关于PLC安全的一次实验

等待未来66大顺 FreeBuf 1周前

一、引言

（1）随着工业 4.0 的高速发展，工业自动化程度越来越高，但工控设备暴露在公网的情况也越发明显。而其中尤其以PLC最为明显，这些PLC设备的来源多为国外厂商，安全变得不可控。所以如何检测针对PLC的攻击就显得极其重要。

（2）针对PLC有许多种攻击方式，低威胁的攻击可以监控PLC设备的状态，中高威胁的攻击可以控制PLC的启停，甚至可以对PLC进行PLC inject注入攻击。本实验需要实现对PLC的一次注入攻击，注入数据并不会对PLC造成损害，主要是了解注入的过程以及如何进行检测。

二、实验设备与环境

实验设备：两台通用PC或工控主机（本文运行环境为工控主机）实验环境：ubuntu操作系统、snort、ISF攻击框架、socat端口转发工具

三、相关技术介绍

3.1 ISF攻击框架

3.1.1 ISF介绍

ISF是一款针对工业控制系统的漏洞利用框架。该工具基于开源项目routersploit，采用Python语言开发，它与MetaSploit框架有些相似，使用此框架可以完成对PLC的多种攻击操作，它集成了很多对PLC进行攻击的脚本，通过傻瓜式的使用可以降低攻击的入门门槛，从而更加方便攻击者或安全工程师对PLC进行测试。

3.1.2 ISF攻击框架安装

为了方便大家使用，官方集成了butterfly网页终端(web terminal)，使我们可以从浏览器访问Linux系统的后台（类似ssh连接）。这样就使得一台isf攻击服务器，多人共同使用的场景成为可能，更大程度上方便了教学使用，也方便了大家的安装。ubuntu1、安装Docker，请确保网络连接状态完好

```
apt-get install -y docker.io
systemctl start docker
```

2、 创建docker镜像，以root用户运行如下命令：

```
mkdir -p /root/isfdocker
cd /root/isfdocker
wget https://github.com/w3h/isf/raw/master/docker/Dockerfile
docker build -t isf:v1 .
```

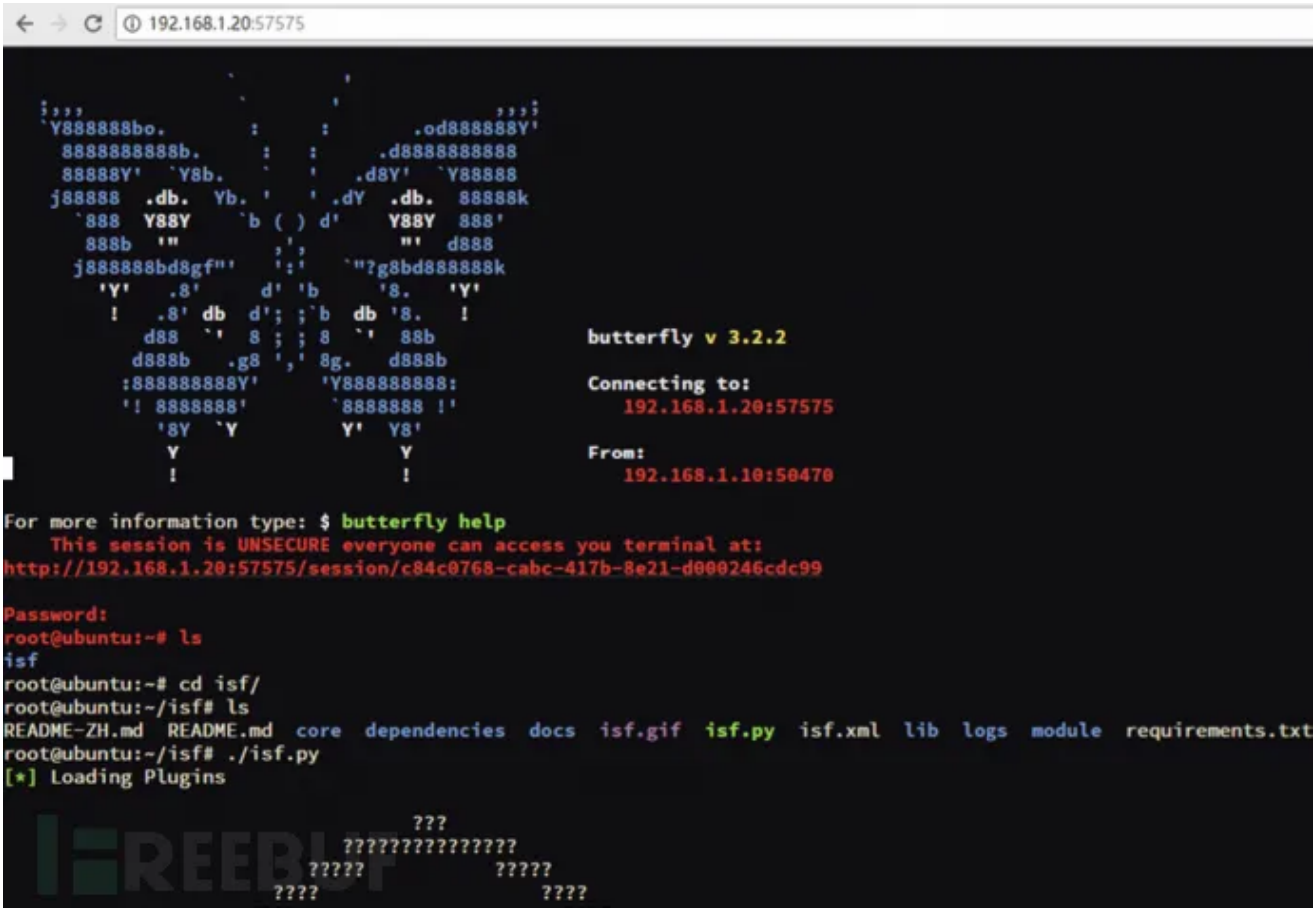
3、 请保持网络状态良好，等待最后一步运行完毕后，运行如下指令，就可以从浏览器打开。

```
docker run --net=host isf:v1 butterfly.server.py --host='0.0.0.0' --unsecure
```

在浏览器中输入：http://ip:57575注：ip地址为linux系统的ip地址，使用ifconfig就可查询。输入密码：123456，即可登录登录完毕后，执行以下指令运行isf

```
cd /root/isf/
./isf.py
```

运行效果如下：



3.2 PLC inject

3.2.1 介绍

PLC inject可以通过公网PLC访问到深层次的工业网络。可以实现的方法就是将PLC变成网关，这种方法在缺乏适当的防护功能的PLC上是可行的。技术娴熟的攻击者拥有某一个PLC的访问权限时，可以往上面上传或者下载代码，只要PLC设备支持对应的编码格式。而且代码被上传到PLC中后，就有很难被发现的特点，因为它不会中断程序的运行。当恶意代码被注入到PLC中后，会增加PLC中的代码量，如何我们定时观测原有代码和注入恶意代码后的程序，这两者的运行效果有明显的差异，然而其对生产过程的影响微乎其微。除非管理者主动监听从PLC中发出的恶意访问流量，否则很难在生产过程中发现。

3.2.2 PLC inject攻击过程

攻击者注入代码后，它会与PLC上的正常代码一起运行；对本地网络进行扫描，同时攻击者可以从PLC中下载扫描结果，之后在注入一个socks代理，攻击者可以通过通过充当代理的PLC访问本地网络内的所有PLC，这种攻击方式危害性极大，需要引起极大的重视。

3.3 Snort

3.3.1 简介

Snort是当前国际上非常著名的基于误用检测的网络入侵检测系统开放源码软件，采用规则匹配机制检测网络分组是否违反了事先配置的安全策略。安装在一台主机上就可以监测整个共享网段，一旦发现入侵和探测行为，具有将报警信息发送到系统日志、报警文件或控制台屏幕等多种实时报警方式。Snort不仅能够检测各种网络攻击，还具有网络分组采集、分析和日志记录功能。相对于昂贵与庞大的商用产品而言，Snort具有系统规模小、容易安装、容易配置、规则灵活和插件（plug-in）扩展等诸多优点。源代码和不同操作系统版本的可执行程序可从 www.snort.org网站免费下载。

3.3.2 组成

Snort主要由分组协议分析器、入侵检测引擎、日志记录和报警模块组成。协议分析器的任务就是对协议栈上的分组进行协议解析，以便提交给入侵检测引擎进行规则匹配。入侵检测引擎根据规则文件匹配分组特征，当分组特征满足检测规则时，触发指定的响应操作。日志记录将解析后的分组以文本或Tcpdump二进制格式记录到日志文件，文本格式便于分组分析，二进制格式提高记录速度。报警信息可以发送到系统日志；也可以采用文本或Tcpdump二进制格式发送到报警文件；也容许选择关闭报警操作。记录到报警文件的报警信息有完全和快速两种方式，完全报警记录分组首部所有字段信息和报警信息，而快速报警只记录分组首部部分字段信息。

3.3.3 安装

安装Snort Pre-Requisites输入如下命令，安装相关依赖环境

```
sudo apt-get install -y build-essential
sudo apt-get install -y libpcap-dev libpcre3-dev libdumbnet-dev
sudo apt-get install -y bison flex
```

创建目录，将snort相关文件保留在同一文件夹下

```
mkdir ~/snort_src
cd ~/snort_src
```

从Snort网站上下载和安装最新的DAQ

```
cd ~/snort_srcwget https://snort.org/downloads/snort/daq-2.0.6.tar.gztar -xvzf
```

当你运行./configure时，你应该可以看到如下的输出，这些数据表明了模块是可使用的

```
Build AFPacket DAQ module.. : yesBuild Dump DAQ module..... : yesBuild IPFW DA
```

安装Snort

```
sudo apt-get install -y zlib1g-dev liblzma-dev openssl libssl-devsudo apt-get i
```

运行如下命令，升级相关包

```
sudo ldconfigsudo ln -s /usr/local/bin/snort /usr/sbin/snort
```

3.3.4 Snort配置

在使用snort之前，需要根据保护网络环境和安全策略对snort进行配置，主要包括网络变量、预处理器、输出插件及规则集的配置，位于etc的snort配置文件snort.conf可用任意文本编辑器打开。除内部网络环境变量HOME_NET之外，在大多数情况下，可以使用snort.conf的默认配置。

由于我们不想使用root权限来运行snort，所以需要创建相关用户。同时也需要建立工作目录。运行命令如下：

```
# Create the snort user and group:sudo groupadd snortsudo useradd snort -r -s /
```

移动配置文件，运行命令如下

```
cd ~/snort_src/snort-2.9.9.0/etc/sudo cp *.conf* /etc/snortsudo cp *.map /etc/s
```

由于我们将在一个文件下面编写snort规则，所以注释掉单个rules文件

```
do sed -i "s/include \$RULE\_PATH/#include \$RULE\_PATH/" /etc/snort/snort.conf
```

使用编辑器修改配置文件，将HOME_NET更改为自己电脑所在的CIDR地址

```
sudo vi /etc/snort/snort.conf
```

打开后，更改地址

```
ipvar HOME_NET 10.0.0.0/24
```

从104行开始，设置如下文件路径

```
_LIST_PATH /etc/snort/rules/iplistsvar BLACK_LIST_PATH /etc/snort/rules/iplists
```

为了使测试snort变得更加容易，消除564行的注释，使其看起来像这样

```
include $RULE_PATH/local.rules
```

3.3.5 Snort的使用

打开位于/etc/snort/rules/local.rules 文件，就可以编写相应的规则，如下是一个规则示例

```
msg:"ICMP test detected"; GID:1; sid:10000001; rev:001; classtype:icmp- event;)
```

完成规则编写后，在console中输入

```
ocal/bin/snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
```

启动snort，开始检测

四、实验内容

4.1 原理介绍

要想实现注入攻击，需要将一台部署snort的计算机连接PLC，并设置PLC的反向代理，设置反向代理可用 socat TCP4-LISTEN:502,reuseaddr,fork TCP4:192.168.20.108:502，ip地址需要根据实际情况进行更改。在一台部署攻击框架

ISF的主机上，使用isf工具对PLC进行注入攻击，具体实现如下，在isf安装成功后，安装实验辅助知识的介绍运行攻击框架，然后再使用 use use Modbus*PLC*Injecter启动攻击脚本，并输入目的ip以及端口号；输入端口号完毕后会提示选择功能类型，选择方法1）后，设置起始地址为0并输入攻击文件路径，在这自己选择负载文件，输入文件路径，snort规则只需要匹配到对应的负载数据即可检测到攻击。

检测示例规则如下：alert tcp any any -> $any 502 (msg:"plcinject"; content:"|d0 9d 00 00 00 06 01 03 00 80 00 01|"; sid:001111111; GID:001; priority:0;)。content内的值可根据PLC inject注入的数据进行更改，规则也可写多条。

4.2 实验步骤

1.将PLC设备插入到服务器eth1网口中，并ping 192.168.20.108，查看是否可到达，若可达就进行下一步；

2.在Terminal里输入docker run -p 502:502 -v /data/snort:/var/log/snort -it 10.10.2.82/ids:v6.0 /bin/bash，启动docker镜像，并输入ids；

3.在另一台可以访问服务器的电脑中进入Terminal，使用cd命令打开isf的工作目录；

4.在Terminal输入sudo python2 isf.py运行框架并根据提示输入用户密码；

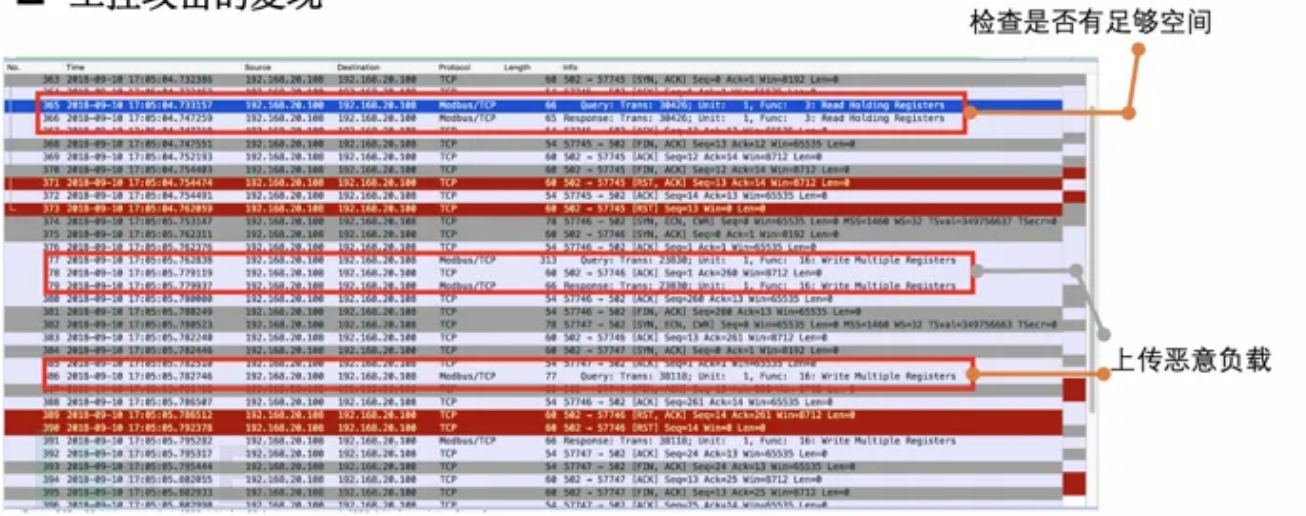
5.在运行框架成功后，使用use Modbus*PLC*Injecter，并输入目标ip地址以及端口号；

6.输入端口号完毕后会提示选择功能类型，选择方法1）后，设置起始地址为0并输入攻击文件路径/isf/module/touches/plcinjector/payload.txt,路径名根据攻击文件的路径自行修改；

7.选择执行，开始注入，效果如下图

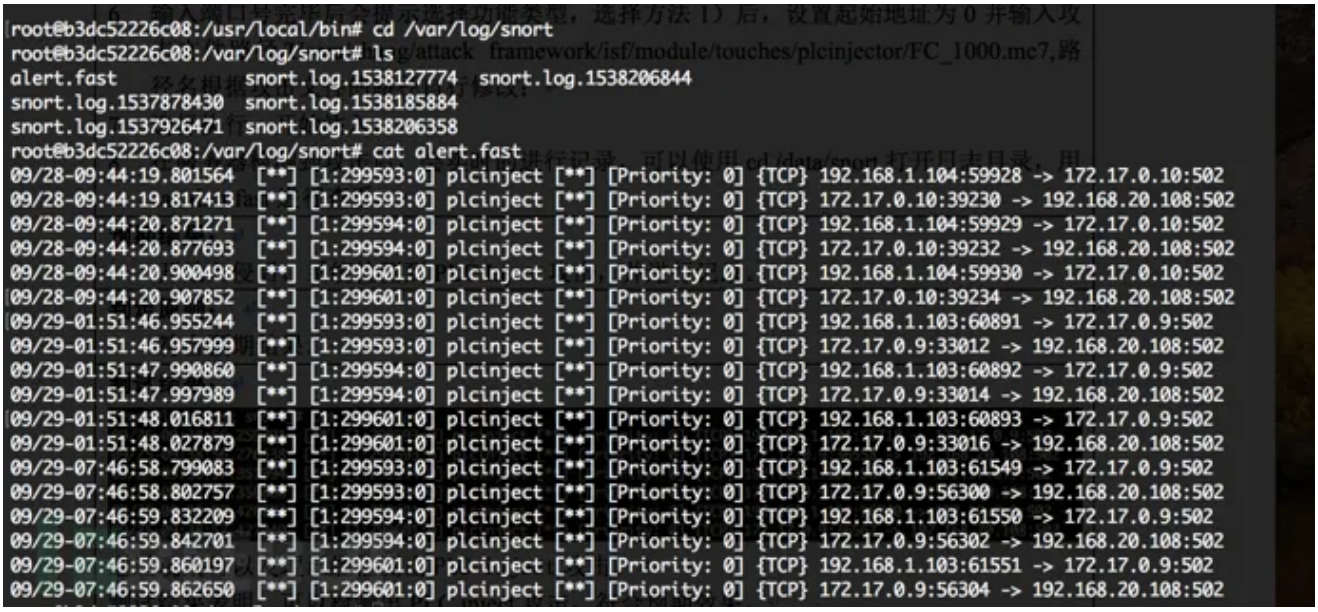
PLC注入攻击关键步骤

工控攻击的复现



8.在服务器检测到攻击后，会实时记录，可使用cd /data/data/snort打开日志记录，用cat alert.fast进行查看。

具体结果如下图



注：由于环境配置的不同，上述步骤可能有细微区别，本文直接将相关环境封装到docker内。详细部署步骤请参考第三大节内容，疏漏之处请谅解！

五、防御建议

由于PLC inject攻击利用了工控协议缺乏安全验证机制的漏洞，可以很轻易的实现攻击。为此可采用IDS、IPS等技术提高对特定攻击的检测防御能力。具体可采用以下措施

> 1）减少攻击路径；在工作环境内，可将设备使用防火墙进行单向隔断，阻断攻击路径，避免将设备接入公网
>
> 2）部署蜜罐；利用蜜罐对攻击进行诱捕，即采用假目标吸引攻击，从而保护真实设备，
>
> 3）尽可能更新系统；在设备空闲时段应尽量更新系统固件，弥补漏洞，降低被攻击风险，
>
> 4）尽量采用S7 comm plus等专业私有协议设备；由于私有协议暂时没被破解，可利用这一优势，增加攻击成本，
>
> 5）对上层网络进行安全防护，采用防火墙禁止恶意IP访问，采用IDS检测已进来的攻击，采用杀软检测本地PC已有的病毒，防止攻陷上层网络，进入业务环境，攻击设备

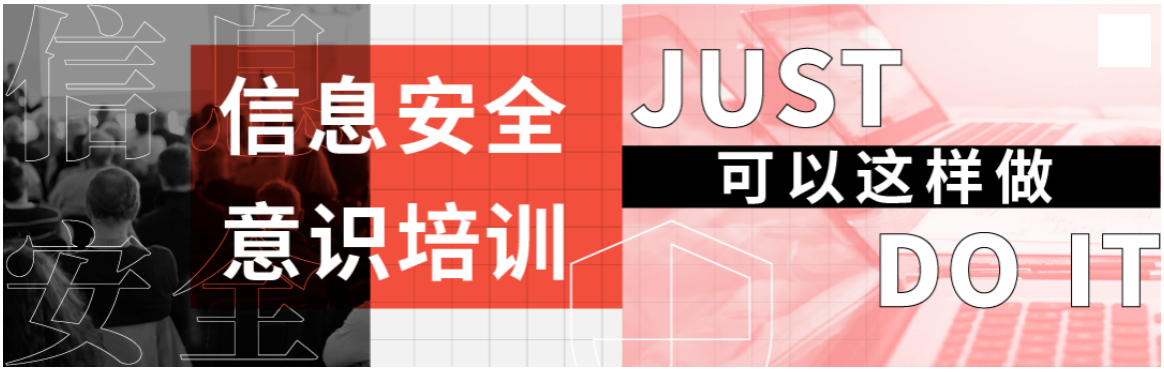
六、总结

本文主要对一次modbus PLC的注入攻击进行了记录，由于PLC本身缺乏有效的安全校验机制，导致其无法有效检测传入信息的合法性，使得攻击者采用合法的通信方式就可以完成

攻击。相对于modbus PLC，使用私有协议S7的PLC增加了一些安全验证机制，但依然存在安全问题。

*本文作者：等待未来66大顺，转载请注明来自FreeBuf.COM

精彩推荐



阅读原文