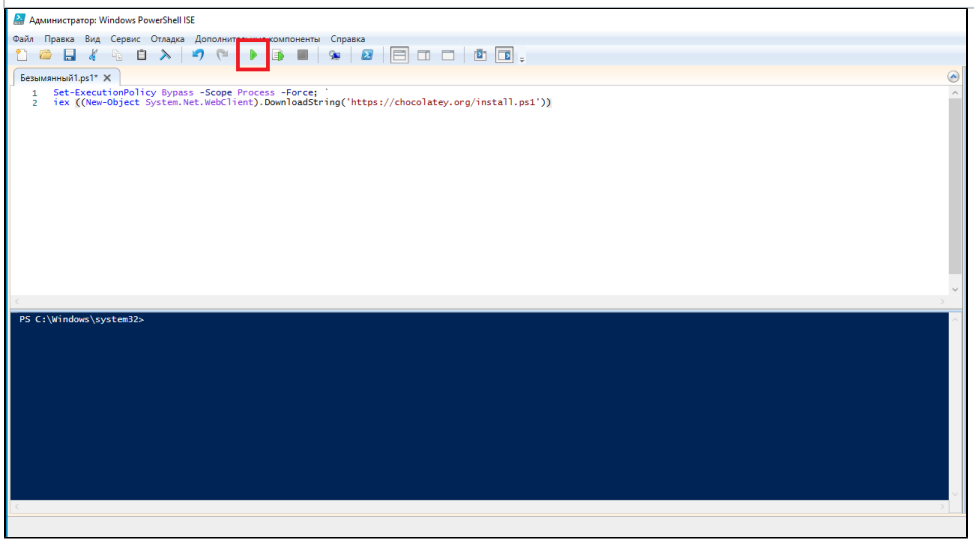
Для того, чтобы скрипт работал при старте ОС и подписывал файл до входа в учётную запись пользователя, необходимо запустить скрипт как службу Windows.
Для этого используем NSSM (Non-Sucking Service Manager) — это утилита, позволяющая устанавливать исполняемые файлы приложений в качестве служб в ОС семейства Microsoft Windows.

1. Для установки NSSM, необходимо запустить PowerShell. Для этого, откроем поиск и наберём PowerShell ISE. Запустить его нужно от имени администратора.

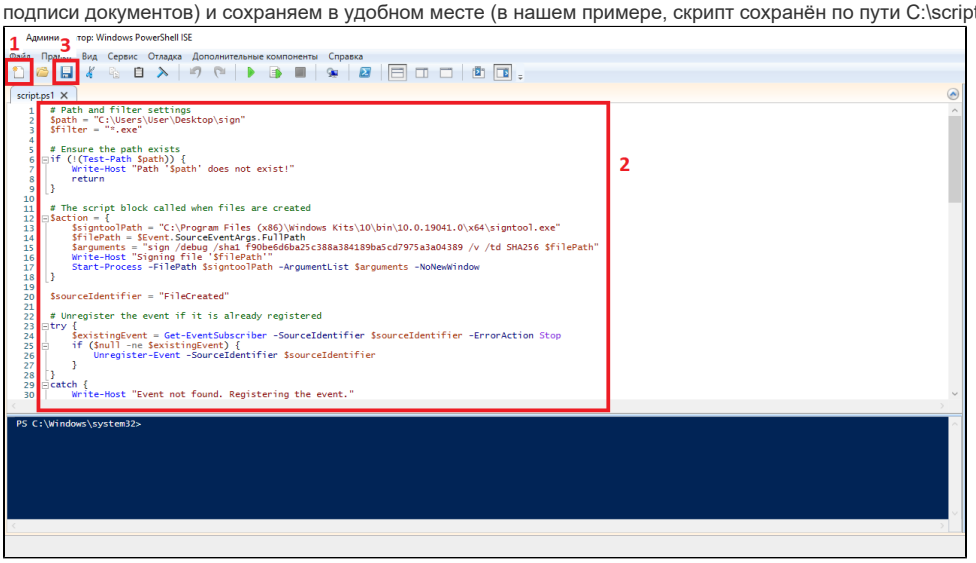


2. Создаём новый файл и запускаем команды для установки NSSM:

```
Set-ExecutionPolicy Bypass -Scope Process -Force;
iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))
choco install nssm
```



3. Создаём новый сценарий в PowerShell ISE, прописываем наш скрипт для подписания файлов (из раздела Создание скрипта для подписи документов) и сохраняем в удобном месте (в нашем примере, скрипт сохранён по пути C:\script.ps1)



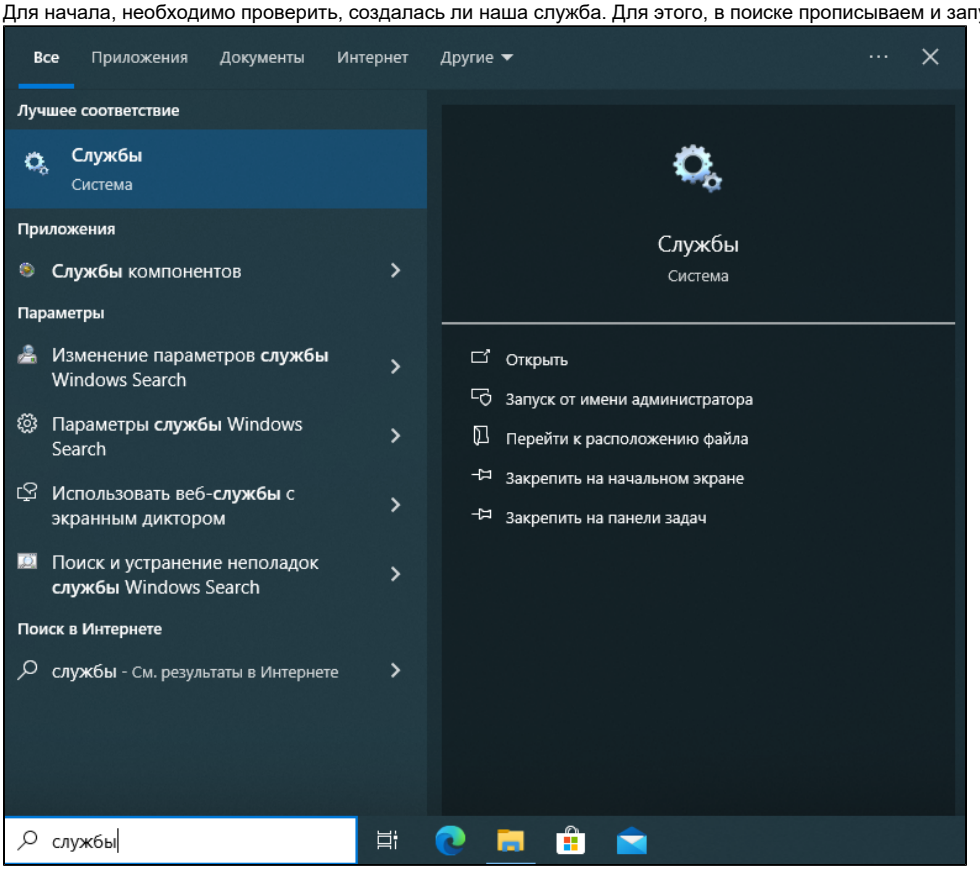
4. Далее, прописываем и запускаем команду для добавления нашего скрипта в службы Windows. В данном примере, служба будет называться "script".

```
$NSSMPath = (Get-Command "C:\ProgramData\chocolatey\bin\nssm.exe").Source
$NewServiceName = "script"
$PoShPath= (Get-Command powershell).Source
$PoShScriptPath = "C:\script.ps1"
$args = '-ExecutionPolicy Bypass -NoProfile -File "{0}"' -f $PoShScriptPath
& $NSSMPath install $NewServiceName $PoShPath $args
& $NSSMPath status $NewServiceName



Start-Service $NewServiceName
Get-Service $NewServiceName
```

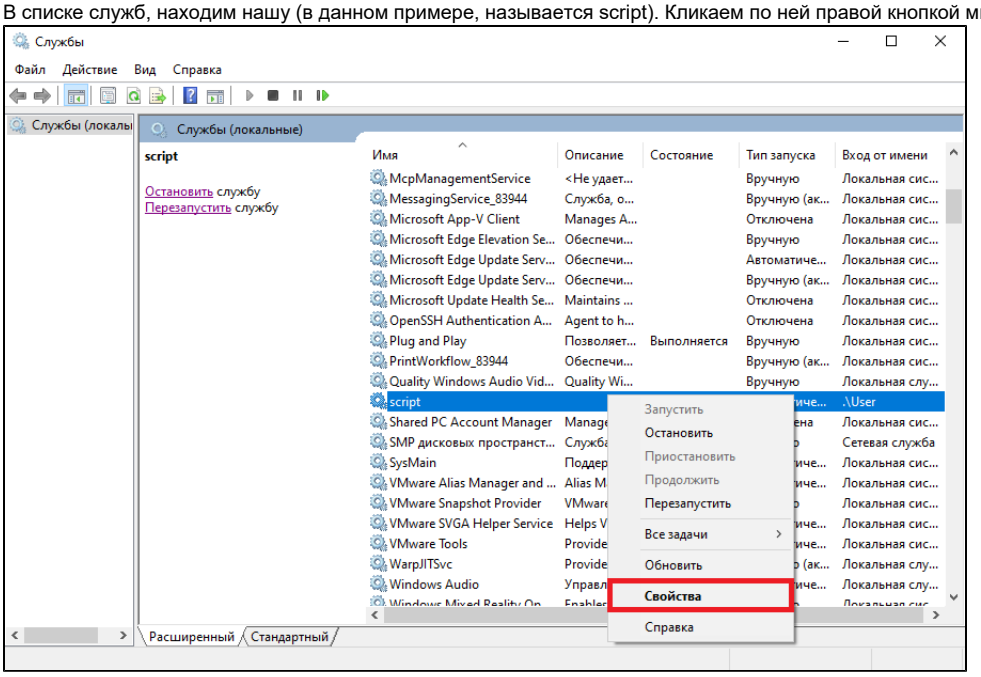
5. Если необходимо удалить службу, можно в командной строке использовать команду sc delete “Имя Службы”

Настройка службы

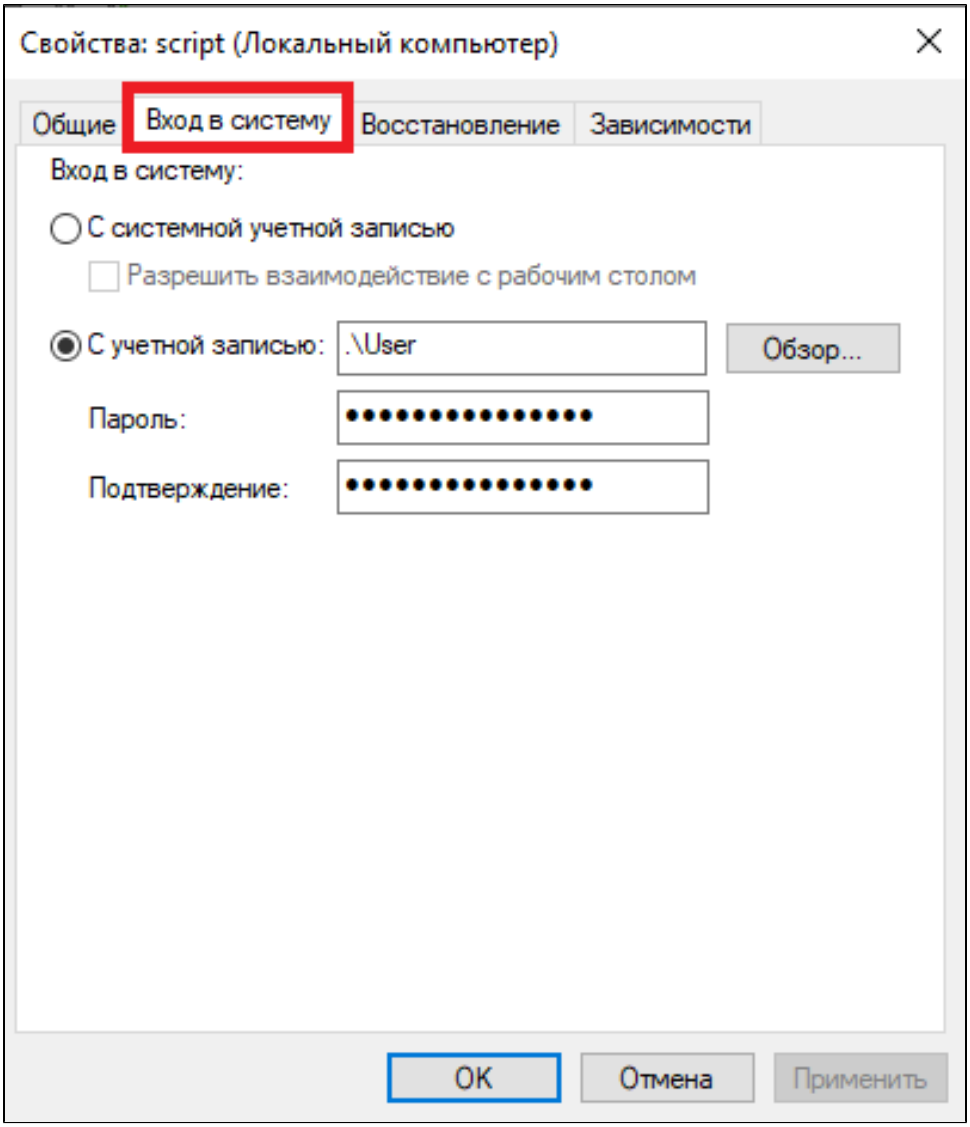
1. Для начала, необходимо проверить, создалась ли наша служба. Для этого, в поиске прописываем и запускаем Службы.



2. В списке служб, находим нашу (в данном примере, называется script). Кликаем по ней правой кнопкой мыши и переходим в Свойства.



3. В свойствах переходим во вкладку Вход в систему и указываем имя и пароль пользователя, от имени которого будет запускаться служба. Применяем и перезагружаем компьютер.



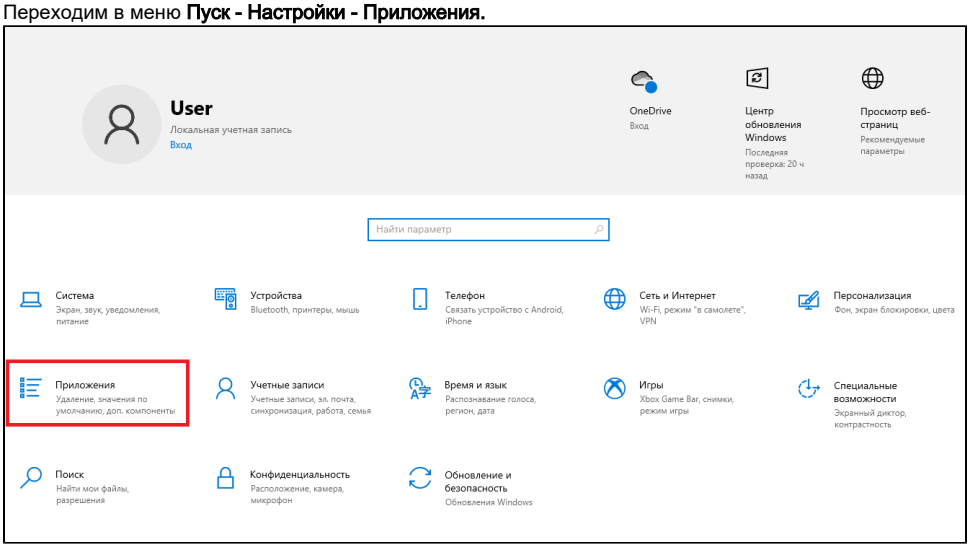
4. После настройки, службу необходимо перезапустить.

Подключение по SSH

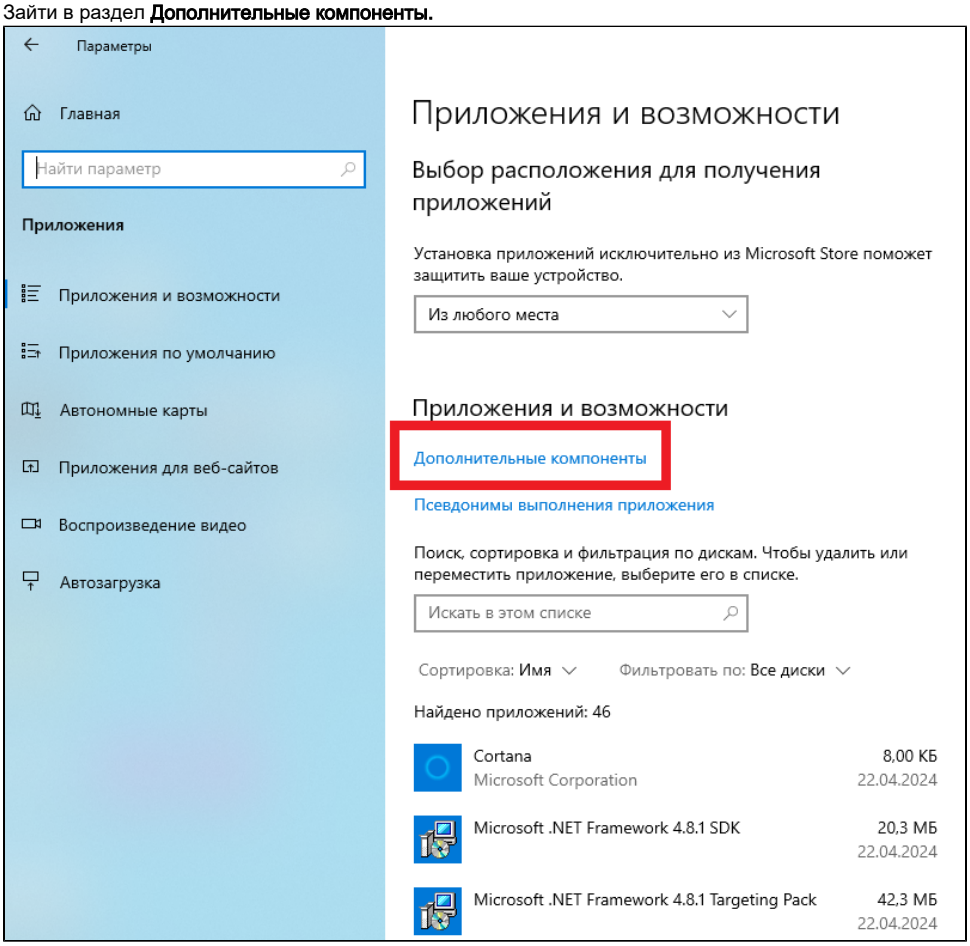
Подписание файлов работает даже при переносе файлов в папку sign удалённо, через ssh.

Для подключения по ssh необходимо установить компонент OpenSSH Server на одну из клиентских машин (данный компонент доступен к установке при версии Windows 10 старше 1809).

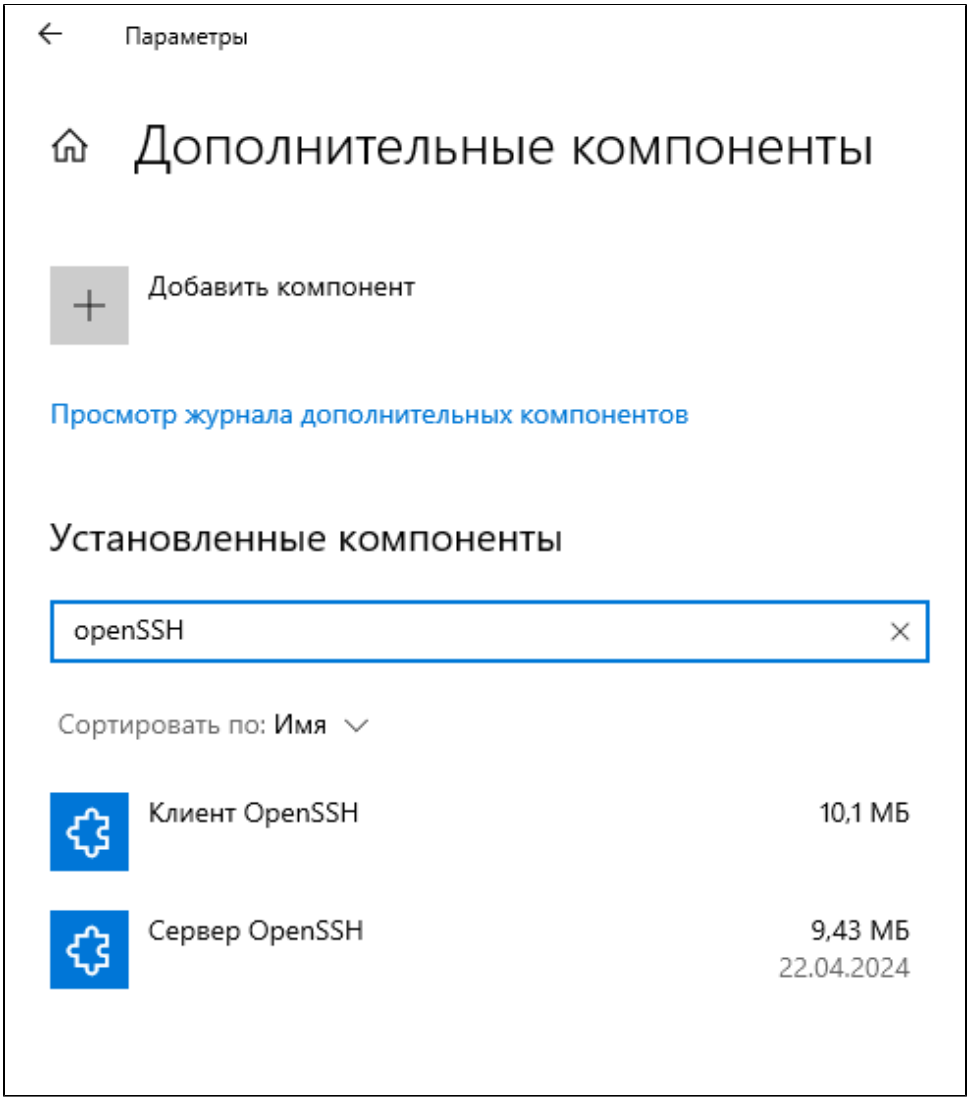
1. Переходим в меню **Пуск - Настройки - Приложения.**



2. Зайти в раздел **Дополнительные компоненты.**



3. Нажать **Добавить компонент** и ввести в поиске **Сервер OpenSSH**. Также убедитесь, что на обеих машинах установлен компонент **Клиент OpenSSH**.



4. Для подключения к серверу OpenSSH, необходимо в командной строке ввести команду:

```
ssh username@IPaddress
```

Где username - это имя пользователя сервера, а IPaddress - это IP адрес сервера OpenSSH.

Так как OpenSSH по умолчанию работает через порт 22, необходимо убедиться, что у клиента и у сервера есть разрешения на использование этого порта.

Итог

Если вы настроили кеширование PIN-кода и создали службу как указано в инструкции, то при подключенном Рутокене к машине, у вас будет подписываться указанный вами в скрипте файл до логина в учётную запись пользователя. Далее, служба будет ожидать следующие файлы для подписи в папке sign.