



# Effective Practice: Network Registration System Scanner

## Submitting Institution:

University of Connecticut

## Date Submitted:

1/12/2004

## Category:

Vulnerability Assessment

## Subject Terms:

Network Vulnerability Assessment , Vulnerability Scanning

## Background:

As of fall 2003, the University of Connecticut network consists of approximately 20,000 hosts, 11,000 of which are in the residential halls. Our commercial Internet link is about 200 Mbps, and the Internet2 link is 155 Mbps. We have been using NetReg as our MAC-based automatic host registration system since 2000. We use Nessus for scheduled and on-demand network vulnerability testing.

In late August 2003, we faced the prospect of 11,000 student computers being connected to an already taxed university network. We assumed that a large portion of these computers were not patched versus the MS03-026 vulnerability (RPC-DCOM). We needed a way to scan for vulnerable hosts as soon as they connected to the network, so users could patch their systems before they were infected with the Blaster or Welchia (Nachia) worms.

We had contributed to the refinement of Nessus plugin #11808, which detects hosts vulnerable to RPC-DCOM, and first tried to tie that into NetReg. We were unhappy with the poor speed and stability of launching so many Nessus scans, so we developed our own scanner that would work closely with NetReg. Using this combination, we were able to identify vulnerable computers as they first connected to our network and automatically direct them to the patch they needed without involving support staff.

## Description:

NetReg Scan has two main components. The first is `rpcscan`, which was developed in house by Keith Bessette and Mike Lang. The second is the NetReg Scan cgi page, which was developed by Josh Richard of the University of Minnesota-Duluth and Mike Lang.

`Rpcscan` is c code developed for Linux that contains the same basic functionality as Nessus plugin #11808, which is similar to EEye and ISS's free RPC-DCOM vulnerability scanners. It scans for hosts vulnerable to MS03-026 and MS03-039 (RPCSS) and returns either human-readable output or output that NetReg Scan understands. It can be used as a stand-alone CLI scanner on Linux, as opposed to the GUI and CLI tools released by Microsoft, ISS, and EEye that run only on Windows. Developed to be quick and light-weight, `rpcscan` is the fastest way we could scan Class B-sized networks.

NetReg Scan is a cgi page that is meant to be inserted "in front" of NetReg, usually after a splash page that directs users to follow a link. It initiates a prescan with `nmap`, then calls `rpcscan` if it detects a Windows computer (for example, open tcp ports 135). `Rpcscan`, in NetReg Scan mode, returns if the host is vulnerable to MS03-026/039 or if it has been properly patched. If the host is detected to be vulnerable, NetReg Scan redirects users to a support Web page that informs them what steps they need to take to patch their computers. If the host is not vulnerable, it forwards them on to NetReg, where they can register normally.

## Benefits:

`Rpcscan` runs on Linux, which is what many universities use for their network registration and vulnerability testing servers.

Scanning with rpcscan is much faster than scanning with Nessus, mostly since Nessus can do many more things than just scan for RPC-DCOM.

By immediately determining if the computer has been properly patched versus RPC-DCOM, NetReg Scan strongly encourages users to properly patch their computers before they are allowed past the registration system and onto the public network. We found that when users were prompted with clear instructions about how to properly patch their computers—and were denied access to the Internet until they did so—a large percentage of them took protective measures they would not ordinarily take. We automatically detected as vulnerable and handed out patches to about one-third of our student network, which was over 3,500 computers.

Using NetReg Scan kept the Blaster and Welchia infection rate on our student network to about five percent of all hosts. This, in turn, saved the residential network from becoming overburdened from worm traffic and unresponsive. While the specific technologies used in this instance will certainly change, the basic philosophy seems sound. Scan unknown hosts for major vulnerabilities before you allow them to connect to your public network, and give them an easy way to correct the problem themselves.

### Shortcomings:

The most obvious shortcoming of rpcscan is it only scans for two things: RPC-DCOM and RPCSS. There are no plans to modify it to keep up with additional Microsoft vulnerabilities. This is why we plan to switch back to using Nessus to initially scan for vulnerable hosts (see Future Plans).

Additionally, NetReg usually works by putting unregistered computers into a relatively large private network space (10.x), where it is possible that previously infected hosts will attempt to spread the worm to other computers in the same 10.x subnet. It is possible to make these private pools of addresses very small to decrease the chance of this happening. In practice, we did not see a large number of hosts become cross-infected in the private pre-registration space.

Many student computers came back to campus in the fall after being connected to relatively unmonitored home broadband networks. Some of these computers were already infected with the Blaster or Welchia worms before they connected to our network. Since Welchia patched the MS03-026 vulnerability, these systems could not be detected and quarantined using NetReg Scan. Our goal was to get those infected (and patched) computers through NetReg and into our public space as fast as possible. Since all student computers in the public space had been through NetReg Scan, they were nearly all patched and therefore immune to Blaster and Welchia.

### Future Plans:

We plan to return to Nessus as the scanning engine for NetReg Scan. We do not have the resources to develop additional scanners for future vulnerabilities, and the Nessus community usually does a great job. The problem of load and speed should be able to be solved with additional hardware, including the possibility of clustering together a number of scanning servers or by throttling the number of scanning jobs we submit at one time.

### References:

<http://www.netreg.org/>  
<http://www.nessus.org/>  
<http://security.uconn.edu/netregscan/>  
[http://security.uconn.edu/uconn\\_response.html](http://security.uconn.edu/uconn_response.html)

### Return on Investment:

Although rpcscan took many person-hours of time to develop and debug, we felt it was an excellent return on investment. The NetReg Scan cgi page was more straightforward and benefited from lots of feedback from the community. Because rpcscan was so lightweight it did not require any additional hardware, and it ran decently on our aging P2-based NetReg server. We successfully registered about 9,000 computers in the first two days students returned to campus, which seems to be an excellent rate when compared to commercial network registration systems.

We estimated about two hours of support time were needed to repair and patch up a computer after a Welchia or Blaster infection; by detecting 3,500 computers as vulnerable and automatically directing them to a patch, we may have saved 7,000 hours of support staff time.

