Flirting with SATAN

by
Nancy Cook and Marie Corbin
March 1, 1996

Purpose:

Explain how SATAN was used to reduce security exposures at a large Corporation.

History:

We were chartered to define the current state of our TCP/IP network security and reduce the level of exposure ASAP. Various tools were available but we chose SATAN for its breadth of checks, ease of use, availability, speed, and modifiability. Cost was not an issue for this project.

We then developed a plan on how best to canvas all nodes. When the enormity of the project finally hit us, we decided to augment SATAN's reporting capabilities with a few Perl scripts.

What we did and how/why we did it is detailed below. We could not have reduced our exposures (from 40% to 4% in 6 months on over 14,000 systems) as quickly without SATAN. It carried an authenticity which made our admins quickly sit up and take corrective action. And for that, Dan and Wieste, we are eternally grateful.

Strategy:

Simply put, our strategy was to check all of our registered IP addresses for network security problems (with SATAN). These problems were then reported to the DNS administrators who were responsible for getting the problems corrected or justified. The final security state of the systems was reported to upper management. We did not report on the initial state of problems because we were focusing on the unresolved/final state. (If the local administrators were quick to fix ALL their vulnerabilities, why tell their management about the past?)

Network Probing Tools:

We wrote Perl5 scripts that automatically generated reports (straight from the SATAN database), ran the scans (watched for a scan to finish and then started the next one that had not been scanned in 30 days) and then once a month, generated reports for all the DNS domains that we have defined (and a "not reported" report for all systems found but not in "assigned" address spaces).

We also wrote Perl5 scripts that remove entries from the database all-hosts files and facts files. If a host was renamed, removed or moved from one network to another, we needed to be able to alter the database. This tool we called db_cleanup.

We have 11 Class B networks (example would be 111.111.1-255.1-255 as one network). We have set up directories for each of these networks under /opt/satan/satan-1.1.1/results. These are our databases. Once per month a scan of each network is begun. We do subnet expansion (and

have proximity set to 0 by default) on files like:

```
./satan -d 111.111 -a 2 -s 111.111.1.1
./satan -d 111.111 -a 2 -s 111.111.2.1
...
./satan -d 111.111 -a 2 -s 111.111.255.1
```

for each of our 11 networks. Then our report generator is used along with our DNS subnet lists to generate reports.

A file like the following is stored in the gen_jobs directory for each of our DNS domains. The report generator will run all jobs in gen_jobs once a month, determine what has changed, give overall network statistics and mail a note to administrator to distribute reports.:

```
./gen_rpt 111.111 32 37 44 45 94 would be in file gen_lon_rpt
```

which would result in the London subnets being summarized in the resulting report. The above example would compile a report for the subnets 111.111.32 111.111.37 111.111.44 111.111.45 111.111.94 which could be as many as 5x255 systems for the London subnets.

The reports look like this:

```
                                                    Wed Feb 14 17:10:09 CST 1996

      ------------------------------------
      ---- Management Level Summary  ----
      ------------------------------------

IP address range: 111.111
     for subnets:       32  37  44  45  94


                          active hosts: 64
        # of hosts with vulnerabilities: 1  (1 %)
         # of different vulnerabilities: 2
             total # of vulnerabilities: 2



      ------------------------------------
      ----  Detailed Level Summary   ----
      ------------------------------------



 Scans of this subnet were conducted on Jan 29, 1996
 Subnet 32 had 7 active hosts,  The following had vulnerabilities:
-------------------------------------------------------------------------------------------------
                           most severe  -->       -->      -->      -->     -->     least
-------------------------------------------------------------------------------------------------
vulnerability ordinal      1      2     3 4     5    6    7      8      9       10    11
-------------------------------------------------------------------------------------------------
IP address      hostname  |TFTP  |Write|R|Modem|Bad |Send|NFS   |Remote|NFS   |Export|  X
                          |File  |-able|E|unres|FTP |Mail|unrest|Shell |export|via   |server
                          |Access|FTP  |X|trict|Vers|Vuln|ricted|Vuln. |un-   |Port- |Vulner
                          |      |Dir. |D|ed   |ion |    |export|      |priv. |Mapper|able
-------------------------------------------------------------------------------------------------
158.23.32.8     dnsax3           |     | |     |    |    |     7|      |     9|      |

 Subnet 37 had 5 active hosts and no vulnerabilities.

 Subnet 44 had 11 active hosts and no vulnerabilities.

 Subnet 45 had 40 active hosts and no vulnerabilities.

 Subnet 94 had 1 active hosts and no vulnerabilities.
-------------------------------------------------------------------------------------------------
 The following hosts were scanned and no vulnerabilities were found:
        dnsct1          dnsvx1          dnsvx2          dnsct3      xxxxke2
```

```
158.23.32.254    111.111.37.21    111.111.37.22    111.111.37.23    111.111.37.24
111.111.37.200          dnssn6           dnssn7           dnssn8           dnssn9
      dnssn10         dnssn11          dnssn12          dnsrt1           dnsrt2
      dnsdm1e   111.111.44.254          dnssn20          dnssn21          dnssn22
      dnssn23         dnssn24          dnssn25          dnssn26          dnssn27
      dnssn28         dnssn29          dnssn30          dnssn31          dnssn32
      dnssn33         dnssn34          dnssn35          dnssn36          dnssn37
      dnssn38         dnssn39          dnssn40          dnssn41          dnssn42
      dnssn43         dnssn44          dnssn50          dnssn53          dnssn45
      dnssn46         dnssn47          dnssn48          dnssn49          dnssn70
      dnssn51         dnssn52          dnssn54       xxxxxx       xxxxxx1
       dnsct2   111.111.45.254       dnsrt-le1

-----------------------------------INFORMATIONAL MESSAGES-----------------------------------
--------------------------------------------------------------------------------------------
```

To assistance the sysadmins, we ranked the vulnerabilities by severity so if they could only work on a few at a time, they knew which ones to work on first. The ranking was a factor of two issues: ease with which to fix the problem and ease of which to break in and cause serious damage. So a #1 is the easiest to fix and posed the most serious threat to system security. (Let the arguments begin!)

We then made available the tip sheets that come with SATAN and referenced them to the numbering system as noted. Vendor advisories were also collected from the web and given to our sysadmins to help them know which patches to apply. In some cases, we walked novice administrators through the patch process. In other cases where the patches gave us problems, we worked with the vendors until the problems were resolved (especially helpful for our overseas administrators.) In essence, we did as much of the leg work as we could so the effort required by the sysadmins was minimized.

In the report, the Remote shell vulnerabliity really spoke to several types of vulnerabilities (e.g. /.rhosts, hosts.equiv and passwordless accounts), so we specified which type of rsh vulnerability each system had at the bottom of the report in the INFORMATIONAL MESSAGES section.

If NIS was vulnerable it was also noted in the Informational Messages due to column limitations (you can only get SO much in 72 columns). We also put a switch in the report generator to tell if any vulnerable systems had not been scanned in x days. This watched for systems that had been problems that had been moved or hidden and we couldn't tell.

We currently have over 14000 systems (up from the initial scan in September 1995 of 12000) in 30 DNS domains on every continent in the world. Less than 4% of these have any SATAN detectable holes now and all have been examined for risk. Some of these are justified like a cdrom exported to the world so the exposure is even less than 4%.

We have varied response when doing the network scans. On one network with 214 systems on the entire network it took only 12 hours. Other networks can take 4 days for 2000+ systems. A lot depends on how many tests the inference engine fires off, the bandwidth between the scanning site and the scanned site, the other loads on the network etc .... We are able to scan all of our systems once per month on the HP.

Problems:

The project was relatively problem free. The tools worked as advertised

which is not the norm so we lucked out there.

SATAN has been amazingly bulletproof. Several times we were accused of causing network storms and were ALWAYS able to prove that it was not us. I know of only two systems that have been affected noticeably by the SATAN scans. Two highly modified VAX's running OLD TCP/IP implementations kind of got overwhelmed by the scans. We now schedule those scans for off hours.

There were a few times we thought we had a problem with SATAN and as we researched it, we found out the problem was with our understanding of the tool. For example, we modified the ./rules/todo file to unconditionally check for certain rsh conditions, when all we really needed to do was set the -u parameter on SATAN to say that we were to be considered an untrusted machine. Live and learn!

By the way, we don't pick up every system once a month when scanning. Since we start jobs whenever one has completed and because many of our networks cross over more than 12 hours of time zone, if the scans fire off when folks are not at work or machines are down for maintenance, we do not pick them up. But over time, we are getting the vast majority - at present more than 90% of the systems that have ever responded to an fping are rescanned every month.

Some performance analysis was done while SATAN was running. We are heavily memory bound, but we could also have used a bigger cpu. When not doing subnet expansion, the memory is not nearly as strained.

Configuration:

HP 9000/720 workstation with 48MB memory:
The OS was HP-UX 9.05 with all the known security patches installed. We also added extra security tools for the protection of the workstation against breakins and believe me we had plenty of attempts. We installed:
- tripwire (detects if files have changed)
- tcp wrappers (reports when a network service is accessed and by whom)
- cops (checks for internal security holes)
- SATAN (checks for network security holes)
- ESM (checks for internal security holes)
- syswatch (checks for changing conditions on the system, e.g. growing disk space)
- spionin (reports what files changed in the last 24 hours)

Suggestions for future usage of SATAN:

Use SATAN to track CERT alert progress:
One thing we did during the SATAN phase of the project was to incorporate checks for new CERT alerts in the SATAN engine. For example, we added checks for the known "password-less accounts" on IRIX systems. We also added 2 checks that pingware had but SATAN initially didn't. This idea should be continued as much as possible. That way we can quickly find systems that are actually vulnerable to the new CERT alert and track the progress of their clean up. Right now we send out CERT alerts but have no way of tracking if anyone is fixing the problems. Not all CERT alerts can be detected by SATAN, but my guess is a significant number can be.

To add the "password-less accounts" check, we modified the /opt/satan/satan-1.1.1/rules/todo file.

Continuation of Services:

Throughout this project, we were asked if these services (e.g. SATAN scans, password cracking) were to be offered on a continuous basis. The clients found this quite valuable. We explained that we configured the HP-UX workstation and Cray with that in mind. The CRAY (due to it's CPU capacity) is an excellent platform for cracking passwords. If need be, we can run the "crack" program on other platforms, but we do not encourage all sites to run it on their own systems, due to the extreme need for CPU capacity. This is definitely a service that lends itself well to being shared (done centrally).

The same is true for SATAN but not so much for the extreme CPU capacity need, but more from the learning curve point of view. It took a bit of work to thoroughly understand the SATAN tool (and I'm not sure we're really there yet) and what it was doing so we could efficiently operate it on the large scale that we did. It was not originally intended to be run on 12,000+ nodes which is obvious when you realize the reporting mechanism is totally GUI. That's why we developed numerous Perl5 scripts to maintain and report on the SATAN databases. We couldn't have met the project deadline without these extra scripts.

Summary:

This is one of the most successful projects that either of us has participated in. Cleaning up our WAN is a high priority because it is the backbone of our business but touting security has never before been a service that people called and thanked us for. For the first time in our experience, "selling" security was easy. The holes pointed out by SATAN were all legitimate and the impact was obvious both to management and administrators. The tools made the job easy when it could have been a nightmare. In summary, it was a joy to be able to make a difference and have fun doing it.

About the Authors:

Nancy Cook, Computer Security Advisor, has over 15 years of experience with programming, administration and computer security for Defense Contractors and Energy Industries. She is also the PROUD mother of 8 children and still thinks she is the luckiest woman alive that her husband chose her to be his wife. She can be reached at 214-951-3550.

Marie Corbin, Unix Systems Consultant, has over 15 years experience in computing for the Defense Department and Energy Industries. She can be reached at mariecor@aol.com. She is glad that she only has 1 child and that Kyle and Billy are so cool.