**hymmnos.c分析--功能9、10**

**一****. 监视特定网络包，开启后门（功能9）**

**1.模块功能原理**

这一部分的功能首先rootkit需要hook上netfilter协议栈，监听7777端口的网络数据包，当收到协议类型为icmp echo的数据包，同时检查数据包头中是否含有字符串“tonelico”，若有，则认定为需要攻击者需要开启后门，返回一个shell给攻击者。

**2.核心函数**

2.1 注册钩子函数并且判断指令数据包

magic\_packet\_hook(const struct nf\_hook\_ops \*ops, struct sk\_buff \*socket\_buffer, const struct net\_device \*in, const struct net\_device \*out, int (\*okfn)(struct sk\_buff \*))

该函数在netfilter协议栈上注册了钩子，监听网络数据包。对于进入的数据包，首先检测是否为icmp协议，，检查icmp头和数据段是否不为空，检查icmp协议的类型是否为echo，若不满足以上条件，则不是rootkit需要的指令包，将包继续向上发出。继续检查，取出数据包中包头数据段开头的内容，是否为字符串“tonelico”，若是则认定为该数据包为攻击者发出的指令包，需要rootkit开启后门反弹shell，在函数内进一步调用decode\_n\_spawn函数。

2.2 解析网络数据包

decode\_n\_spawn(const char \*data)

在magic\_packet\_hook函数中实际上是先将硬编码的TOKEN值进行异或以后直接和数据包中的数据进行比较，因此在验证成功以后还需要进一步解析数据。首先对数据段进行异或获取数据明文，按照约定的格式，以空格为分隔符获取ip和端口号，调用shell\_exec\_queue函数反弹一个shell。

2.3 反弹shell

shell\_exec\_queue(char \*path, char \*ip, char \*port)

shell\_execer(struct work\_struct \*work)

exec(char \*\*argv)

这三个函数的功能为反弹一个shell。首先接收参数，第一个参数为shell的地址，第二、三个参数分别为ip地址及端口号。创建一个系统任务，优先级为GFP\_KERNEL级别，初始化任务，将shell的地址、ip、端口传给任务，在shell\_execer中使用函数call\_usermodehelper(argv[0], argv, envp, UMH\_WAIT\_PROC)创建一个进程，这个函数能够异常方便地在内核中直接新建和运行用户空间程序，并且该程序具有root权限。其中第一个参数为要执行shell的路径，第二个为程序参数，第三个为环境变量参数，第四个UMH\_WAIT\_PROC使得其可以直接在中断的上下文中使用。

**3. 其他相关函数**

3.1 int atoi(char \*str)

将输入的字符串转化为数字，返回一个整型。

3.2 void s\_xor(char \*arg, int key, int nbytes)

将输入的字符串arg按位与输入的key进行异或。

3.3 void regist\_backdoor(void)

实现后门注册功能，将后门标志位更改为1，则调用magic\_packet\_hook函数注册钩子函数。

3.4 void unregist\_backdoor(void)

实现后门注销功能，将后门标志位更改为0，则同时取消对netfilter的hook。

**4.数据包构造文件**

攻击者需要构造指令数据包，具体的代码注释见文件send\_cmd.c。

**二.将当前进程权限改为root（功能10）**

分两种情况实现rootkit的提权。实现将rootkit权限更改为root级别的关键是将进程的uid、gid等设置为0。

当内核版本小于2.6.30时，手动设置当前进程的uid、suid、euid、gid、egid、fsuid、fsgid为0，再使用cap\_set\_full函数使当前进程获得最大的权能。

内核版本2.6.30之后弃用了cap\_set\_full函数，直接调用commit\_creds函数就可以获得root权限。