

Re: Honeypot stats

Source: <http://unix.derkeiler.com/Newsgroups/comp.os.vms/2006-01/msg00536.html>

- *From:* "Neil Rieck" <n.rieck@xxxxxxxxxxxx>
 - *Date:* Wed, 11 Jan 2006 19:03:06 -0500
-

"William Webb" <william.w.webb@xxxxxxxxxx> wrote in message
<news:8660a3a10601111249p7507acd1rc19f52a343055289@xxxxxxxxxxxxxxxxxxxx>

On 1/11/06, JF Mezei <jfmezei.spamnot@xxxxxxxxxxxx> wrote:

> Neil Rieck wrote:

>

>> I think you'd agree that "if the addition of new applications exposes
>> security problems, then the OS wasn't very secure in the first place".

>

[...snip...]

>I'm looking at an article in SC Magazine entitled "Security starts
>with coding" which contains the following paragraph:

>

>"Many of the techniques for secure coding have been left out of
>courses for software developers. Without proper knowledge of how to
>build secure software, programmers run the risk of jeopardizing
>development projects."

>

>Or entire OSES for that matter.

>

I think you just proved my point (sort of). Of course it is the best of both worlds when all programmers from the application level all the way up to the OS level concern themselves with security. My point being that if an inexperienced programmer makes a mistake, then thank heavens a good OS like OpenVMS is there to act like a security/safety net.

Now I'd like to add a personal comment to your second paragraph; I wonder if this is related to the difference between "computer sciences in an academic sense" and "computer engineering"?

<timetravel>

In the early 1990's I did some freelance design work on an embedded controller system based upon the Motorola 68HC11F1. We manufactured a few hundred of these and although the h/w work was interesting, the firmware work was even more so. I wrote my own monitor as well as all the interrupt-driven device drivers and everything worked an apparent 99.9% of the time. Well I'm sure you can imagine what happened next? My client came

Re: Honeypot stats

back and asked me to improve upon the 99.9%. So I got in touch with some design people in Detroit who were doing this sort of thing in the hostile environment of automobiles. One kind soul sent me a photocopy of an article from "Embedded Magazine" (I think) that really blew my socks off. One big problem in embedded design is EMI. Imagine a JUMP or BRANCH statement that is executing at the time EMI flips one of your address lines: you end up in a different section of the ROM. The article suggested all kinds of cool things but here is just one example:

Setting aside a special counter (or flag if memory is scarce) for every subroutine; you increment the counter upon routine entry and then decrement the same one upon routine exit. On each pass through the monitor you inspect all the counters and they should all be set to their initial values. If your CPU accidentally jumped from one routine to another, the counter associated with the first routine would be too high while the counter associated with the second routine would be too low. Your monitor might be able to record and/or recover from this but without this kind of defensive programming you might never know. Also, this kind of fault could accumulate for days before you ever saw a stack error that forced a hard reset.

p.s. the Automotive guy told me that some of his defensive programming techniques came from talking to a defence contractor software developer who had moved into commercial space applications. (now I know why my GrandAm never craps out)

Now you are not going to get this kind of information learning how to write an accounting program at community college. IMHO this much rarer stuff is going to be passed from person to person in a specialized engineering environment (like Yoda talking to Luke Skywalker). I wonder what the folks at JPL are doing right now?

</timetravel>

So I said all of the above to say this: programming with security in mind must be a totally different experience.

Neil Rieck
Kitchener/Waterloo/Cambridge,
Ontario, Canada.
http://www3.sympatico.ca/n.rieck/links/cool_openvms.html

-
- *Follow-Ups:*
 - ◆ **Re: Honeypot stats**
 - ◇ From: Paul Sture

- *References:*
 - ◆ **Re: Honeypot stats**

Re: Honeypot stats

◇ *From:* William Webb

- Prev by Date: ***Re: sho proc/mem***
- Next by Date: ***Re: OT: time to market with the 8086***
- Previous by thread: ***Re: Honeypot stats***
- Next by thread: ***Re: Honeypot stats***
- Index(es):
 - ◆ ***Date***
 - ◆ ***Thread***