to know how we can manage stonehand appropriately so we don't have another contract extension.

########################################################## 1609
From sprocket_mailer_daemon Tue Sep 3 08:07:13 1991
To: bradsi
Subject: mail problem

Trouble sending mail on `sprocket', Tue Sep 3 08:07:13 1991

========== Transcript follows ==========
alias file error: /usr/lib/mail/aliases.hash
jonro richt t-scotsa
1 alias errors
No local user named "jonro"
No local user named "richt"
No local user named "t-scotsa"
3 delivery errors
4 total errors

========== Message follows ==========
>From bradsi Tue Sep 3 08:07:13 1991
To: t-scotsa
Cc: jonro richt
Subject: Re: Discussion Group results this week
Date: Tue, 03 Sep 91 08:02:46 PDT

thanks scott. the next machine visits sounded interesting.

last fall I bought a few Next machines. after the initial interest, they just lay unused. problem is there aren't many apps for Next, and all the major sw companies have killed their on-going next development.

| * Apps seen to drive operating system success.

tell me more about mm integration into nextstep.

########################################################## 1610
From sprocket_mailer_daemon Tue Sep 3 08:26:18 1991
To: bradsi
Subject: mail problem

Trouble sending mail on `sprocket', Tue Sep 3 08:26:18 1991

========== Transcript follows ==========
alias file error: /usr/lib/mail/aliases.hash
jonro
1 alias errors
No local user named "jonro"
1 delivery errors
2 total errors

========== Message follows ==========
>From bradsi Tue Sep 3 08:26:18 1991
To: jonro
Subject: Re: Winmktg Weekly Status Report
Date: Tue, 03 Sep 91 08:21:49 PDT

thanks, jon.

i don't want alex to move. we need someone doing the oem thing in our group.

########################################################## 1611
From dennisad Tue Sep 3 08:30:55 1991
To: winprog
Cc: bradsi davidw johnen philba
Subject: Re: Terminator
Date: Tue, 03 Sep 91 08:30:03 PDT

In case you never heard the details on this PM crasher/basher app...

>From georgem Fri Aug 30 18:18:46 1991
To: dennisad
Subject: Re: Terminator

This is the app that MS used to prove that OS/2 was as easy to crash as Windows.

> From ericfo Tue Aug 13 13:03:03 1991
> To: petes
> Cc: bohmu markcl
> Subject: Re: Terminator
> Date: Tue Aug 13 13:06:48 1991
>
> I'm the author, you can see it anytime. I'm not here this week, but
> hope that next week would be okay...
>
> I do not release the sources or binary. Certainly willing to discuss
> what it does in gory details...

I have tried calling several times with no luck, so I figured I'd try email. I just have a few questions.

(1) What is terminator's basic strategy?
(2) What means does it use to go about it?
(3) Is it "realistic", i.e., is there a significant chance that a buggy application would cause the same effect?

Lost your last email in my overflowing email.

(1) What is terminator's basic strategy?
(2) What means does it use to go about it?

Details below...

(3) Is it "realistic", i.e., is there a significant chance that a buggy application would cause the same effect?

It is realistic. Direct hardware access illustrates that OS/2 2.0 cannot claim to be a secure system. It is inappropriate for many mission critical apps. OS/2 is architected for efficiency which defeats many benefits of process protection. Bad apps can crash the system, parameter validation is not complete, buggy apps are a problem. OS/2 is better protected than Windows, but cannot claim to be secure. NT is secure, OS/2 is on the other end of the spectrum with Windows.

Abstract

The OS/2 and Windows operating systems share certain architectural designs which are often overlooked when robustness comparisons are made. The process protection that OS/2 applications are provided creates a level of protection not found in Windows such that errant applications are less likely to corrupt other applications or the system itself. However, OS/2 negates specific aspects of this protection in order to support applications that directly access hardware, file system and other applications.

These architectural features exclude OS/2 from being considered for U.S. government security approval and many mission critical installations. Windows/NT provides a newly designed robust and secure kernel designed >from the start for C and B level security.

Description

The Terminator III application (TIII) is useful for visualizing areas where OS/2 provides insufficient robustness. TIII has identical affect on retail shipping OS/2 1.21 and 1.3 as well as the latest OS/2 2.0 Beta release. TIII does not exploit beta-quality code; in fact there are no version checks in TIII. The code runs identically on all OS/2 systems.

TIII makes no use of internal knowledge of OS/2 operating system "holes" or back doors. The techniques that are used in TIII will be familiar to most OS/2 programers.

TIII demonstrates the following OS/2 characteristics:

- o Incomplete parameter validation causing system and application data corruption
- o Deficiency of the PM synchronous messaging model
- o Vulnerablity of OS/2 to direct and/or incorrect hardware access
- o Wild pointers that corrupt system causing data corruption and system crashes

TIII only highlights a few of the OS/2 architecture design limitations. Features such as spurious system rebooting, random application GP faults (except TIII), file system corruption, and overuse of critical system wide resources (even in 32-bit OS/2 2.0) could also be demonstrated.

## Usage

It is recommended that the OS/2 system being used to demo TIII is fully backed up. Save all data in currently running applications. Shut down important applications before running TIII.

1)
- o Start several PM apps such as the system editor (E.EXE)
- o Start TIII (BADAPP.EXE)

2)
- o Bring up a file into the E editor
- o Select TIII menu option: Using an Invalid PM Handle
  - All titlebars become corrupted
  - Data in the E editor is corrupted
  - More data is corrupted in E editor as the data is scrolled
  - Select Save As... menu in E editor
  - Note that all edit controls in dialog box are corrupt
  - Select items in list boxes and note corruption
- o Reselecting this menu option disables continued data corruption

3)
- o Select TIII menu option: Improper Keyboard Handling
  - Random characters appear in some applications
  - Change focus to different applications to see affect
- o Reselecting this menu option disables continued data corruption

4)
- o Select TIII menu option: Message Processing Bottleneck
  - System appears hung for 30 seconds
  - Illustrates that OS/2 employs cooperative message based multitasking just like Windows

5)
- o Select TIII menu option: Hardware Access
  - Only works on EGA/VGA display hardware
  - Affects horizontal video sync

6)
- o Select TIII menu option: Wild Pointer
  - Every 0.5 seconds, 20 random bytes of system global shared data is corrupted
  - System will crash within a few minutes
  - Often very visual if menus are displayed, switch between running applications, min/max running applications, etc.

## Technical Details

TIII is a 16-bit OS/2 application which runs at ring3 but includes a ring2 code segment for directly accessing hardware. Ring2 code is allowed to directly access I/O ports even on OS/2 2.0 (for compatibility with existing 16-bit OS/2 1.x apps and dlls).

Invalid window handles are used to corrupt titlebar, edit control, multiline edit control, and list box data.

To bottleneck the messaging system, a simple DosSleep() call was placed in the message handling procedure. The affect to a user is that the system is hung. OS/2 requires cooperative multitasking just like Windows in the PM interface negating many of the benefits of the underlying pre-emptive multitasking OS/2 base.

OS/2 maps shared DLL data segments and shared allocated memory across all process address spaces. This defeats a major benefit of process

protection and separate address spaces. A wild pointer can corrupt data not intended for use by the errant application, such as critical system data (linked lists, pointers, resources, etc.) or other application data.

## The Windows/NT Solution

Windows/NT does not derive from the OS/2 2.0 architecture and the above techniques if placed in an application running on Windows/NT would cause an exception and the application to be tossed out of the system.

## Safety Features

The following safety features were added to assure that TIII would have limited affect on unsuspecting users if unauthorized copies are distributed.

TIII employs an expiration feature. The executable no longer will corrupt data after 8/31/1991. After the expiration date, the application executes but none of the options function.

It is necessary to specify IOPL=YES or IOPL=BADAPP in the CONFIG.SYS file. If this is not added, TIII will not run. TIII could have been written such that the IOPL statement would be unnecessary.

When running the Message Processing Bottleneck option, it is possible to use CTRL-ESC to bring up the Kill Application dialog box. It is possible to disable this feature so that it is impossible to kill TIII.

The source code will not be distributed. It is not desirable to motivate additional interest or competition in creating this type of application.

## Distribution

TIII may not be distributed, duplicated, or demonstrated without the written consent of the author. If you do not know who the author is, then you cannot distribute, duplicate or demonstrate TIII.

## Reverse Engineering

It is a violation to disassemble, reverse engineer or modify the TIII application.

## Responsibility

Responsibility for using the TIII application resides solely with the person using the application. TIII will cause data corruption in any executing application. It is remotely possible that TIII will cause unrecoverable hardware failure; this was not the design intent, but video and/or system hardware failure is possible when using the Hardware Access option.

>From ericfo Thu Aug 29 15:32:04 1991
To: petes
Subject: Re: Terminator
Date: Thu Aug 29 15:46:22 1991

> >From petes Thu Aug 29 13:40:30 1991
> To: ericfo
> Cc: petes
> Subject: Re: Terminator
> Date: Thu Aug 29 13:39:58 1991
>
> Great! Now I have some more specific questions...
>
> > 6)
> > o Select TIII menu option: Wild Pointer
> > - Every 0.5 seconds, 20 random bytes
> > of system global shared data is corrupted
> > - System will crash within a few minutes
> > - Often very visual if menus are displayed,
> > switch between running applications, min/max
> > running applications, etc.
>
> This is the really interesting one. How does TIII generate

> its wild pointers?

Ah, you're asking that I give away all my secrets. The key point is that shared memory is mapped across all processes and there is quite a bit of it. Wild pointers can crash (but more often corrupt) other app shared data, system data, etc.

I hunt down PMWIN shared memory to dramatize in a demo. It is simple, but you have to buy me a beer (or two) to confess...

> Another general question: what percentage of OS/2 apps contain their own ring 2 code? Is this something that is commonly done by applications?

Probably a similar percentage to Windows apps that wish to directly read/write fax cards, scanners, weird cards, scientific instruments, etc. and never wanted to be troubled with a device driver.

This was a large enough percentage under Windows that we were very concerned about breaking apps with Porthole.

Eric

########################################################## 1612
From johnlu Tue Sep 3 08:38:04 1991
To: bradsi
Subject: quick winball status
Date: Tue Sep 03 08:37:33 PDT 1991

as you requested, here is a quick status on winball:

VxD Redir, VxD NetBEUI, VxD File Server (real mode file systems)

- all these are up and running on the teams' machines. we are using the redir and netbeui every day for our normal work. The server is still a little experimental, but I expect that we will start using it regularly next week.

  we'll begin to put these pieces on a few peoples desks outside the group next week. There is still a lot of cleanup work to do on these components — NetBIOS apps don't work yet, SHARE is broken with our server, etc. We'll be working on the server in particular for at least another 2 months to close all these issues.

VxD FAT File System (GordonL)

- up and running for most operations. Some work to be done on directory operations. We will start to tie this into the server next week, and will probably have a VxD file system server within a couple weeks for experimental use.

Print Server

- Work has begun. We've had to add a prespooler to the system, which accepts net jobs and queues them up for submission to the Windows print spooler. We're just about done with the prespooler, and will probably move next week to remote job submission — ie letting remote clients connect to and submit jobs to the prespooler.

Setup

- Work has begun. We have a few components done, notably a net card detection DLL that works on the market leading net cards. We're working now to port the Windows 3.1 installer to enhanced mode. I don't expect us to have a usable setup program for at least a month.

Net DDE