VPNFilter更新，加入7个新模块（下）

angel010 / 2018-09-28 20:54:26 / 浏览数 2373 安全工具 工具 顶(0) 踩(0)

本文翻译自：
https://blog.talosintelligence.com/2018/09/vpnfilter-part-3.html

---

https://xz.aliyun.com/t/2813 中提到VPNFilter更新过程中共加入7个新模块，本文继续分析其余4个模块。

netfilter（DOS工具）

Netfilter在命令行中也会有三个参数，但前二个参数是不用的，第三个参数是格式`<block/unblock> <# of minutes>`中引用的字符串。`<# of minutes>`是netfilter在退出前要执行的时间。如果`block`是第三个参数的第一部分，netfilter就会将下面的规则加入iptables：

```
Chain FORWARD (policy ACCEPT)
target     prot opt source         destination
DROP        tcp -- anywhere        anywhere       tcpflags: PSH/PSH
```

添加了规则后，netfilter会等30秒然后删除该规则。然后与`# of minutes`的时间进行比较，如果还有剩下的时间，该进程就再次执行。添加和删除的循环能确保event中一直有该规则。

一旦超时，该程序就退出。Signal handlers会在netfilter程序中安装，如果程序接收到SIGINT或SIGTERM，netfilter程序会删除iptables规则，然后退出。

最后，`unblock`参数可以删除之前用`block`参数添加的iptables规则。

Netfilter模块可能主要是用于限制对特定形式的加密应用的访问。

portforwarding

Portforwarding模块会执行下面的命令和参数：

```
portforwarding <unused> <unused> "start <IP1> <PORT1> <IP2> <PORT2>"
```

根据这些参数，portforwarding模块会通过安装下面的iptables规则来转发特定端口和IP的流量到另一个端口和IP：

```
iptables -t nat -I PREROUTING 1 -p tcp -m tcp -d <IP1> --dport <PORT1> -j DNAT --to-destination <IP2>:<PORT2>

iptables -t nat -I POSTROUTING 1 -p tcp -m tcp -d <IP2> --dport <PORT2> -j SNAT --to-source <device IP>
```

这些规则使通过受感染设备的到IP1: PORT1的流量被重定向到IP2: PORT2
。第二条规则会修改重定向的流量的源地址到受感染的设备来确保响应消息发送给受感染的设备。

在安装ipables规则前，portforwarding模块首先会创建一个到IP2 port2的socket连接来检查IP2是否可达。但socket关闭前也没有数据发送。
与其他操作iptables的模块类似，portforwarding模块会进入添加规则、等待、删除规则的循环以确保规则一直保留在设备中。

socks5proxy

socks5proxy模块是一个基于开源项目shadowsocks的SOCKS5代理服务器。服务器不使用认证，通过硬编码来监听TCP
5380端口。在服务器开启前，socks5proxy
fork会根据模块提供的参数连接到C2服务器。如果服务器不能在短时间（几秒）内响应，fork就会kill父进程然后退出。C2服务器会响应正常执行或中止的命令。

该模块含有下面的使用字符串，虽然与socks5proxy模块的参数不一致，但是这些设置不能通过命令行参数进行修改：

```
ssserver
   --username <username> username for auth
   --password <password> password for auth
   -p, --port <port> server port, default to 1080
   -d run in daemon
   --loglevel <level> log levels: fatal, error, warning, info, debug, trace
   -h, --help help
```

socks5proxy模块的真实命令行参数为：

```
./socks5proxy <unused> <unused> "start <C&C IP> <C&C port>"
```

socks5proxy模块会确认参数的个数大于1，但是如果有2个参数，其中一个是SIGSEV信号进程就会奔溃，说明恶意软件工具链在开发过程中有质量缺陷。

tcpvpn

tcpvpn模块是一个反向TCP（Reverse-TCP）VPN模块，允许远程攻击者访问已感染设备所在的内部网络。该模块与远程服务器通信，服务器可以创建类似TunTap之类的设
Strike这款渗透测试软件的VPN Pivoting功能。

所有数据都是RC4加密的，key是用硬编码的字节生成的。

```
"213B482A724B7C5F4D77532B45212D215E79433D794A54682E6B653A56796E457A2D7E3B3A2D513B6B515E775E2D7E533B51455A68365E6A67665F34527A7
```

与tcpvpn模块关联的命令行语法：

```
./tcpvpn <unused> <unused> "start <C&C IP> <C&C port>"
```

MikroTik

Winbox Protocol Dissector

研究人员在研究VPNFilter时，需要了解这些设备是如何被入侵的。在分析MikroTik设备时，研究人员发现了一个开放端口TCP
8291，配置工具Winbox用端口TCP 8291进行通信。

来自这些设备的流量多为二进制数据，因此我们无法在不使用协议解析器的情况下来分析该协议所能触及的访问路径。因此，研究人员决定自己开发协议解析器，协议解析器

比如，CVE-2018-14847允许攻击者执行目录遍历来进行非认证的凭证恢复。协议解析器在分析该漏洞中起了很大的作用。

Winbox协议

Winbox来源于MikroTik提供的Winbox客户端，用作Web GUI的替代方案。

官方文档称，Winbox是一个小工具，可以使用快速简单地通过GUI来管理MikroTik
RouterOS。这是一个原生的Win32程序，但也可以通过Wine运行在Linux以及MacOS上。所有的Winbox接口函数都尽可能与控制台函数耦合。但Winbox无法修改某些高

但Winbox协议并非官方名词，只是与官方客户端匹配，因此选择沿用该名词。

使用解析器

解析器安装起来非常简单，由于这是一个基于LUA的解析器，因此无需重新编译。只需要将`Winbox_Dissector.lua`文件放入`/$HOME/.wireshark/plugins`目录即可。
8291端口的所有流量。

来自客户端/服务器的单条消息解析起来更加方便，然而实际环境中总会遇到各种各样的情况。观察实时通信数据后，我们证实Winbox消息可以使用各种格式进行发送。

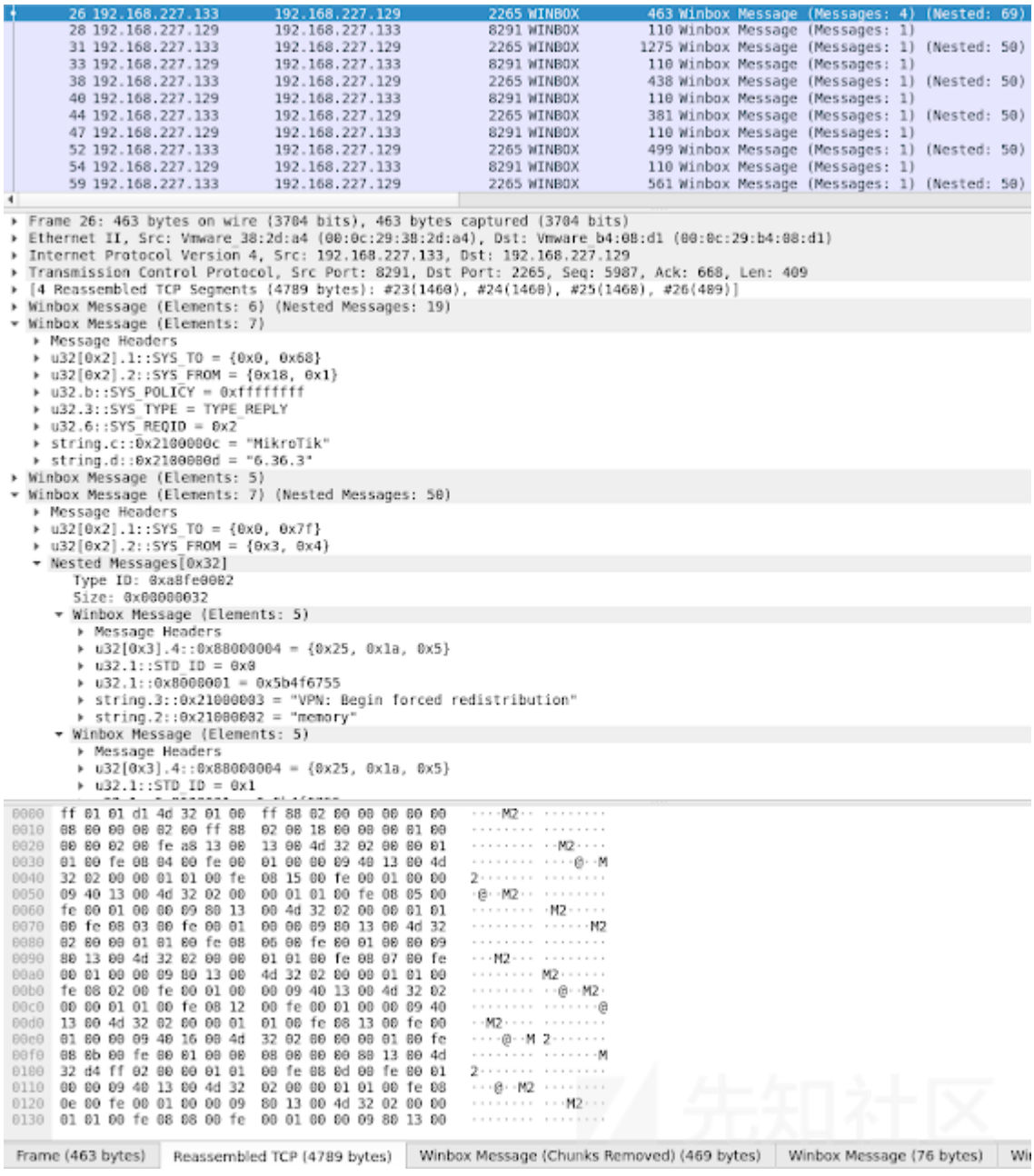
我们捕获过的Winbox通信数据具备各种属性，比如：

1. 在同一个报文中发送多条消息；
2. 消息中包含1个或多个2字节的“chunks”数据，我们在解析之前需要删除这些数据；
3. 消息过长，无法使用单个报文发送——出现TCP重组情况；
4. 包含其他“嵌套”消息的消息。

在安装解析器之前捕获得到数据包如下图所示：

```
0000  ff 01 01 d1 4d 32 01 00  ff 88 02 00 00 00 00 00   ····M2·· ········
0010  08 00 00 00 02 00 ff 88  02 00 18 00 00 00 01 00   ········ ········
0020  00 00 02 00 fe a8 13 00  13 00 4d 32 02 00 00 01   ········ ··M2····
0030  01 00 fe 08 04 00 fe 00  01 00 00 09 40 13 00 4d   ········ ····@··M
0040  32 02 00 00 01 01 00 fe  08 15 00 fe 00 01 00 00   2·······  ········
0050  09 40 13 00 4d 32 02 00  00 01 01 00 fe 08 05 00   ·@··M2·· ········
0060  fe 00 01 00 00 09 80 13  00 4d 32 02 00 00 01 01   ········ ·M2·····
0070  00 fe 08 03 00 fe 00 01  00 00 09 80 13 00 4d 32   ········ ······M2
0080  02 00 00 01 01 00 fe 08  06 00 fe 00 01 00 00 09   ········ ········
0090  80 13 00 4d 32 02 00 00  01 01 00 fe 08 07 00 fe   ···M2··· ········
00a0  00 01 00 00 09 80 13 00  4d 32 02 00 00 01 01 00   ········ M2·····
00b0  fe 08 02 00 fe 00 01 00  00 09 40 13 00 4d 32 02   ········ ··@··M2·
00c0  00 00 01 01 00 fe 08 12  00 fe 00 01 00 00 09 40   ········ ·······@
00d0  13 00 4d 32 02 00 00 01  01 00 fe 08 13 00 fe 00   ··M2···· ········
00e0  01 00 00 09 40 16 00 4d  32 02 00 00 00 01 00 fe   ····@··M 2·······
00f0  08 0b 00 fe 00 01 00 00  08 00 00 00 80 13 00 4d   ········ ·······M
0100  32 d4 ff 02 00 00 01 01  00 fe 08 0d 00 fe 00 01   2·······  ········
0110  00 00 09 40 13 00 4d 32  02 00 00 01 01 00 fe 08   ···@··M2 ········
0120  0e 00 fe 00 01 00 00 09  80 13 00 4d 32 02 00 00   ········ ···M2···
0130  01 01 00 fe 08 08 00 fe  00 01 00 00 09 80 13 00   ········ ········
```

安装Winbox协议解析器后，Wireshark可以正确地解析通信数据，如下图所示：



获取解析器

思科Talos团队开源了该工具，下载地址：https://github.com/Cisco-Talos/Winbox_Protocol_Dissector

点击收藏 | 0 关注 | 1

上一篇：攻击者是如何绕过Duo双因子身份验证的 下一篇：利用Java Security M...

1. 0 条回复
   - 动动手指，沙发就是你的了！