

Re: IP Logging in the Security Event log

Source: <http://www.derkeiler.com/Newsgroups/microsoft.public.win2000.security/2002-11/10991.html>

From: Eric Fitzgerald [MS] (ericf@online.microsoft.com)

Date: 11/05/02

From: "Eric Fitzgerald [MS]" <ericf@online.microsoft.com>

Date: Tue, 5 Nov 2002 12:47:46 -0800

> *Not knowing what needs to take place in the Event*
> *log/network areas to make this happen, I still can't see*
> *it being to big a deal. Obviously somewhere in the OSI*
> *Windows 2k knows what the IP of the incomming request is,*
> *it's just a matter of pulling it from whatever layer that*
> *information is being kept at and dragging it up to the*
> *application layer to be inserted into the Event log*
> *message.*

Hi Robert,

Most people haven't got the slightest idea at what it takes to make this kind of change and back-port it, so I'll give you some insight into the process of adding this into Windows.

Our security system is network-independent. Typically authentication works like this:

1. An application (typically a service) receives network requests in its own manner (IIS uses HTTP, LanManServer uses CIFS/SMB, RPCService uses MSRPC, etc.). The protocol, port, and encoding are completely at the application's discretion. Negotiation of auth protocol and transfer of credentials are usually protocol-specific.
2. The application calls (Lsa)LogonUser() or AcceptSecurityContext(Ex)() with the supplied credentials.
3. The LSA validates the supplied credentials (either passed to a DC [NTLM/domain] or validated locally [NTLM/Local SAM or Kerberos Service Ticket]). The appropriate audit is generated by the LSA. Note that the LSA generates the audit because LSA is a trusted system service, part of the TCB in C2 parlance. We don't trust applications to generate their own audits.
4. The call returns to the calling app with success or failure and other information necessary to use the new logon session.

There were a number of problems when we decided to add this to Windows .NET Server:

None of the LogonUser or AcceptSecurityFunctions had an extra parameter we could use to transfer socket information such as source IP address and source port. We needed a dedicated buffer that was large enough to hold an IPv6 address. The only thing we could even consider was the machine name field, and we couldn't use that for compatibility reasons and because the buffer might not be large enough in some cases (NetBIOS names are only 16 bytes long; we needed 18 minimum and 32 to be safe, engineering for the future). So we had to figure out a different way to get the information to the LSA via these calls. We could not modify any of these APIs to supply the additional information; they are too well-entrenched and changing the calls would break applications.

Not only that, but we had to support both user-mode and kernel-mode, and the kernel-mode call was further restricted by the limited space available in the LPC call that is used to communicate from NTOSKRNL to LSASS.EXE (128 bytes, most of which was already in use). Furthermore, our calls had to be thread-safe.

So we ended up coming up with a rather elegant solution involving a second API call, but it took weeks of discussion and design to get it right, approval from a Vice President to make the change to Windows .NET Server late in the product cycle, and then a couple of weeks to code it up and test, before we checked it in.

Lastly, I then had to approach each major team in Windows that use our logon APIs, convince them of the value of participating, and have them modify their components to make sure that the IP address information was available in the general vicinity of their logon calls, and then to make our extra function calls to provide us with the address information. This involved coordinating dozens of people and half a dozen components.

Right now the code is checked in and running in production in Microsoft. However this is very "young" code; we have already found one bug in our stress testing. We don't like to check code into a Service Pack until it is mature.

And our service pack story is also more complex because (1) we don't like to check new features into a service pack and this straddles the line between new feature and plugging a repudiation hole, and (2) this code touches over a dozen system components, and would introduce a horrible dependency into the Service Pack tree. If this were checked in as-is, then any hotfix involving any of the components touched would have to include all the dependency files (about 40 or so). For instance, if you installed a hotfix for IIS after this fix were checked in, you'd actually have to patch the kernel and LSA and RPC and Active Directory and some other stuff. So we have to re-engineer the fix to back-port it to Windows 2000; we've begun discussions with the Service Pack team and think we have a workable design but SP5 is a long way off.

I hope this helps you understand. Making a change to a network operating system is a nontrivial task, especially when it involves sensitive or widely-used code, and especially when our hundreds of millions of customers expect stability and security. We can't afford to make a mistake while in a rush to add something new.

Eric

"Robert Minneman" <robertminneman@earthlink.net> wrote in message news:521601c28468\$1142c950\$37ef2ecf@TKMSFTNGXA13...

> >The auditing system was designed ~1990-1991, and IP
> addresses were not even
> >on the radar screen at that time.
>
> Ok, it was kind of obvious as this doesn't appear to have
> changed since the original version of NT came out (the
> fact that Win2k and OS/2 still share some of the same
> registry keys is a great indicator at how long it takes
> for some of this stuff to change).
>
> >There was some customer demand for this even in the Win2k
> time frame but for
> >various reasons we could not and did not attempt to make
> that change for
> >Windows 2000.
>
> I would have expected demand for this to be picking up
> around NT 4.0's release when the internet was just picking
> up and people were beginning to consider the concept
> of "firewalling" their networks, although I understand
> that there is a need to prioritize what gets on the
> development list.
>
> >We added IP address to logon audits in Windows .NET
> Server, and we're going
> >to attempt to back-port this to Service Pack 5 for
> Windows 2000 (not SP4).
>
> Not knowing what needs to take place in the Event
> log/network areas to make this happen, I still can't see
> it being to big a deal. Obviously somewhere in the OSI
> Windows 2k knows what the IP of the incoming request is,
> it's just a matter of pulling it from whatever layer that
> information is being kept at and dragging it up to the
> application layer to be inserted into the Event log
> message.
>
> Regardless, this is a great reason for me to get to SP5 as
> soon as it's released.
>
> Thank you very much for the information.

> *the*
>>> *box stupid" that has kept the "we hate Microsoft" crowd*
> *so*
>>> *full of ammo as to WHY everyone should hate Microsoft.*
>>>
>>> *Bah, I guess I'm just frustrated, I really don't want to*
>>> *have to have yet ANOTHER log to monitor on this box.*
>>>
>>> *Robert Minneman*
>>> *robertminneman@earthlink.net*
>>
>>
>>
>>.
>>