OWASP

Open Web Application Security Project

从CTF到工控安全

--孙鑫斌

OWASP
Open Web Application Security Project

1、工业的发展历程

2、工控安全的基础知识

3、工控比赛中遇到的典型问题

4、比赛和实际结合的一些经验体会

5、对未来的一些感悟

OWASP
Open Web Application Security Project

工业1.0

时间：18世纪60年代至19时间中期（大概是1760年至1850年）。

特点：在工业1.0中，我们都知道，瓦特改良了蒸汽机，从而开创了以机器代替人工的工业浪潮。工业1.0使用的机器都是以蒸汽或者水力作为动力驱动，虽然效率并不高，但是因为首次用机器代替人工，因此具有非常重要的划时代的意义。

关键词：蒸汽和水力驱动，效率低，功能有限，机器首次代替手工。

工业安全：由于没有互联网，这时候的工业安全都为蒸汽机人为破坏，零件老化等物理安全。

OWASP
Open Web Application Security Project

工业2.0

时间：19世纪70年代至20世纪初（大概是1870年到1900年）

特点：工业1.0的机械功能单一，驱动效率低，显然满足不了刚刚尝到工业革命甜头的人类。工业2.0的机器和动力相较于工业1.0，有着非常巨大的进步。并且，得益于电话机的发展，人类之间的通讯变得简单快捷，信息在人类之间的传播为第三次工业革命奠定了基础。

关键词：电力驱动，内燃机，飞机，电话信息交流，设备多样化。

工业安全：开始出现信息通信传输了，这时候的工业安全开始出现信息截断、密码破解、伪造信息，社会工程等安全攻击的雏形。

OWASP
Open Web Application Security Project

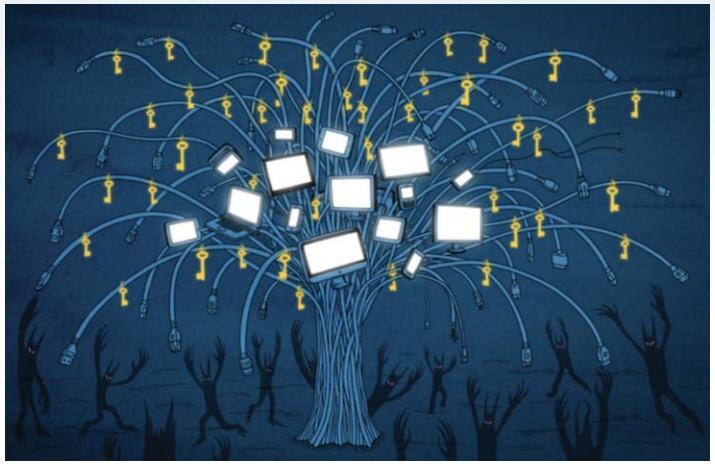
工业3.0

时间：二十世纪五十年代至今（大概是1950年到今天）

特点：工业3.0相对于工业2.0发生了更加巨大的变化。第三次工业革命不再局限于简单机械，原子能、航天技术、电子计算机、人工材料、遗传工程等具有高度科技含量的产品和技术得到了日益精进的发展。以互联网为信息技术的发展和应用几乎把地球上的每个人都联系了起来。工业中的生产出现了各种各样的机器人，以往那些高危、复杂、枯燥的工序都可以使用机器人代替，并且得到更大的经济效益。人类在这个时代的“野心”不再局限于放眼所及的地球，而是星辰大海，并且在航天技术的高速发展下得到了实现。

关键词：高新技术，机器人，自动化控制普及。

工业安全：产品漏洞、协议漏洞、载体服务器漏洞、人为因素等。



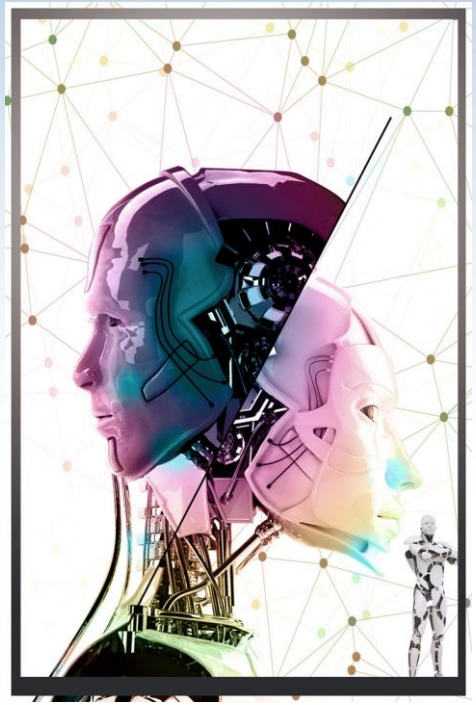
工业4.0

时间：当下-未来

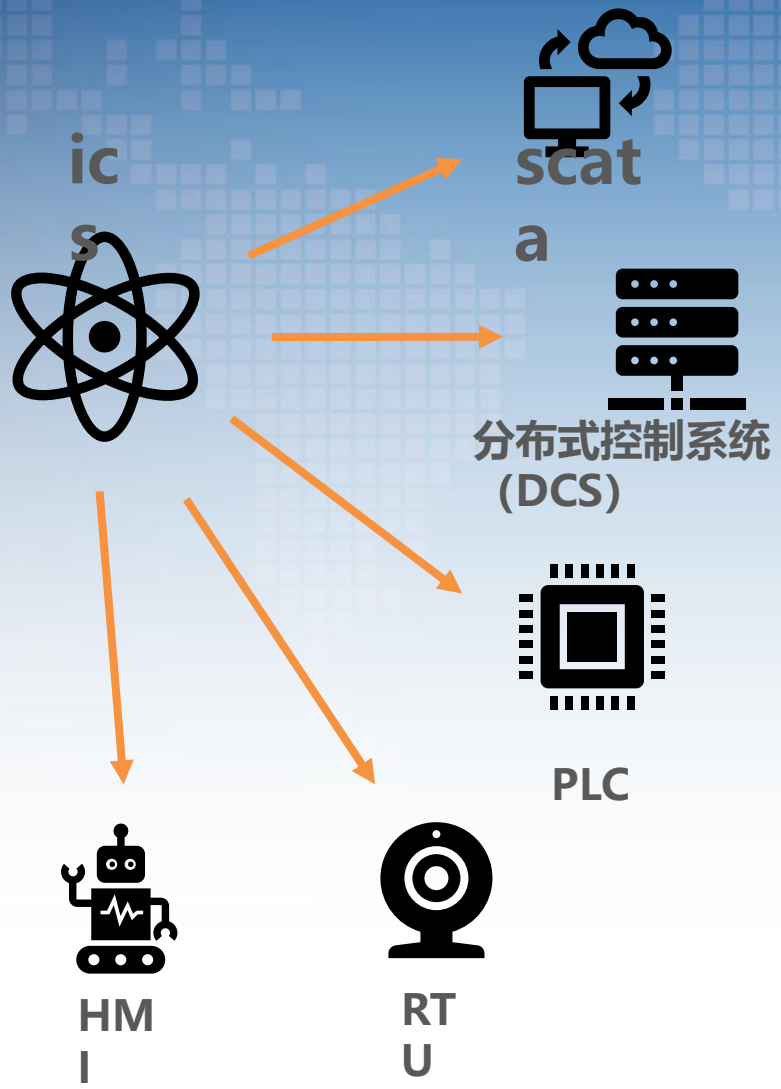
特点：工业4.0旨在将一切的人、事、物都连接起来，形成“万物互联” 。

关键词：物联网，大数据，无人工厂，高度智能。

工业安全：局域网、向传统互联网安全贴近，并一同发展。



工业安全的基础知识



工业控制系统（ISC):

是由各种自动化控制组件以及对实时数据进行采集、监测的过程控制组件共同构成的确保工业基础设施自动化运行、过程控制与监控的业务流程管控系统。

数据采集与监控系统(SCADA)

分布式控制系统(DCS)

可编程控制器(Programmable Logic Controller,PLC)

远程终端(Remote Terminal Unit,RTU)

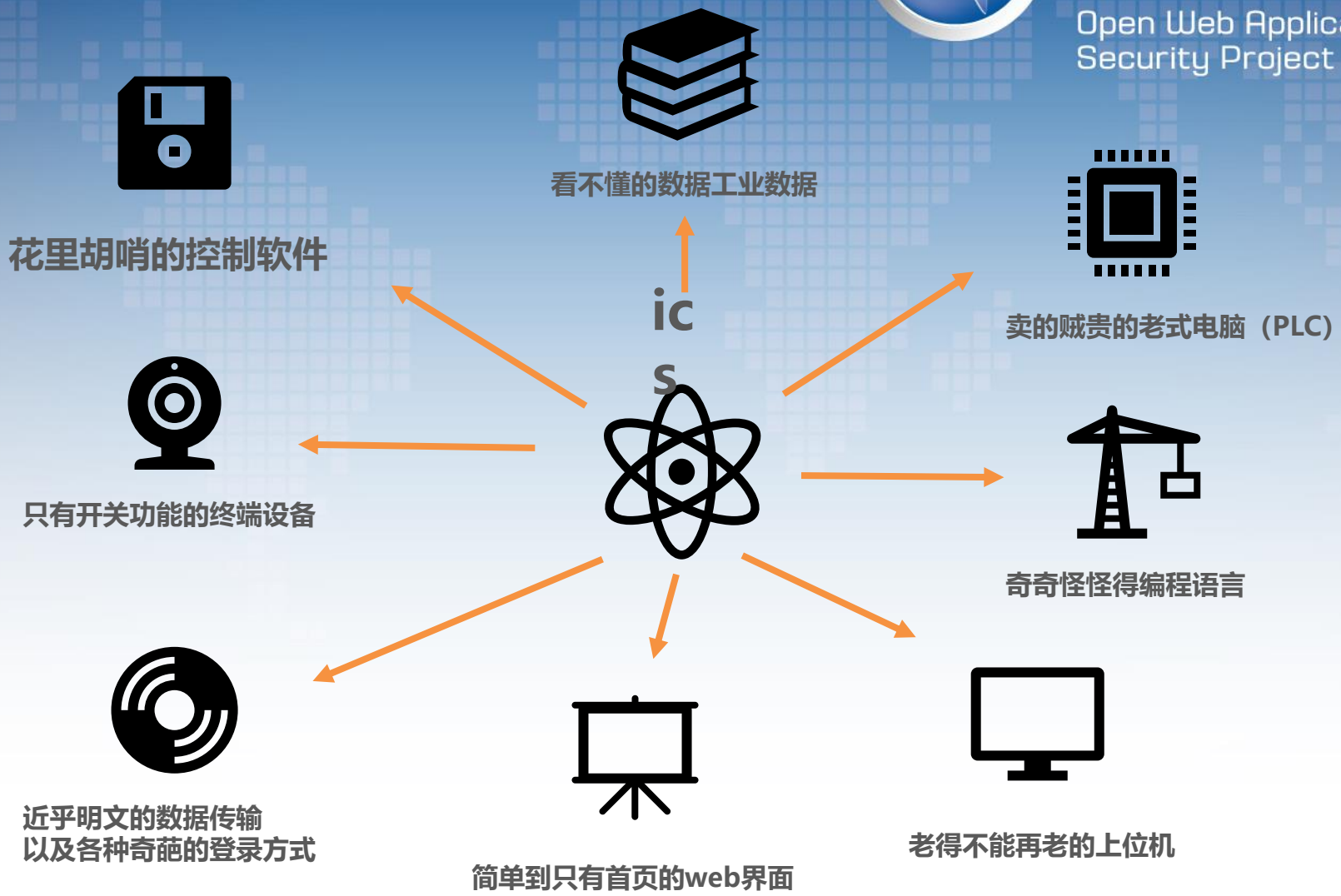
人机交互界面设备(Human Machine Interface,HMI)

以及各个设备之间的接口通讯技术。

然而在我眼里的工控安全是这样的

工业安全的基础知识

OWASP
Open Web Application Security Project



那作为一个web手，怎么玩工控

工控比赛中遇到的典型问题

2017年第一次参加工控比赛

- 2017年工...大赛报名.doc
- 20171221
- 飞秋FeiQ.exe.重命名
- 工控漏洞挖...使用手册.pdf
- 漏洞挖掘.txt
- 施耐德140...检测报告.rar
- bootp-dhcp...er_2.3.2.zip
- dos.zip
- Everything-...67b.x64.zip
- exploit_ics.tar.gz
- jd-gui-1.4.0.jar
- Mod_RSsim.msi
- MODBUS_R...D_TOOL.py
- modFuzzer.py
- Modscan32_4205.zip
- modscan32.zip
- modscan64.zip
- npp.7.3.2.Installer.exe
- NuDesign_..._v10.2.0.exe
- plcscan.rar
- qtester104-v1.18.zip
- s7-enumerate.nse
- s7clientdemo.rar
- s71200-en...rate-old.nse
- SCADAPASS-ftp密码.zip
- SCADAPASS-master
- SchneiderTFE.zip
- SnmpWalk.zip
- SuperScan.zip
- Wireshark-...4-2.4.3.exe
- xrdp-master.zip

- 【工控安全...的漏洞_files
- 【工控安全...的漏洞.html
- 工控安全 | ...（上）_files
- 工控安全 | ...（上）.html
- 工控安全 | ...（下）_files
- 工控安全 | ...（下）.html
- 工控安全科...统网络_files
- 工控安全科...统网络.html
- 工控系统威...弱性分析.pdf
- 工业控制系...全调查_files
- 工业控制系...全调查.html
- 漏洞分析_Vx...围统计_files
- 漏洞分析_Vx...围统计.html
- 速看！SCAD...现状！_files
- 速看！SCAD...现状！.html
- SCADA渗透测试_files
- SCADA渗透测试.html

- asia-16-Sp...PLC-wp.pdf
- asia-17-Kr...Systems.pdf
- Blackhat_工...备资料.rar
- eu-17-Atch...etworks.pdf
- eu-17-Lei-...Plus-wp.pdf
- us-16-FitzP...nts-wp.pdf
- us-16-Melr...Security.pdf
- us-17-Beer...vement.pdf
- us-17-Bran...eed-wp.pdf
- us-17-Nich...tion-wp.pdf
- us-17-Quar...ots-wp.pdf
- us-17-Stag...etworks.pdf

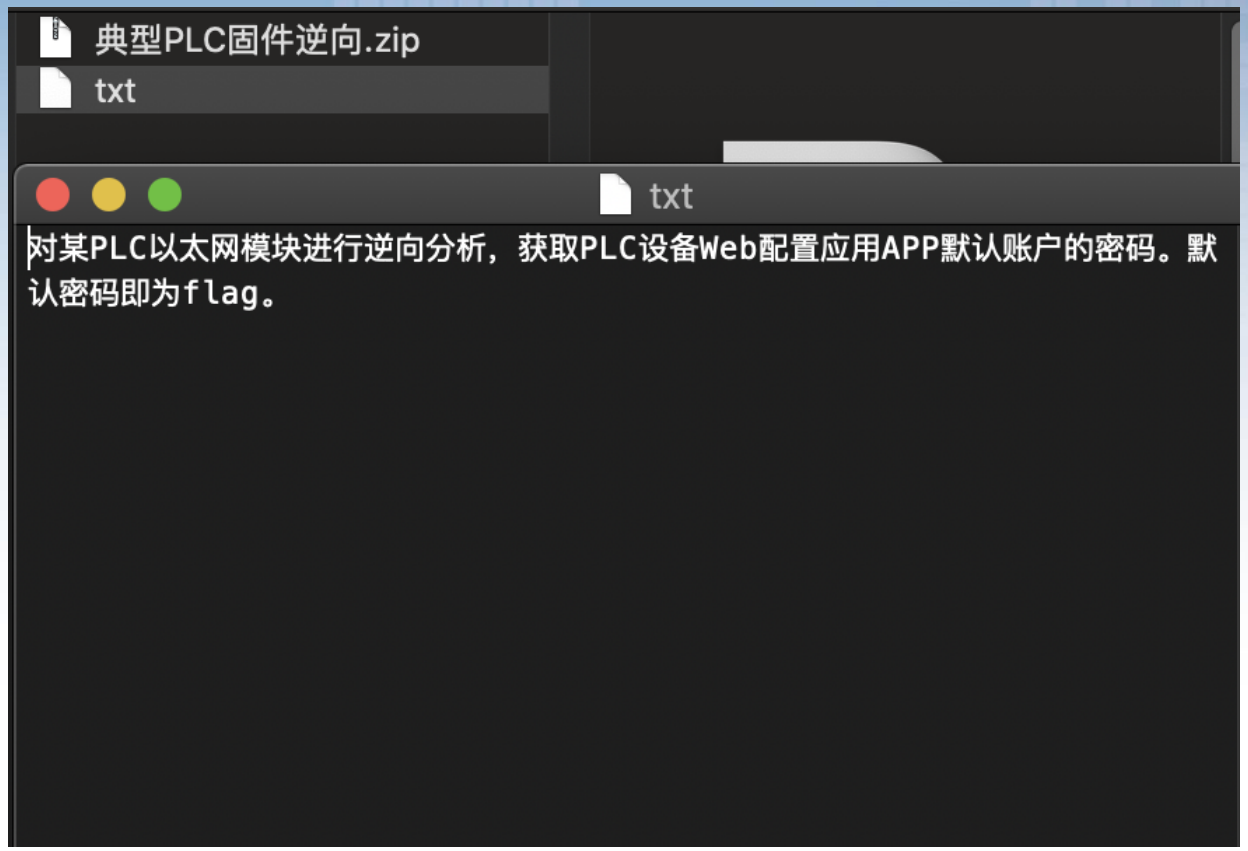
- 典型PLC固件逆向
- 电力系统协议分析
- 恶意软件逆向分析
- 恶意软件样本分析
- 工控设备固件分析
- 工业协议分析3
- 工业协议分析4
- 工业APP安全分析
- 梯形图运算1
- 梯形图运算2
- 组态工程密码破解
- PLC通讯协议分析

参考资料：

http://plcscan.org 哈工大的灯塔实验室
http://www.autop.com.cn/
http://www.gongkong.com/safety1/
http://www.cechina.cn/
https://ics-cert.us-cert.gov/
http://www.cnvd.org.cn/
http://www.wooyun.org/
https://www.defcon.org/
https://www.blackhat.com/
……
等等

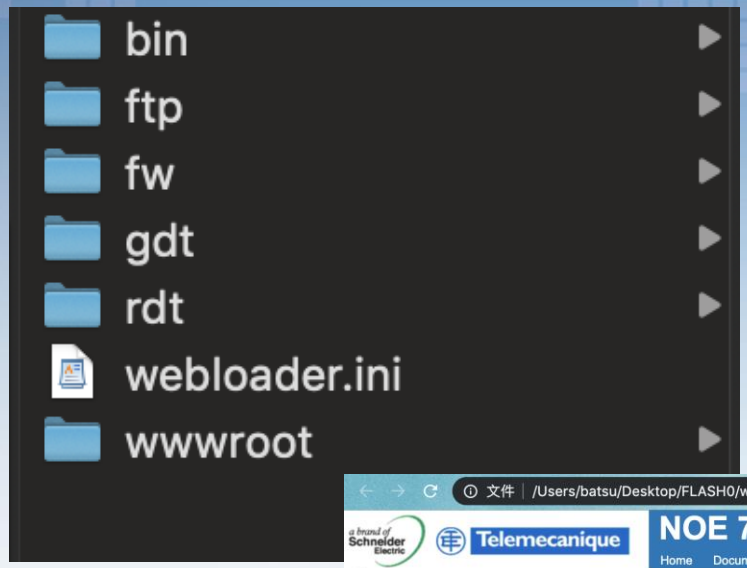
OWASP
Open Web Application Security Project

0x01 典型PLC固件逆向



工控比赛中遇到的典型问题

解压后的文件目录



作为鬼畜的web手，先看看index.html有啥鬼东西

工控比赛中遇到的典型问题

通过谷歌找到一个相关的资料

VXWorks固件NOE-711后门账号漏洞分析

by zgd

2019年1月

以Schneider厂商的Quantum系列系列PLC的NOE-711以太网模块的固件为例，学习和了解一下基于VxWorks操作系统的嵌入式设备固件的一些常用分析方法。

（I）web配置app默认账户和密码的获取

github下载到含固件的文件：

fw/fw.ini文件中包含固件版本信息：

wwwroot/classes/内的jar文件是Web配置端APP文件

使用JD-GUI反编译软件对jar文件逆向分析，并搜索PASSWORD，直接找到了web配置app默认的用户名和密码。

https://blog.csdn.net/Homewm/article/details/90515507

工控比赛中遇到的典型问题

OWASP
Open Web Application Security Project

那么jd-gui来了，轻松获取硬编码的默认账号密码

SAComm.jar

- META-INF
- com.schneiderautomation
  - dt
  - factorycast
  - ftpsession
  - misc
    - Compare.class
    - SortVector.class
    - TextFiles.class
  - namespace
  - vars
- DummyApplet.class

Search

Search string (* = any string, ? = any character):

PASSWORD

Search For: Type, Constructor, String Constant, Field, Method

Limit To: Declarations, References

1 matching entry:

- SAComm.jar
  - com.schneiderautomation.misc
    - TextFiles.class

Open Cancel

TextFiles.class

```
package com.schneiderautomation.misc;

import com.schneiderautomation.ftpsession.FTPSession;
import com.schneiderautomation.ftpsession.FileInfo;
import com.schneiderautomation.ftpsession.FtpSessionException;
import java.io.BufferedInputStream;
import java.io.BufferedOutputStream;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.io.Writer;
import java.net.URL;
import java.net.URLConnection;
import java.util.Locale;
import java.util.Vector;

public class TextFiles
{
  private static final String USER = "sysdiag";
  private static final String PASSWORD = "factorycast@schneider";
  private boolean fAutoConnect = true;
  private Thread engine;
  private String host;
  private Locale locale;
  private FTPSession ftp;

  public TextFiles(String paramString, Locale paramLocale)
  {
    this.host = paramString;
    this.locale = paramLocale;
  }

  public synchronized String[] list(String paramString1, String paramString2)
    throws IOException
  {
    connect();
    String[] arrayOfString;
    try
    {
      this.ftp.changeDir(paramString1);
      Vector localVector = this.ftp.getFileList();
      for (int i = localVector.size() - 1; i >= 0; i--)
      {
        FileInfo localFileInfo1 = (FileInfo)localVector.elementAt(i);
        if (!localFileInfo1.getName().endsWith(paramString2)) {
          localVector.removeElementAt(i);
```

0x02 工控协议分析3

OWASP
Open Web Application
Security Project

Wireshark查看数据包，发现存在两种协议:modbus/tcp和S7COMM

719	17.975339	192.168.254.134	192.168.254.143	Modbus/TCP	66
720	17.975950	192.168.254.143	192.168.254.134	Modbus/TCP	65
721	18.006867	192.168.254.134	192.168.254.143	TCP	62
722	18.007068	192.168.254.143	192.168.254.134	TCP	62
723	18.007076	192.168.254.134	192.168.254.143	TCP	54
724	18.007444	192.168.254.134	192.168.254.143	Modbus/TCP	133
725	18.007649	192.168.254.134	192.168.254.143	Modbus/TCP	66
726	18.008772	192.168.254.143	192.168.254.134	Modbus/TCP	66
727	18.009290	192.168.254.134	192.168.254.143	TCP	54
728	18.009454	192.168.254.143	192.168.254.134	TCP	60
729	18.009464	192.168.254.143	192.168.254.134	Modbus/TCP	103
730	18.010009	192.168.254.143	192.168.254.134	TCP	60
731	18.010022	192.168.254.134	192.168.254.143	TCP	54
732	18.037685	192.168.254.134	192.168.254.34	TCP	54
733	18.039984	192.168.254.134	192.168.254.143	Modbus/TCP	66
734	18.040699	192.168.254.143	192.168.254.134	Modbus/TCP	103
735	18.042588	192.168.254.134	192.168.254.34	S7COMM	85
736	18.042949	192.168.254.34	192.168.254.134	S7COMM	97
737	18.148516	192.168.254.134	192.168.254.143	Modbus/TCP	66
738	18.149329	192.168.254.143	192.168.254.134	Modbus/TCP	65
739	18.150458	192.168.254.134	192.168.254.34	S7COMM	85
740	18.150797	192.168.254.34	192.168.254.134	S7COMM	97
741	18.180658	192.168.254.134	192.168.254.143	Modbus/TCP	66

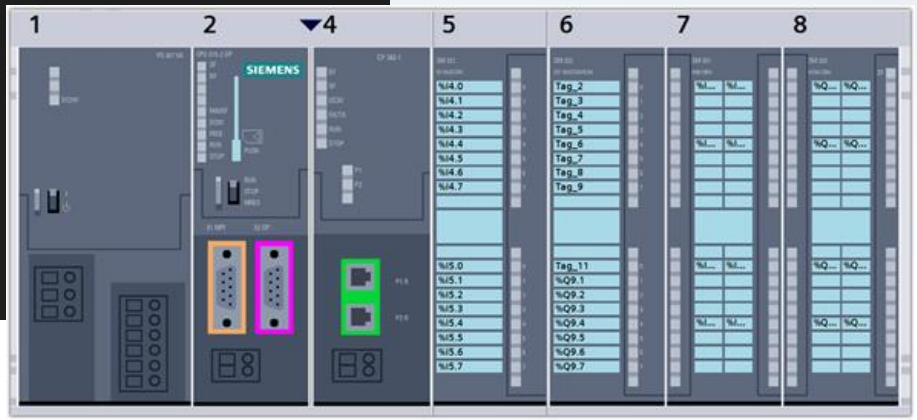
工控比赛中遇到的典型问题

先分析S7COMM协议

s7comm.py

```python
import pyshark

def get_func_s7():
    try:
        captures = pyshark.FileCapture("3.pcap")
        func_codes = {}
        for c in captures:
            for pkt in c:
                if pkt.layer_name == "s7comm":
                    if hasattr(pkt, "param_func"):
                        func_code = pkt.param_func
                        if func_code in func_codes:
                            func_codes[func_code] += 1
                        else:
                            func_codes[func_code] = 1
        print(func_codes)
    except Exception as e:
        print(e)

if __name__ == '__main__':
    get_func_s7()
```

```
➜ Desktop python3 s7comm.py

{'0x000000f0': 2, '0x00000004': 6195}
```



工控比赛中遇到的典型问题

解析modbus协议

附录1：MODBUS-RTU功能码

最常用功能码：

下面“线圈”“寄存器”其实分别直的就是“位变量”“16位变量”

01 (0x01)	读线圈
02 (0x02)	读离散量输入
03 (0x03)	读保持寄存器
04(0x04)	读输入寄存器
05 (0x05)	写单个线圈
06 (0x06)	写单个寄存器
15 (0x0F)	写多个线圈
16 (0x10)	写多个寄存器

OWASP
Open Web Application Security Project

工控比赛中遇到的典型问题

通过浏览数据包发现存在1、2、3、4、16这五个功能码

16（0x10）

```
 1, Func:  16: Write Multiple Registers
 1, Func:  16: Write Multiple Registers
 1, Func:  16: Write Multiple Registers
 1, Func:  16: Write Multiple Registers
 1, Func:  16: Write Multiple Registers
 1, Func:  16: Write Multiple Registers
 1, Func:  16: Write Multiple Registers
 1, Func:  16: Write Multiple Registers
 1, Func:  16: Write Multiple Registers
 1, Func:  16: Write Multiple Registers
 1, Func:  16: Write Multiple Registers
 1, Func:  16: Write Multiple Registers
 1, Func:  16: Write Multiple Registers
```

4(0x04)

```
1; Unit:   1, Func:   4: Read Input Registers
2; Unit:   1, Func:   4: Read Input Registers
3; Unit:   1, Func:   4: Read Input Registers
4; Unit:   1, Func:   4: Read Input Registers
5; Unit:   1, Func:   4: Read Input Registers
6; Unit:   1, Func:   4: Read Input Registers
7; Unit:   1, Func:   4: Read Input Registers
8; Unit:   1, Func:   4: Read Input Registers
9; Unit:   1, Func:   4: Read Input Registers
10; Unit:  1, Func:   4: Read Input Registers
11; Unit:  1, Func:   4: Read Input Registers
12; Unit:  1, Func:   4: Read Input Registers
13; Unit:  1, Func:   4: Read Input Registers
```

1(0x01)

```
2; Unit:   1, Func:   1: Read Coils
3; Unit:   1, Func:   1: Read Coils
4; Unit:   1, Func:   1: Read Coils
5; Unit:   1, Func:   1: Read Coils
6; Unit:   1, Func:   1: Read Coils
7; Unit:   1, Func:   1: Read Coils
8; Unit:   1, Func:   1: Read Coils
9; Unit:   1, Func:   1: Read Coils
0; Unit:   1, Func:   1: Read Coils
1; Unit:   1, Func:   1: Read Coils
2; Unit:   1, Func:   1: Read Coils
```

2(0x02)

```
it:  1, Func:  2: Read Discrete Inputs
it:  1, Func:  2: Read Discrete Inputs
it:  1, Func:  2: Read Discrete Inputs
it:  1, Func:  2: Read Discrete Inputs
it:  1, Func:  2: Read Discrete Inputs
it:  1, Func:  2: Read Discrete Inputs
it:  1, Func:  2: Read Discrete Inputs
it:  1, Func:  2: Read Discrete Inputs
it:  1, Func:  2: Read Discrete Inputs
it:  1, Func:  2: Read Discrete Inputs
it:  1, Func:  2: Read Discrete Inputs
it:  1, Func:  2: Read Discrete Inputs
it:  1, Func:  2: Read Discrete Inputs
it:  1, Func:  2: Read Discrete Inputs
it:  1, Func:  2: Read Discrete Inputs
it:  1, Func:  2: Read Discrete Inputs
it:  1, Func:  2: Read Discrete Inputs
```

3(0x03)

```
781; Unit:   1, Func:   3: Read Holding Registers
782; Unit:   1, Func:   3: Read Holding Registers
783; Unit:   1, Func:   3: Read Holding Registers
784; Unit:   1, Func:   3: Read Holding Registers
785; Unit:   1, Func:   3: Read Holding Registers
786; Unit:   1, Func:   3: Read Holding Registers
787; Unit:   1, Func:   3: Read Holding Registers
788; Unit:   1, Func:   3: Read Holding Registers
789; Unit:   1, Func:   3: Read Holding Registers
790; Unit:   1, Func:   3: Read Holding Registers
791; Unit:   1, Func:   3: Read Holding Registers
792; Unit:   1, Func:   3: Read Holding Registers
793; Unit:   1, Func:   3: Read Holding Registers
```

工控比赛中遇到的典型问题

OWASP
Open Web Application Security Project

四个读操作，一个写操作，因此重点看看16的状态码内容，可以看到有一个和其他完全不一样的报文。

modbus&&modbus.func_code==16&&ip.src==192.168.254.134

No.	Time	Source	Destination	Protocol	Length	Info
8591	147.304412	192.168.254.134	192.168.254.143	Modbus…	133	Query: Trans: 0; Unit: 1, Func: 16: Write Multiple Registers
8735	150.362988	192.168.254.134	192.168.254.143	Modbus…	133	Query: Trans: 0; Unit: 1, Func: 16: Write Multiple Registers
8899	153.371023	192.168.254.134	192.168.254.143	Modbus…	133	Query: Trans: 0; Unit: 1, Func: 16: Write Multiple Registers
9077	156.374289	192.168.254.134	192.168.254.143	Modbus…	133	Query: Trans: 0; Unit: 1, Func: 16: Write Multiple Registers
9262	159.376112	192.168.254.134	192.168.254.143	Modbus…	133	Query: Trans: 0; Unit: 1, Func: 16: Write Multiple Registers
9450	162.377607	192.168.254.134	192.168.254.143	Modbus…	133	Query: Trans: 0; Unit: 1, Func: 16: Write Multiple Registers
9624	165.378542	192.168.254.134	192.168.254.143	Modbus…	133	Query: Trans: 0; Unit: 1, Func: 16: Write Multiple Registers
9799	168.382089	192.168.254.134	192.168.254.143	Modbus…	133	Query: Trans: 0; Unit: 1, Func: 16: Write Multiple Registers
9964	171.383455	192.168.254.134	192.168.254.143	Modbus…	133	Query: Trans: 0; Unit: 1, Func: 16: Write Multiple Registers
10148	174.384722	192.168.254.134	192.168.254.143	Modbus…	133	Query: Trans: 0; Unit: 1, Func: 16: Write Multiple Registers
10331	177.385612	192.168.254.134	192.168.254.143	Modbus…	133	Query: Trans: 0; Unit: 1, Func: 16: Write Multiple Registers
10511	180.386037	192.168.254.134	192.168.254.143	Modbus…	133	Query: Trans: 0; Unit: 1, Func: 16: Write Multiple Registers
10693	183.387124	192.168.254.134	192.168.254.143	Modbus…	133	Query: Trans: 0; Unit: 1, Func: 16: Write Multiple Registers
10874	186.402158	192.168.254.134	192.168.254.143	Modbus…	133	Query: Trans: 0; Unit: 1, Func: 16: Write Multiple Registers
11052	189.402833	192.168.254.134	192.168.254.143	Modbus…	133	Query: Trans: 0; Unit: 1, Func: 16: Write Multiple Registers
11238	192.437445	192.168.254.134	192.168.254.143	Modbus…	133	Query: Trans: 0; Unit: 1, Func: 16: Write Multiple Registers
11420	195.439855	192.168.254.134	192.168.254.143	Modbus…	133	Query: Trans: 0; Unit: 1, Func: 16: Write Multiple Registers
11598	198.440702	192.168.254.134	192.168.254.143	Modbus…	133	Query: Trans: 0; Unit: 1, Func: 16: Write Multiple Registers
11779	201.442304	192.168.254.134	192.168.254.143	Modbus…	133	Query: Trans: 0; Unit: 7, Func: 16: Write Multiple Registers
11961	204.442627	192.168.254.134	192.168.254.143	Modbus…	133	Query: Trans: 0; Unit: 1, Func: 16: Write Multiple Registers
12142	207.443953	192.168.254.134	192.168.254.143	Modbus…	133	Query: Trans: 0; Unit: 1, Func: 16: Write Multiple Registers
12323	210.467593	192.168.254.134	192.168.254.143	Modbus…	133	Query: Trans: 0; Unit: 1, Func: 16: Write Multiple Registers
12507	213.468040	192.168.254.134	192.168.254.143	Modbus…	133	Query: Trans: 0; Unit: 1, Func: 16: Write Multiple Registers
12688	216.489826	192.168.254.134	192.168.254.143	Modbus…	133	Query: Trans: 0; Unit: 1, Func: 16: Write Multiple Registers
12876	219.491597	192.168.254.134	192.168.254.143	Modbus…	133	Query: Trans: 0; Unit: 1, Func: 16: Write Multiple Registers

工控比赛中遇到的典型问题

追踪流看看，好像和别的不太一样，看起来像decode

异常的包

Wireshark · 追踪 TCP 流 (tcp.stream eq 66) · question_1531222648_ctf03.pcap

```
.....9......2..6d6f646275734943537365637572697479 57696e.........?.E..K...<0)+Z............
```

正常的包

Wireshark · 追踪 TCP 流 (tcp.stream eq 52) · question_1531222648_ctf03.pcap

```
.....9......2.....U.vOv..}...:y...U.vOv..}...:y.C....E/....2.I
...        .7N..k.A..+.............
```

工控比赛中遇到的典型问题

OWASP
Open Web Application Security Project

好像是个hex，解码后是个flag

```
➜  ~ python
Python 2.7.15 (default, Jul 23 2018, 21:27:06)
[GCC 4.2.1 Compatible Apple LLVM 9.1.0 (clang-902.0.39.2)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> a='6d6f6462757349435373656375726974795769 6e'
>>> print(a.decode('hex'))
modbusICSsecurityWin
>>>
```

比赛和实际结合的一些经验体会

数据通信
（协议）
协议fuzz..

工控软件、固件

软件fuzz、硬编码...

系统版本
(win\LINUX)

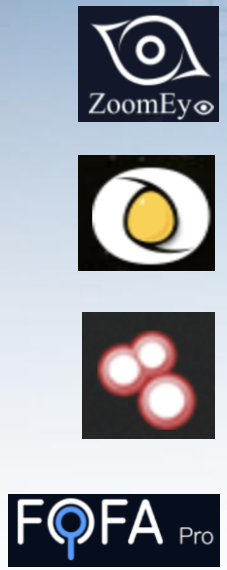
21、23、5900..
、445、3389..

比赛和实际结合的一些经验体会

看到这里好像有些传统安全的影子了，但是又不完全一样，那么怎么结合二者呢。

然后我干了这样一件事

比赛和实际结合的一些经验体会



modbus

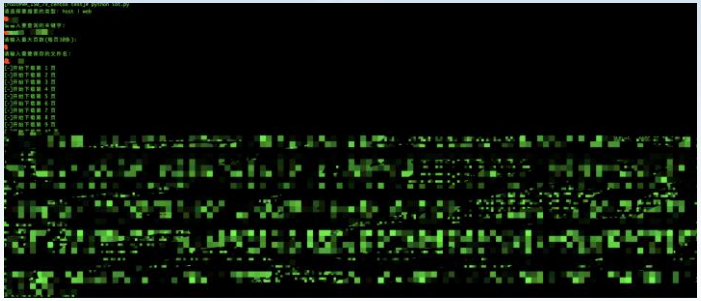
101协议

104协议

502

S7comm

DNP3



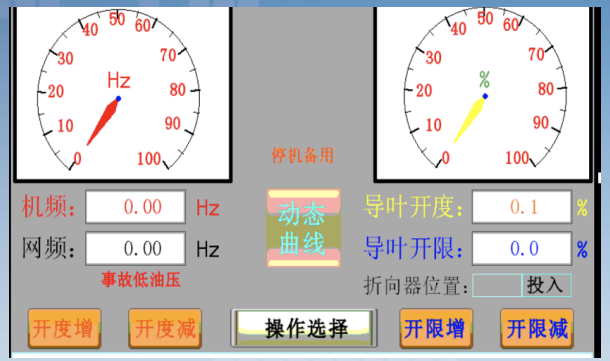
粗IP池

。。。

比赛和实际结合的一些经验体会

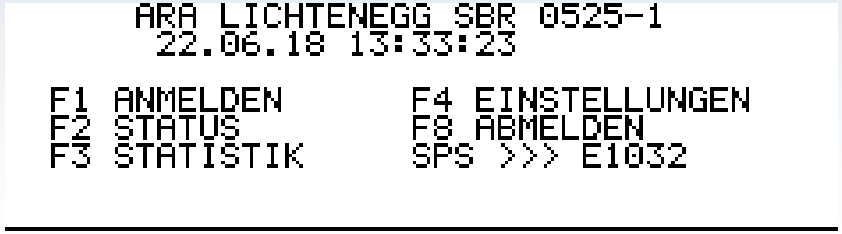
OWASP
Open Web Application Security Project

某发电厂的HMI系统

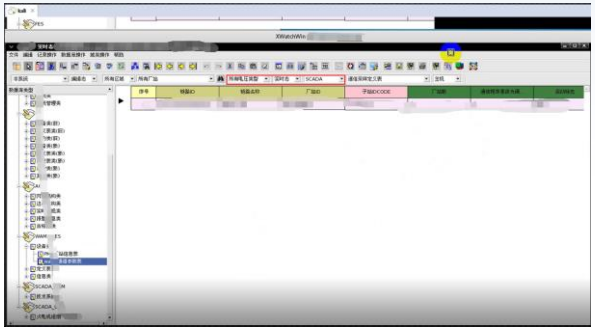


某环境监测系统

某断路器控制系统



某scatas数据采集系统



比赛和实际结合的一些经验体会

某win系统上位机

```
[-] Handler failed to bind to [redacted]
[redacted] Register keyboard
[redacted] - Opening "Run Application"
[redacted] - Waiting 20 seconds...
[redacted] - Waiting 20 seconds...
[redacted] - Typing and executing payload
[redacted] 38 +0800

ls
1.txt
2.txt
3.txt
anaconda-ks.cfg
awrrpt_1_284_286.html
install.log
install.log.syslog
公共的
模板
视频
图片
文档
下载
音乐
桌面
```

```
ifconfig
eth0      Link encap:Ethernet  [redacted]
          inet addr:[redacted]
          inet6 addr: fe80::6e3b:e5[redacted] Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2589862720 errors:0 dropped:0 overruns:0 frame:0
          TX packets:662949062 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:198559397008 (184.9 GiB)  TX bytes:134302597441 (125.0 GiB)
          Interrupt:59

eth2      Link encap:Ethernet  [redacted]
          inet addr:10.0.0.11  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: fe80::6e3b:e5ff:fea5:2026/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:363985733148 errors:0 dropped:0 overruns:0 frame:0
          TX packets:360882536085 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:179496613996708 (163.2 TiB)  TX bytes:28343355935132 (25.7 TiB)
          Interrupt:67

eth3      Link encap:Ethernet  [redacted]
          inet addr:10.0.0.111  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: fe80::6e3b:e5ff:fea5:2027/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1459143863 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:100933263884 (94.0 GiB)  TX bytes:704 (704.0 b)
          Interrupt:71

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1584058096986 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1584058096986 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:100437099915402 (91.3 TiB)  TX bytes:100437099915402 (91.3 TiB)
```

```
cat 1.txt
[redacted]
Full thread dump OpenJDK 64-Bit Server VM (23.2-b09 mixed mode):

"Attach Listener" daemon prio=10 tid=0x00007fc954001000 nid=0x6bdb waiting on condition [0x0000000000000000]
   java.lang.Thread.State: RUNNABLE

"DestroyJavaVM" prio=10 tid=0x00007fc9ec008000 nid=0x6b7d waiting on condition [0x0000000000000000]
   java.lang.Thread.State: RUNNABLE

"Thread-26" prio=10 tid=0x00007fc9ec6e8000 nid=0x6bb7 waiting for monitor entry [0x00007fc9a0c68000]
   java.lang.Thread.State: BLOCKED (on object monitor)
        at com.sccl.socket.server.StreamTool.setSendData(StreamTool.java:178)
        - waiting to lock <0x00000007d5f3d2d0> (a java.util.HashMap)
        at com.sccl.socket.server.StreamTool.readStream(StreamTool.java:79)
        at com.sccl.socket.thread.ClientHandler.run(ClientHandler.java:65)
        at java.lang.Thread.run(Thread.java:722)

"Thread-25" prio=10 tid=0x00007fc9ec6e6000 nid=0x6bb6 waiting for monitor entry [0x00007fc9a0d69000]
   java.lang.Thread.State: BLOCKED (on object monitor)
        at com.sccl.socket.server.StreamTool.setSendData(StreamTool.java:164)
        - waiting to lock <0x00000007d5ee3048> (a java.util.HashMap)
        at com.sccl.socket.server.StreamTool.readStream(StreamTool.java:79)
        at com.sccl.socket.thread.ClientHandler.run(ClientHandler.java:65)
        at java.lang.Thread.run(Thread.java:722)

"Thread-24" prio=10 tid=0x00007fc9ec6e4800 nid=0x6bb5 waiting for monitor entry [0x00007fc9a0e6a000]
   java.lang.Thread.State: BLOCKED (on object monitor)
        at com.sccl.socket.server.StreamTool.setSendData(StreamTool.java:148)
        - waiting to lock <0x00000007d5e942a0> (a java.util.HashMap)
        at com.sccl.socket.server.StreamTool.readStream(StreamTool.java:79)
```

```
br0       Link encap:Ethernet  HWaddr [redacted]
          inet addr:192.[redacted] Bcast:[redacted] Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:27985 errors:0 dropped:0 overruns:0 frame:0
          TX packets:27266 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1957702 (1.8 MiB)  TX bytes:10481168 (9.9 MiB)

br0:0     Link encap:Ethernet  HWaddr [redacted]
          inet addr[redacted] Bcast:[redacted] Mask:255.2[redacted]
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

eth2      Link encap:Ethernet  HWaddr [redacted]
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:28111 errors:0 dropped:0 overruns:0 frame:0
          TX packets:27268 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2357052 (2.2 MiB)  TX bytes:10481296 (9.9 MiB)
          Interrupt:3

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:6434 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6434 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:266972 (260.7 KiB)  TX bytes:266972 (260.7 KiB)
```

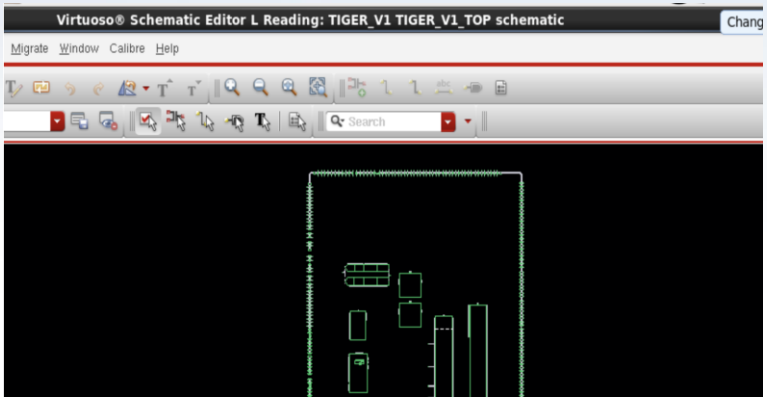
比赛和实际结合的一些经验体会

OWASP
Open Web Application Security Project

某比特率误差检测系统

智能网关

某工控系统编程上位机

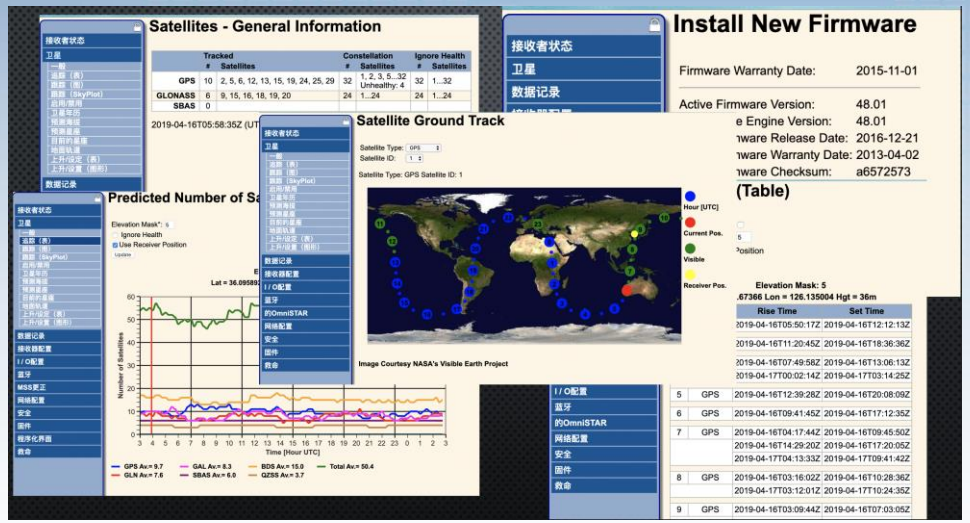


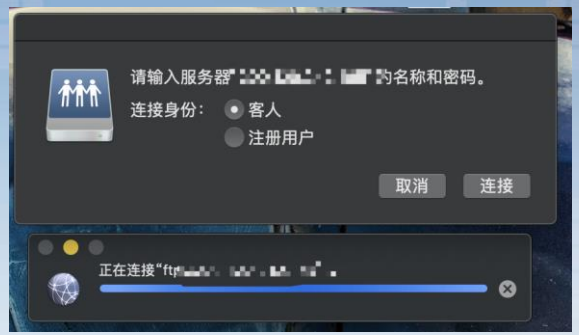

```
SNCLI>?
aim      - This is voip command.
devel    - Developer tools
exit     - End session
help     - Description of the interactive help system
mdm      - This is mdm command.
sys      - This is sys manage command.
test     - This is test command.

SNCLI>sys
SYS>
```

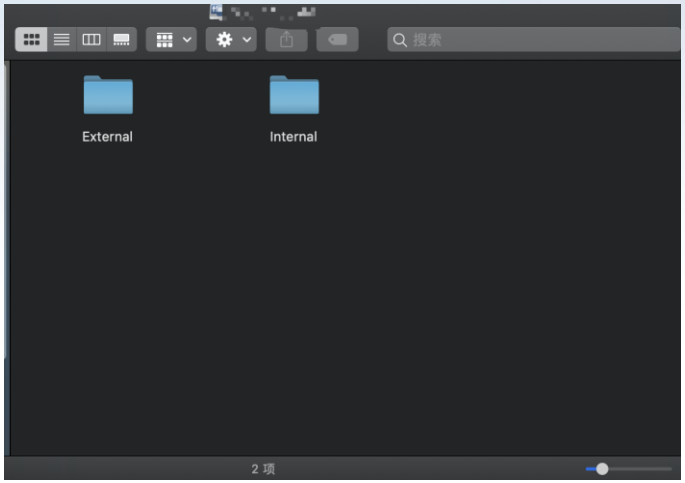
比赛和实际结合的一些经验体会

OWASP
Open Web Application Security Project

不知道什么的系统

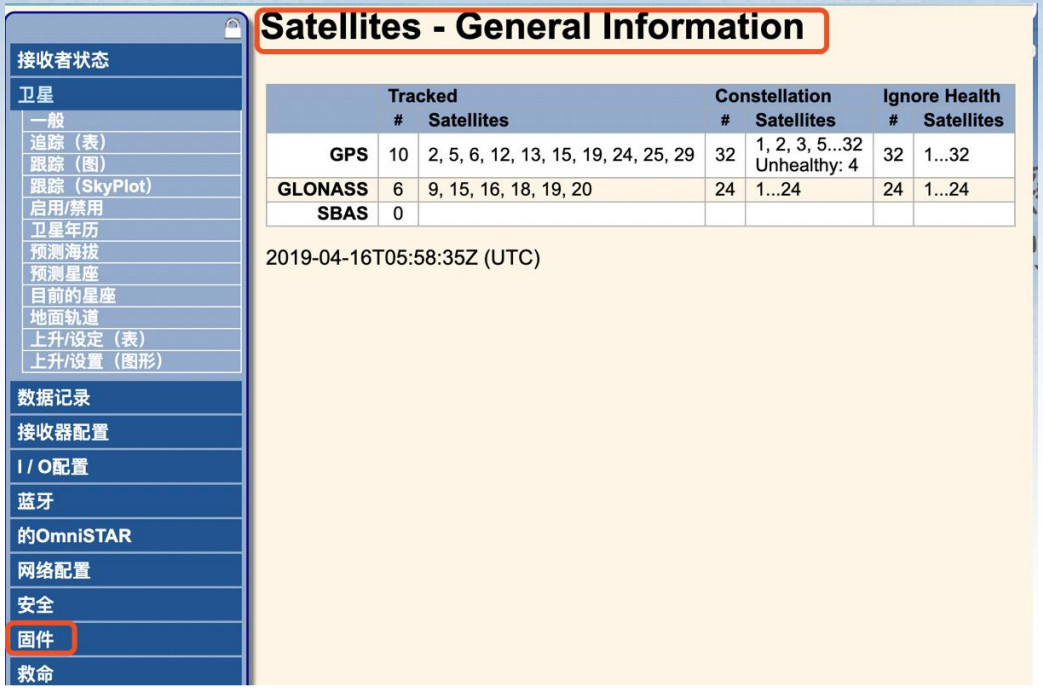






OWASP
Open Web Application Security Project

如果说我将这些页面保存，利用现有比较成熟的文字、图像处理来识别工业控制的关键字，例如：test: Satellites –General Information、固件等



	Tracked		Constellation		Ignore Health	
	#	Satellites	#	Satellites	#	Satellites
GPS	10	2, 5, 6, 12, 13, 15, 19, 24, 25, 29	32	1, 2, 3, 5...32 Unhealthy: 4	32	1...32
GLONASS	6	9, 15, 16, 18, 19, 20	24	1...24	24	1...24
SBAS	0					

那么我们是否能更智能，快速的找出工业控制设备？

OWASP
Open Web Application Security Project

Thank you