# 协议分析又来了

*发表于 2021-01-25 更新于 2021-02-16 分类于 Challenge ， 2019 ， 工业信息安全技能大赛 ， 哈尔滨站*
*Challenge | 2019 | 工业信息安全技能大赛 | 哈尔滨站 | 协议分析又来了*

*点击此处获得更好的阅读体验*

---

## *WriteUp来源*

*https://www.cnpanda.net/ctf/415.html*

## *题目描述*

> *某安全团队捕获黑客滥用工业互联网协议，分析协议内容，找出flag。*

## *题目考点*

## *解题思路*

*从wireshark中发现是LSIS PLC的相关通信内容。在数据包中可观察到对plc的寄存器进行遍历读取，但中间会参杂一些其他的读取。以100为长度。*

*通过程序对相关数据进行筛选*

*对数据进行`xor 0xFF`还原并修整提取*

*使用dnspy对程序进行反编译，找到其中的资源mainwindow.baml*
*复制其中的Grid元素中的RadioButton的所有标签。*

*通过visual studio新建一个WPF工程，把上述内容贴入工程的xaml中即可见到flag*

```
private void Btnreadfile_Click(object sender, EventArgs e)
{
    byte[] read = new byte[9999];
    OpenFileDialog fileDialog = new OpenFileDialog();
    fileDialog.Multiselect = true;
    fileDialog.Title = "请选择文件";
    fileDialog.Filter = "所有文件|*.*"; //设置要选择的文件的类型
    if (fileDialog.ShowDialog() == DialogResult.OK)
    {
        string file = fileDialog.FileName;//返回文件的完整路径
        PcapNGFileReader packetReader = new PcapNGFileReader(FileToStream(file));
        var packets= packetReader.ReadPackets();
        bool isRead = false;
        byte[] readbyte = new byte[65500];
        int iterCount = 100;
        foreach(var item in packets)
        {
            var data = item.Payload.ToArray();
            //读取数据
            if (isRead)
            {
                for(int j = 99; j >= 0; j--)
                {
                    int sub = 99 - j;
                    readbyte[iterCount + j] = (byte)(data[data.Length-sub-1]);

                }
                iterCount += 100;
                isRead = false;
            }
            //处理是否读取保存下一个包
            if (data.Length > 91&&data.Length<95 && data[data.Length - 4] == 0x30 && data[data.Length - 3] == 0x30 && data[data.Length - 2] == 0x64 && data[data.Length -
            {
                isRead = true;
            }
            else
            {
                isRead = false;
            }
        }
        int endloc = 100;
        int count = 1000;
        for (int i = 100; i < 65500; i++)
        {
            if (readbyte[i] == 0)
            {
                endloc = i;
                count--;
                if (count == 0)
                {
                    break;
                }
            }
            else
            {
                count = 1000;
                endloc = 100;
            }
        }
        byte[] getval = new byte[endloc - 1100];
        for (int i = 0; i < getval.Length; i++)
        {
            getval[i] = (byte)(readbyte[i + 100] ^ 0xFF);
        }

        StreamToFile(BytesToStream(getval), file + ".exe");
    }
}

/// <summary>
/// byte[]转换成Stream
/// </summary>
```

```
/// <param name="bytes"></param>
/// <returns></returns>
public Stream BytesToStream(byte[] bytes)
{
    Stream stream = new MemoryStream(bytes);
    return stream;
}

/// <summary>
/// Stream转换成byte[]
/// </summary>
/// <param name="stream"></param>
/// <returns></returns>
public byte[] StreamToBytes(Stream stream)
{
    byte[] bytes = new byte[stream.Length];
    stream.Read(bytes, 0, bytes.Length);
    stream.Seek(0, SeekOrigin.Begin); // 设置当前流的位置为流的开始
    return bytes;
}

/// <summary>
/// 从文件读取Stream
/// </summary>
/// <param name="path"></param>
/// <returns></returns>
public Stream FileToStream(string path)
{
    FileStream fileStream = new FileStream(path, FileMode.Open, FileAccess.Read, FileShare.Read); // 打开文件
    byte[] bytes = new byte[fileStream.Length]; // 读取文件的byte[]
    fileStream.Read(bytes, 0, bytes.Length);
    fileStream.Close();
    Stream stream = new MemoryStream(bytes); // 把byte[]转换成Stream
    return stream;
}

/// <summary>
/// 将Stream写入文件
/// </summary>
/// <param name="stream"></param>
/// <param name="path"></param>
public void StreamToFile(Stream stream, string path)
{
    byte[] bytes = new byte[stream.Length]; // 把Stream转换成byte[]
    stream.Read(bytes, 0, bytes.Length);
    stream.Seek(0, SeekOrigin.Begin); // 设置当前流的位置为流的开始
    FileStream fs = new FileStream(path, FileMode.Create); // 把byte[]写入文件
    BinaryWriter bw = new BinaryWriter(fs);
    bw.Write(bytes);
    bw.Close();
    fs.Close();
}
```

## *Flag*

```
flag{ICSC-F1A!}
```

- ***本文作者：*** *CTFHub*
- ***本文链接：*** *https://writeup.ctfhub.com/Challenge/2019/工业信息安全技能大赛/哈尔滨站/9Hd9hdxRK7cx4YiFucwV2n.html*
- ***版权声明：*** *本博客所有文章除特别声明外，均采用 BY-NC-SA 许可协议。转载请注明出处！*

*# Challenge # 2019 # 工业信息安全技能大赛 # 哈尔滨站*
*flag在哪*
*智能变电站设备异常诊断*