Fun with VxWorks

RAPID7

introduction

RAPID7

Chief Security Officer

metasploit

Founder & Chief Architect

with help from...

Dillon Beresford (NSS Labs)

Shawn Merdinger

David Maynor

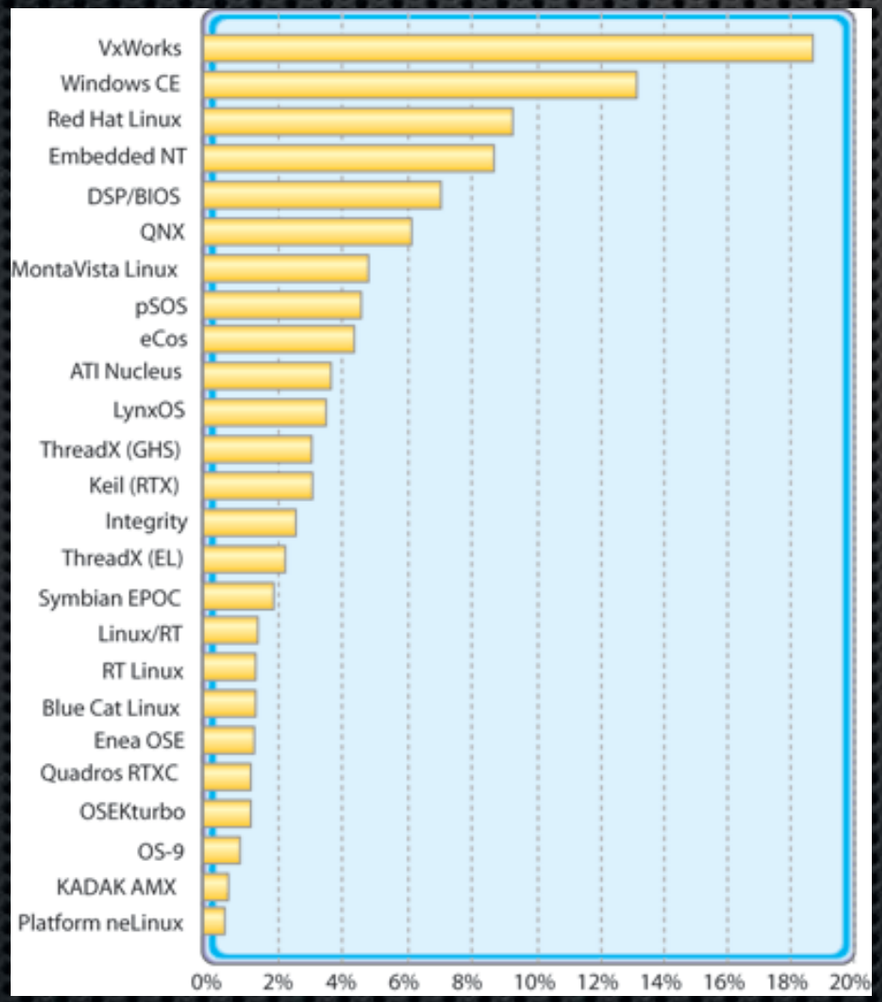
R3L1K

FX

introduction

VxWorks

- An embedded, real-time operating system
- Most widely deployed embedded OS in ~2005



Claimed 300 million devices in 2006

Produced by Wind River Systems, now owned by Intel

http://www.eetimes.com/discussion/other/4025539/Embedded-systems-survey-Operating-systems-up-for-grabs

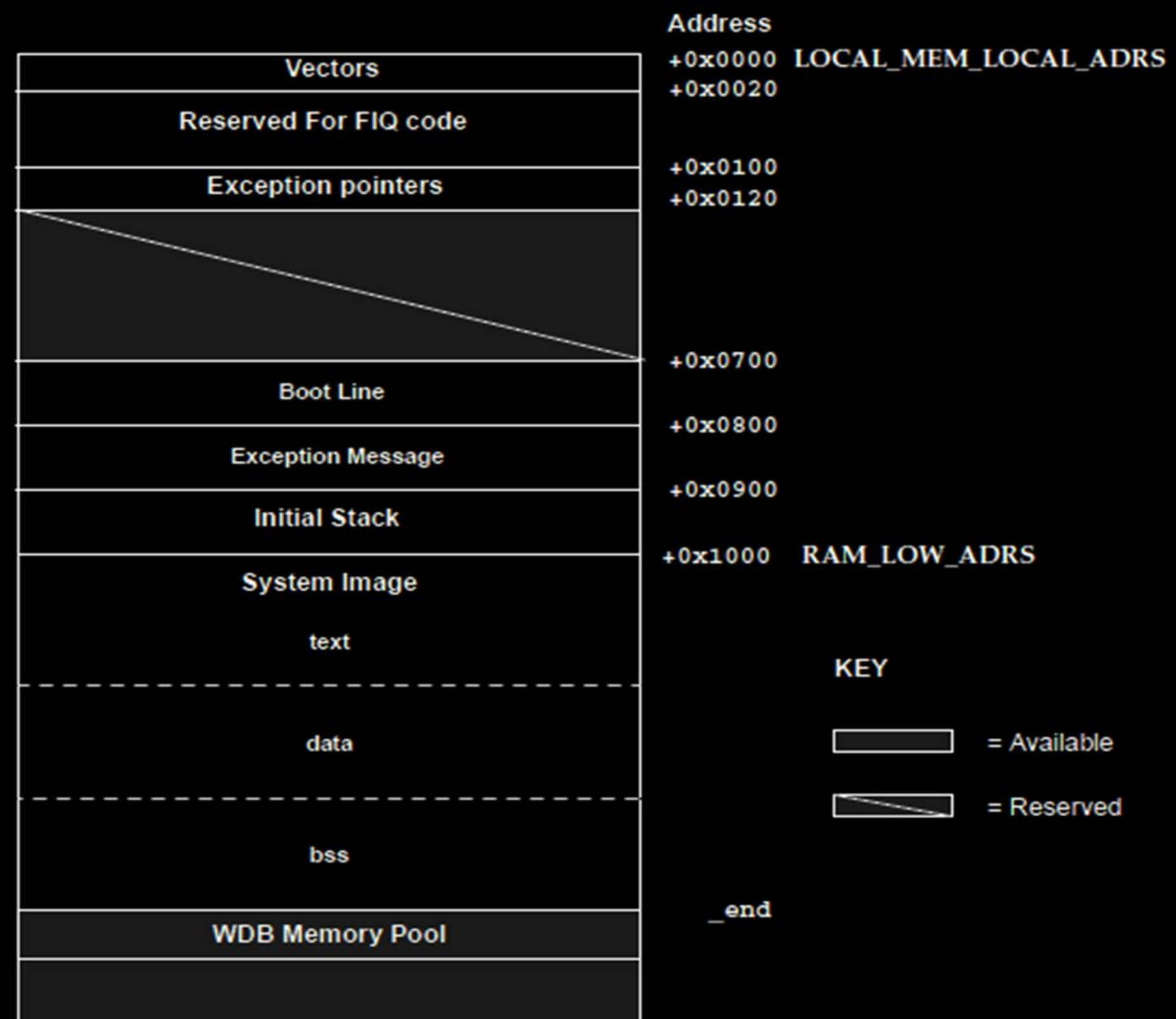
internals

VxWorks internals

- Support for dozens of hardware platforms
  - PowerPC, ARM, MIPS, x86, i960, SPARC
- All “applications” run as kernel threads
- Little memory protection between apps
- Everything runs with the highest privileges
- …but not necessarily the highest priority.

memory layout

Figure G-2 VxWorks System Memory Layout (ARM)

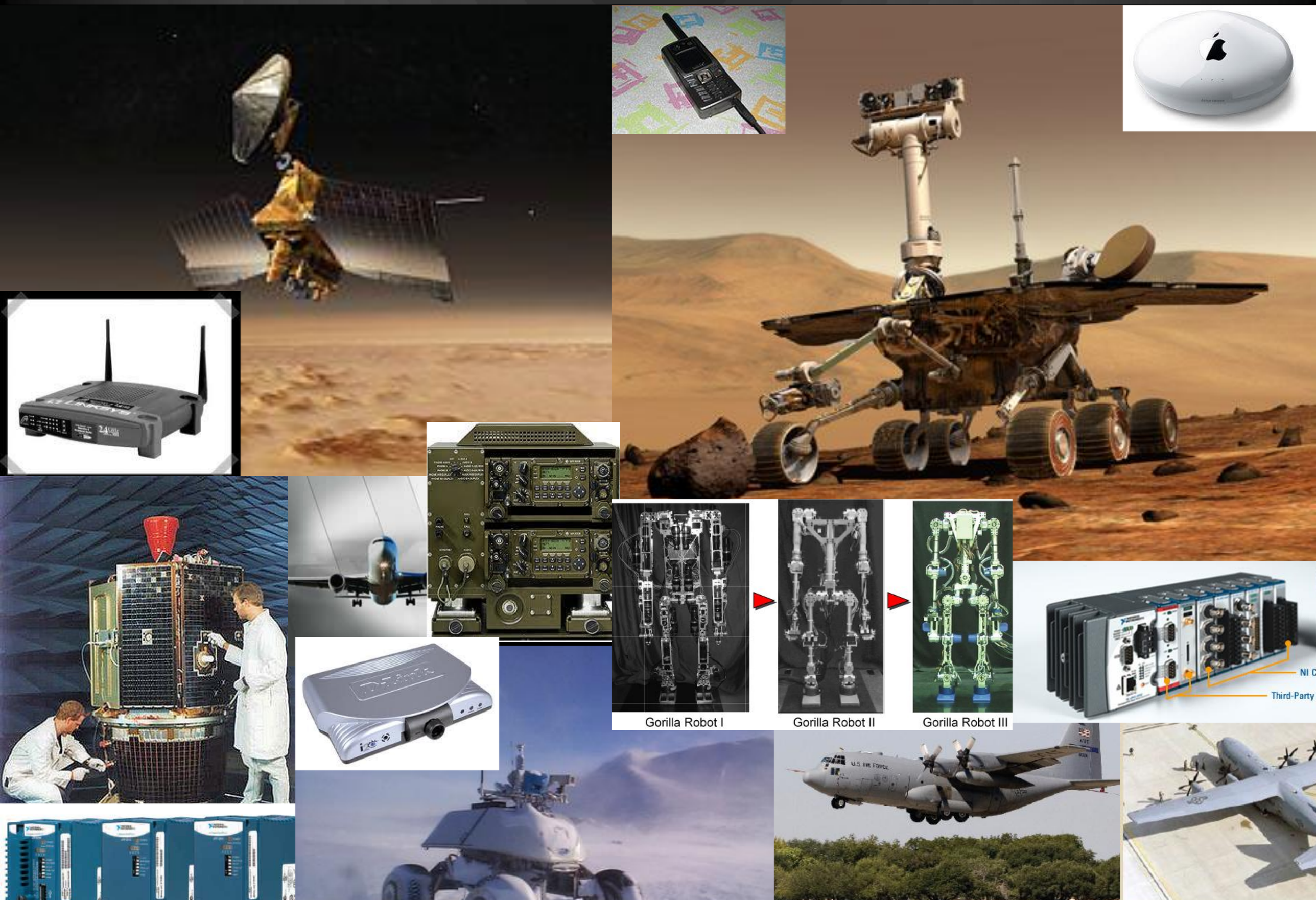


vxworks systems

VxWorks is everywhere

- VoIP phones, telecom equipment, switches
- Satellite, WiFi, microwave, sensors
- RAID controllers and fibre channel switches
- Video conferencing equipment
- Industrial control monitors
- Military routing equipment
- Automobile controls
- Spacecraft

vxworks systems



vxworks customers

vulnerabilities

VxWorks security

- Only 12 CVEs mention VxWorks
- Only 2 refer to flaws in the actual OS
- Bug free or just too boring to hack?

vulnerabilities

A common thread…

- The VxWorks debug service on port 17185
- Lightly mentioned in 2002, 2004, 2005
- CVE-2005-3715 & CVE-2005-3804
- No information on the protocol
- Works on all architectures

"Allows attackers to access the phone OS, obtain sensitive information, and cause a denial of service"

vxworks debug service

Protocol information

- Basic API mentioned in dev docs
- Signed up for a Tornado eval kit
- Wouldn’t connect to VxWorks 5 targets
- Gave up and searched Google…

useful documentation

pudn 程序员联合开发网
programmers united develop net

登录 | 注册会员 | 帮助 | 联系站长
[English Version]

首页 目录 下载 上传 VIP会员 搜索 阅读 留言薄

Pudn.com > Downloads > 源码/资料 > 嵌入式/单片机编程 > VxWorks > **vxWorks_sourcecode**

搜索

文件名称: **vxWorks_sourcecode** 下载 收藏 [5 4 3 2 1]

所属分类:	VxWorks
开发工具:	C-C++
文件大小:	9308 KB
上传时间:	2008-06-09
下载次数:	123
提 供 者:	lixzh

详细说明: **vxworks源码源码解读是学习vxworks的最佳途径-VxWorks source code interpretation is the best way to learn VxWorks**

近期下载过的用户: x86guru 思宇 裴海全 spi liwei wanghui Mr zao W.N. Dong 李勃 wenzk 张玉松 Mr zeng yhu liuxn 于献榕 [查看上载者**lixzh**的更多信息]

相关搜索: vxworks vxworks 源码 vxworks source vxworks sourcecode vxworks 内核

输入关键字，在本站107万海量源码库中尽情搜索: vxworks 搜索 帮助

[at91rm9200.rar] - at91rm9200 的 vxworks bsp包。
[vxworksProgrammerGuide.rar] - vxworks初学者的重要书籍，为其系列中的一个
[h.rar] - vxworks 6.x 的全部头文件，包含CAN

[华为技术手册.rar] - 华为工程师手册，在软硬件方面都有开发规范说明，是一份相当好的开发参考手册
[vxwork_src.rar] - 大名鼎鼎的嵌入式操作系统vxworks的完整的源代码，支持多种体系结构的嵌入式处理器，如arm, x86, i960, mc68k, mips, ppc, sparc等，包含完整的实时多任务处理及网络tcpip, dhcp, rip等协议，tffs文件系统，以及

请登录

帐 号: 注册帐号
密 码: 找回密码
登 录

VxWorks
相关类别

- bcm 4704系统启动函数都在这个文件
- bcm4704芯片的shell入口函数，从vxs
- bcm4704 SOC底层DMA的实现方式
- bcm4704 芯片的硬件探索机制，这个
- bcm4704芯片硬件抽象层具体实现方式
- 有资料大家共享。。。努力。。。努
- motorola mv5500 vxWorks bsp
- RTOS qnx Scheduling task
- 大名鼎鼎的嵌入式操作系统vxworks的
- 该学习文档对嵌入式操作系统Vxworks
- 50篇嵌入式及VxWorks应用论文下载
- s3c2410的vxworks bsp,运行良好，有
- 《嵌入式实时操作系统VxWorks及其开
- vxworks视频教程，讲解了如何使用t
- VxWorks开发实践,包括: 《Tornado
- vxworks下的驱动程序开发教程

useful documentation

www.pudn.com > vxWorks_sourcecode.rar > rpccore.c, change:1998-03-25,size:63256b

Search codes

```
01. /* rpcCore.c - Remote Procedure Call (RPC) backend library */
02.
03. /* Copyright 1995-1998 Wind River Systems, Inc. */
04. #include "copyright_wrs.h"
05.
06. /*
07. modification history
08. --------------------
09. 01o,24mar98,dbt  added support for WDB_CONTEXT_STATUS_GET service.
10. 01n,05mar98,c_c  Added bkendDsaEnable routine.
11. 01m,02mar98,pcn  WTX 2: finish to print the banner log file in rpcCoreInit.
12. 01l,21jan98,c_c  DLLized Target Server implementation.
13.          Removed EXT_FUNC refs.
14. 01k,22oct96,elp  replaced restartTargetServer setting by a call to TGT_RESTART()
15.          (SPR# 6943).
16. 01j,05jul96,p_m  added tgtOps.tgtModeGetRtn init (SPR# 6200).
17. 01i,28jun96,c_s  just include <fcntl.h>; this is more portable.
18. 01h,14jun96,elp  RPC_PROG_UNAVAILABLE means that target is connected to another
19.          tgtsvr (SPR# 4898).
20. 01g,21mar96,pad  now rpcCoreClntCall() clears the evtPending flag when no more
21.          events are pending an rpcCoreEvtPendingClear() has been made
22.          a no-op routine (SPR #6203).
23. 01f,16jan96,elp  WIN32 DLL external functions are called through pointers.
24. 01e,04jan96,p_m  passed RPC error code to bkendLog() (SPR# 4629).
25.          added missing WIN32 history.
26.          cleanup rpcCoreModeSet().
27. 01d,09sep95,wmd  added call to RestartTargetServer() for WIN32.
28. 01c,01sep95,p_m  took care of MTU in rpcCoreVIOWrite() (SPR# 4778).
29. 01b,08jun95,tpr  added WDB_ERR_NO_AGENT_PROC error code, RPC_CANTDECODERES and
30.          RPC_CANTDECODEARGS in rpcCoreClntCall().
31.          reworked seqNumber and added getpid().
32. 01a,31may95,tpr  derived from wdbrpc.c version 02k.
33. */
34.
35. /*
```

vxworks debug service

Metasploit modules

- Created a WDBRPC protocol library
- Created an easy-to-call Mixin
- Wrote modules
  - ❑ **wdbrpc_version**
  - ❑ **wdbrpc_bootline**
  - ❑ **wdbrpc_memory_dump**
  - ❑ **wdbrpc_reboot**

vxworks debug service

DEMO

vxworks debug service

Identifying affected devices

- At least 5 different vendors had flubbed this
- Probably much more where that came from
- Email the vendors and ask?
- Ask Wind River Systems?

vxworks debug service

This is 2010

- Just survey the entire Internet
- Use wdbrpc_bootline as a scanner
- Use tcpdump to capture replies
- Use a VPS with a friendly provider
- Scan, scan, scan!
- Parse the results

vxworks debug service

Preliminary results

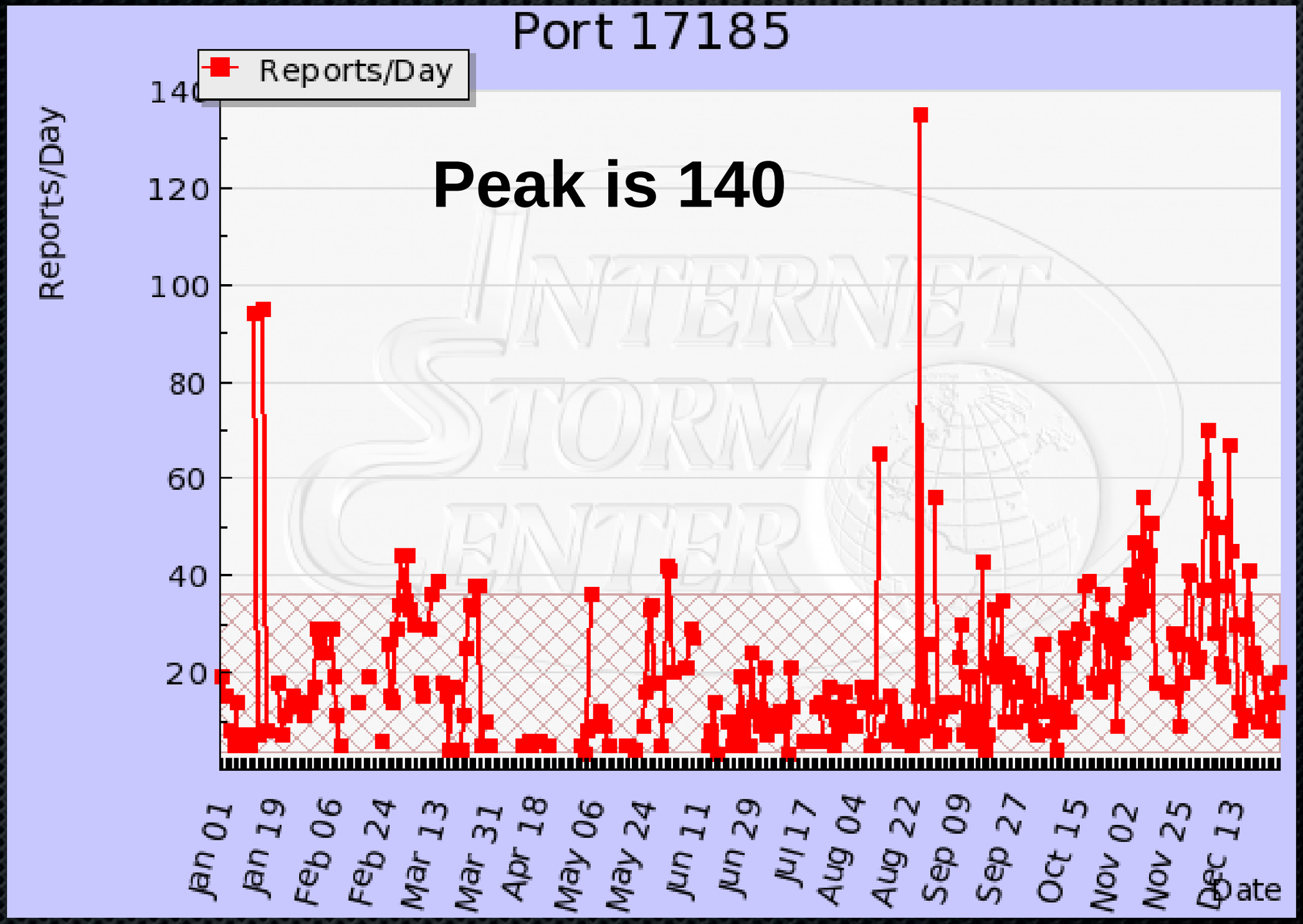
- Scanned 3,185,049,600 IP addresses
- Found over 250,000 vulnerable
- Rescanned those with SNMP
- Organized the results
- SNMP on 25%

vxworks debug service

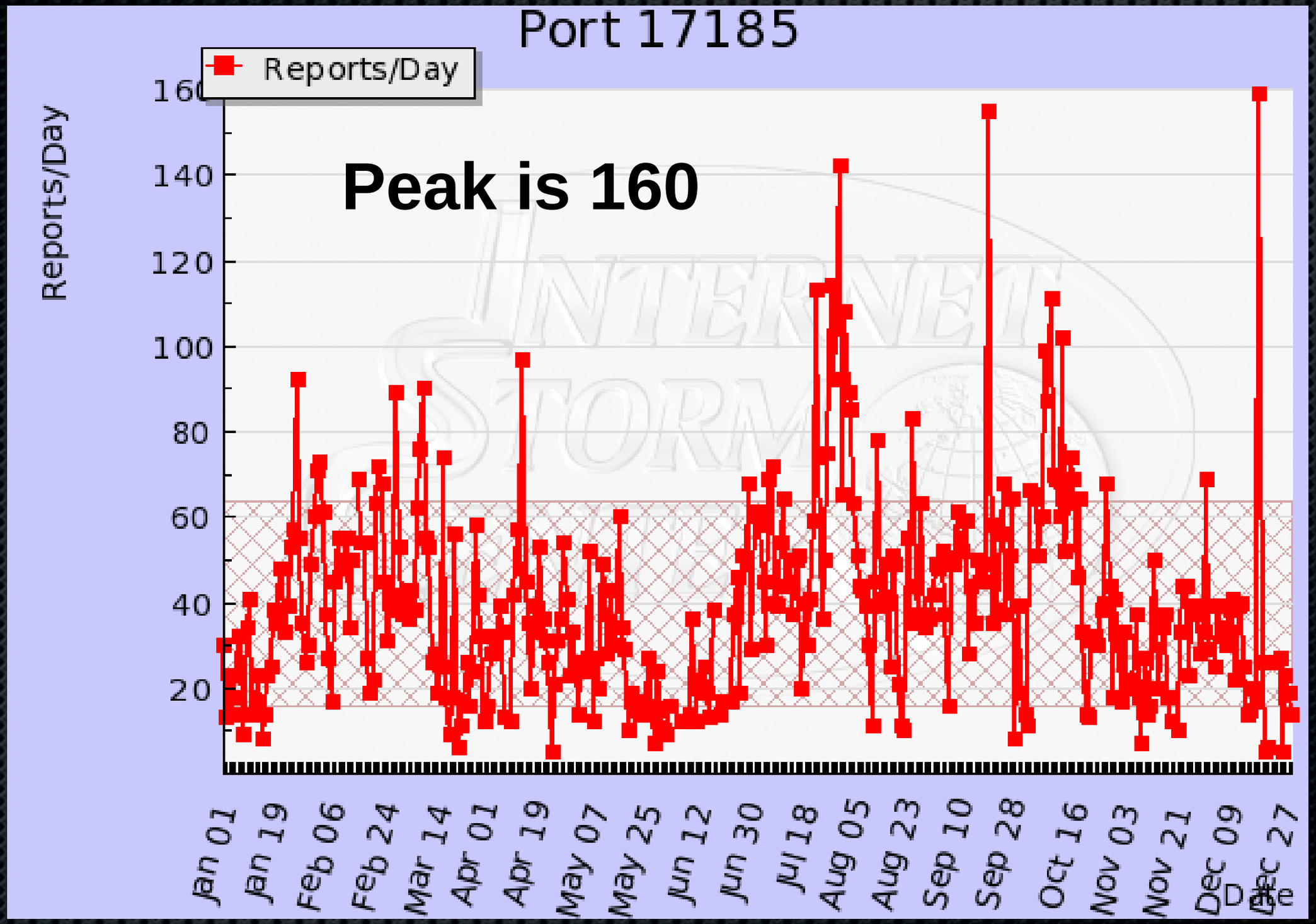
Checking score

- Someone must have noticed this scan
- Lets look through the DShield data...

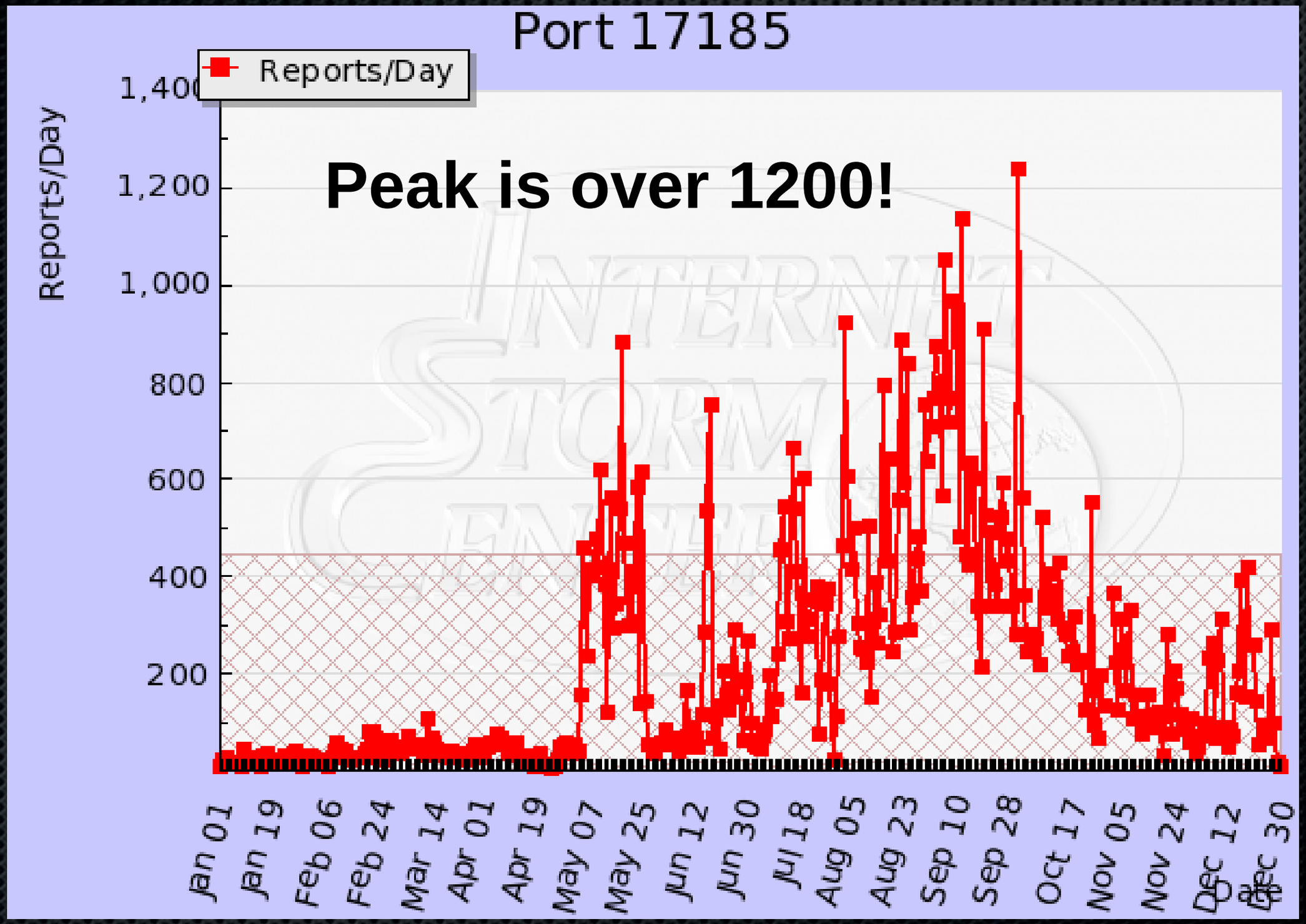
dshield: 2004



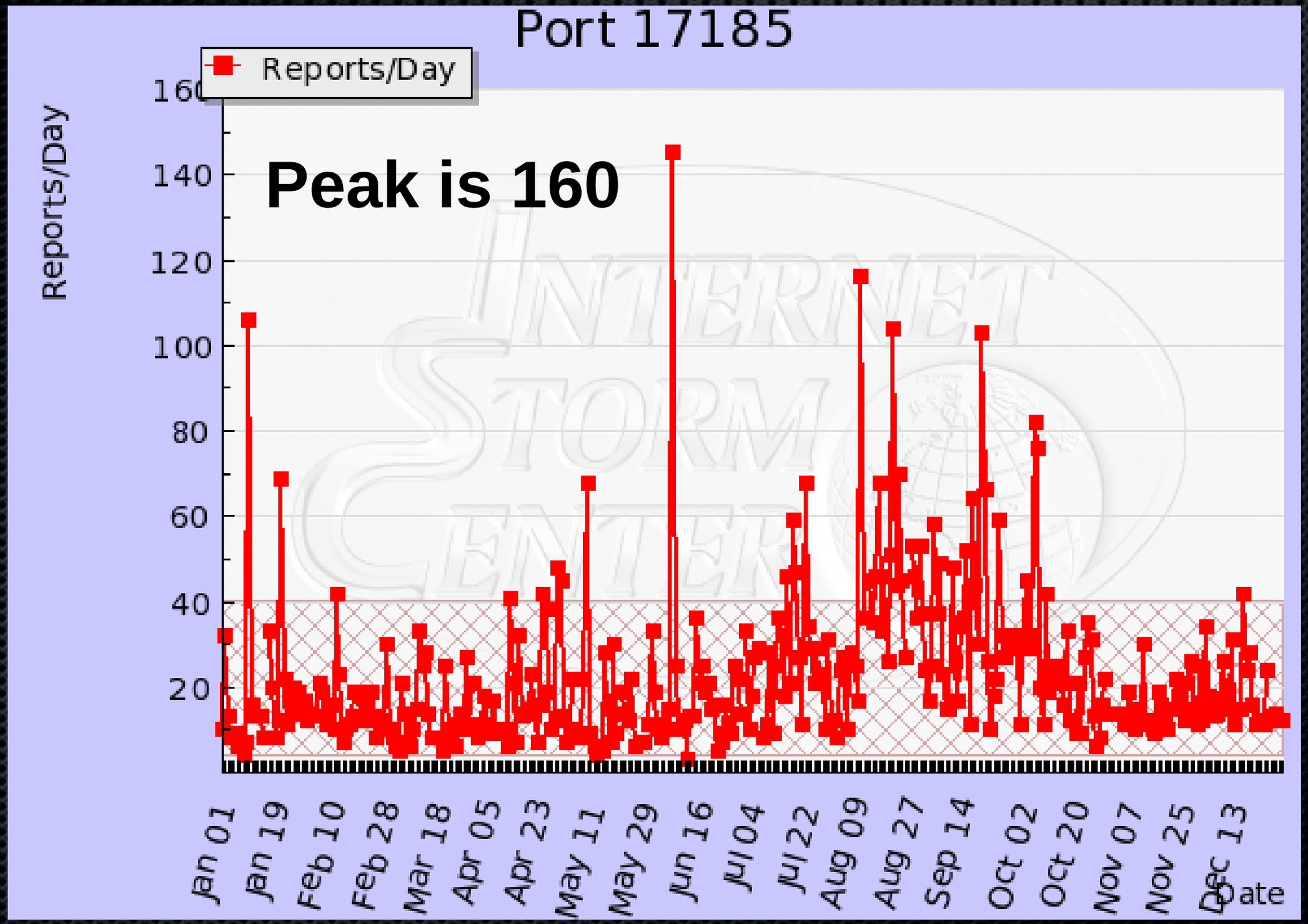
dshield: 2005



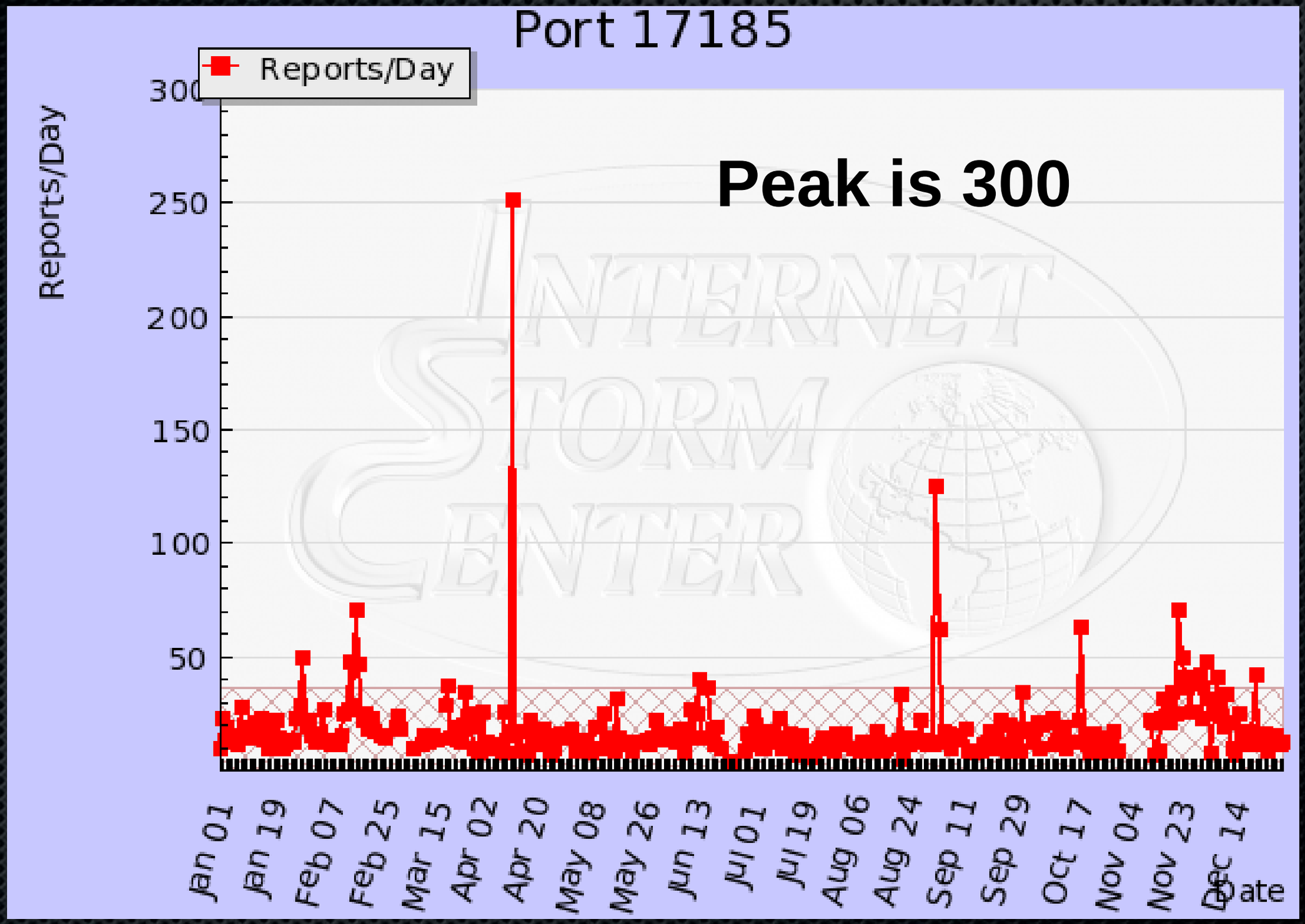
dshield: 2006



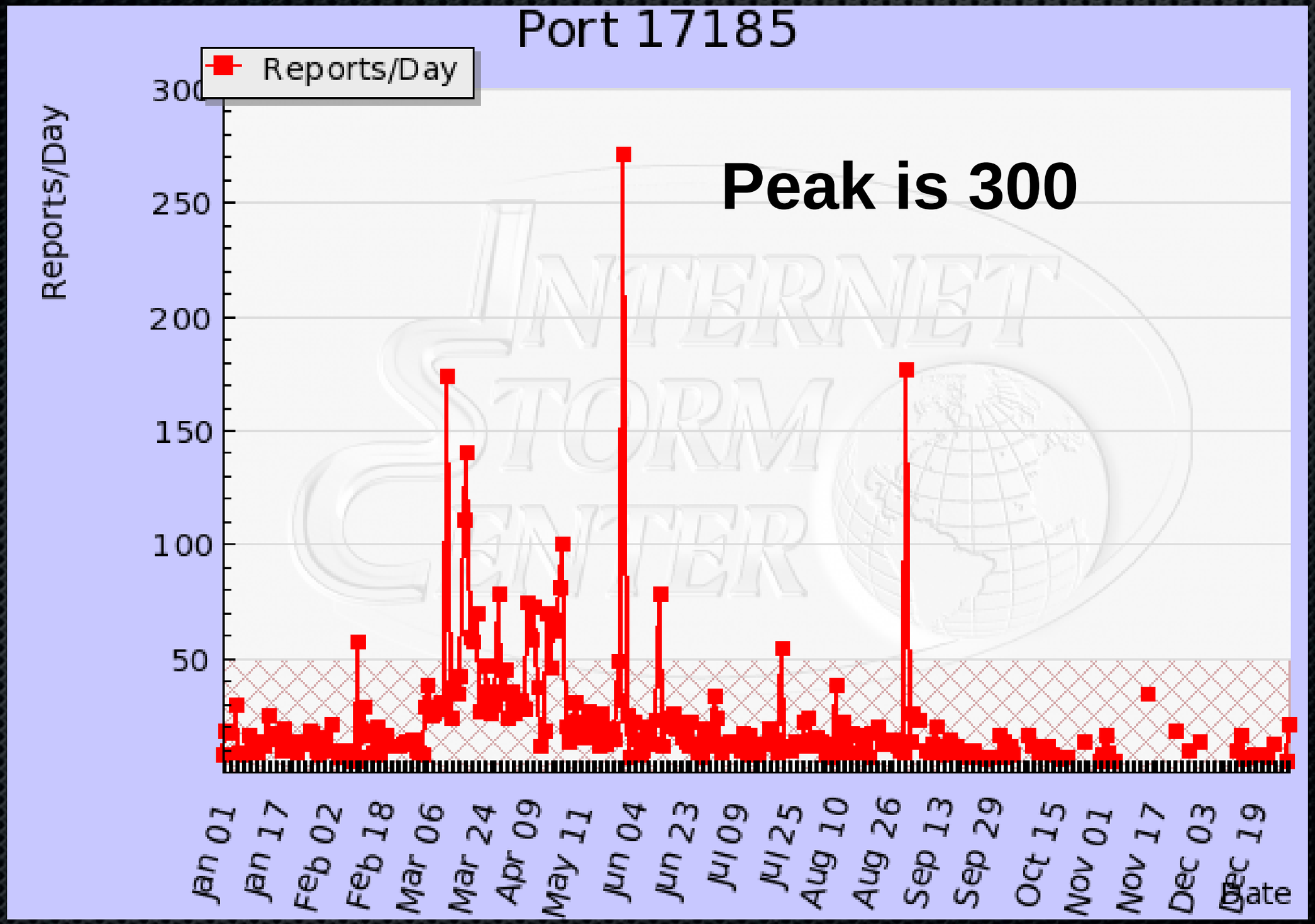
dshield: 2007



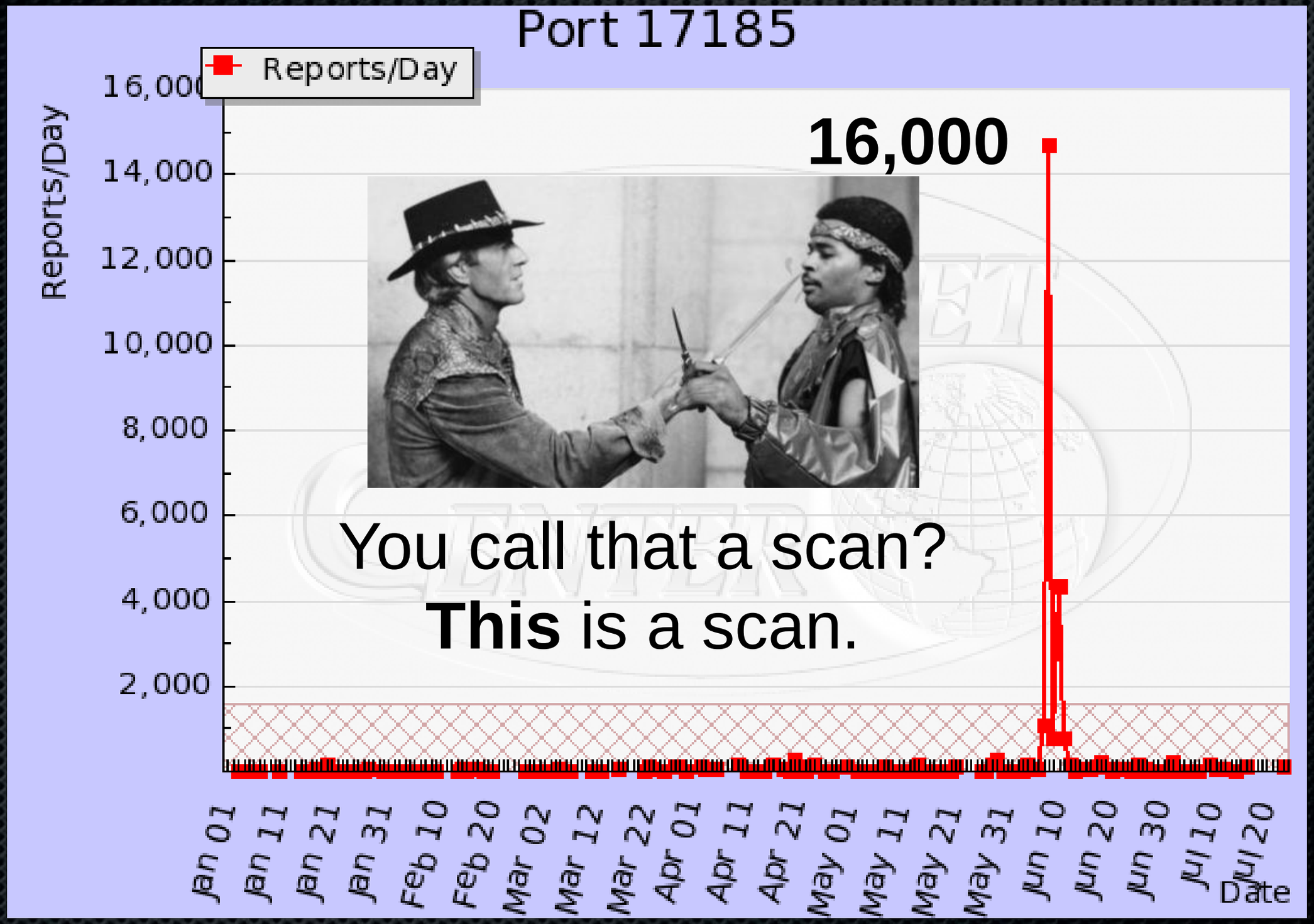
dshield: 2008



dshield: 2009



dshield: 2010



too late, we lost

Winning the internet

- Someone spent a year scanning for these
- This was 4 years ago, nobody noticed

shiny fun things

Exploiting the debug service

- We can read, write, exec memory
- We can reboot the device
- What code should we execute?
- How do we get a shell?

exploiting functionality

Save-game hacking

- Take a memory snapshot of the device
- Make a configuration change
- Take another memory snapshot
- Diff the results
- Patch bytes

exploiting functionality

DEMO – DVC1000

Product has been discontinued

exploiting functionality

Memory scraping

- Locate sensitive information in memory
- Write a “scanner” to find it

exploiting functionality

DEMO – Apple Airport

Latest firmware is patched

advisories

Advisories out August 2nd

- List of affected products and vendors
- Detection code in NeXpose & Metasploit
- No specific exploits until September 2nd

exploiting functionality

Changing the device mode

- Modify the boot flags in memory
- Soft reset the device
- Login remotely

exploiting functionality

Huawei IAD2 boot flags:

0x02 - load local system symbols
0x04 - don't autoboot
0x08 - quick autoboot (no countdown)
0x20 - disable login security
0x40 - use bootp to get boot parameters
0x80 - use tftp to get boot image
0x100 - use proxy arp

exploiting functionality

```
d1n@book: ~
File  Edit  View  Terminal  Help
/home/d1n/Desktop/[illegible] before_boot_flag_change.mem
0000 41D0: 00 00 00 00 00 00 10 00  00 00 42 C0 00 00 42 D8  ........ ..B...B.
0000 41E0: 00 00 42 78 00 00 00 0A  00 00 00 00 00 BC 00 00  ..Bx.... ........
0000 41F0: 00 00 10 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
0000 4200: 63 70 6D 28 30 2C 30 29  68 6F 73 74 3A 76 78 57  cpm(0,0) host:vxW
0000 4210: 6F 72 6B 73 2E 73 74 20  65 3D 36 31 2E 32 33 32  orks.st  e=[illegible]
0000 4220: 2E 31 31 2E 38 35 20 68  3D 30 2E 30 2E 30 2E 30  .[illegible] h =0.0.0.0
0000 4230: 20 67 3D 36 31 2E 32 33  32 2E 31 31 2E 38 31 20   g=[illegible] 2[illegible]1
0000 4240: 75 3D 63 73 70 20 70 77  3D 63 73 70 20 66 3D 30  u=csp pw =csp f=0
0000 4250: 78 34 30 20 74 6E 3D 49  41 44 00 80 00 00 00 00  x40 tn=I AD......
/home/d1n/Desktop/[illegible] after_boot_flag_change.mem
0000 41D0: 00 00 00 00 00 00 10 00  00 00 42 C0 00 00 42 D8  ........ ..B...B.
0000 41E0: 00 00 42 78 00 00 00 0A  00 00 00 00 00 BC 00 00  ..Bx.... ........
0000 41F0: 00 00 10 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
0000 4200: 63 70 6D 28 30 2C 30 29  68 6F 73 74 3A 76 78 57  cpm(0,0) host:vxW
0000 4210: 6F 72 6B 73 2E 73 74 20  65 3D 36 31 2E 32 33 32  orks.st  e=[illegible]
0000 4220: 2E 31 31 2E 38 35 20 68  3D 30 2E 30 2E 30 2E 30  [illegible]5 h =0.0.0.0
0000 4230: 20 67 3D 36 31 2E 32 33  32 2E 31 31 2E 38 31 20   g=[illegible]
0000 4240: 75 3D 63 73 70 20 70 77  3D 63 73 70 20 66 3D 30  u=csp pw =csp f=0
0000 4250: 78 32 30 20 74 6E 3D 49  41 44 00 80 00 00 00 00  x20 tn=I AD......

 Arrow keys move   F find        RET next difference  ESC quit   T move top
 C ASCII/EBCDIC    E edit file    G goto position      Q quit     B move bottom
```

vulnerable systems

Vendors & Devices

#define INCLUDE_WDB

authentication

Getting a shell (quickly)

- Dug into the login process for Telnet & FTP
- The password is hashed, hashes compared
- Tons of static backdoor accounts*
- Password is stored hashed...

```
#ifdef  INCLUDE_SECURITY
#define LOGIN_USER_NAME          "target"
#define LOGIN_PASSWORD           "RcQbRbzRyc"    /* "password" */
#endif  /* INCLUDE_SECURITY */
```

* Check for calls to loginUserAdd()

authentication

Math is hard (apparently)

- The algorithm is indexed in Google
- Used an additive byte sum as the “secret”
- Only 210,000 possible output hashes
- Only ~8,000 are easy to type
- Most passwords within ~4000
- Range is 8-40 characters, \x00 -> \xFF

authentication

Hash output examples

- “password” > 3974 / RcQbRbzRyc
- “passwore” > 3966 / RRc9dydebz
- “howdybob” > 3847 / ReySzQQSRR
- “AAAAAAAA” > 2304 / Rrdeebbe
- “!@$%^WTF” > 2564 / b9SdezeRcb

authentication

Precomputed passwords

- Calculated a “workalike” for all outputs
- Sorted by probability of it working
- Plug this into Metasploit bruteforce

authentication

Brute force is easy

- No account lockouts by default
- Telnet disconnects after 3 attempts
- FTP never disconnects
- FTP allows 4 connections
- Crack most passwords in ~30 minutes

authentication

Combine debug + weak hashes

- Remote memory dump a target device
- Scan the memory dump for hashes
- Find the username as well
- Login!

vxworks

Summary

- These bugs are just the tip of the iceberg
- Metasploit code will drive research
- Expect to see these for a long, long time

Timeline

- Public advisories on August 2nd
- Rapid7 NeXpose checks on August 2nd
- Metasploit scanners on August 2nd
- Exploit modules pushed in early September
- Master password list also in September

vxworks

References

- VU#362332 - http://www.kb.cert.org/vuls/id/362332
- VU#840249 - http://www.kb.cert.org/vuls/id/840249

- http://www.metasploit.com/redmine/projects/framework/wiki/VxWorks
- http://www.rapid7.com/vulndb/lookup/vxworks-wdbrpc-exposed