Ford Crews
GIAC Security Essentials Certification (GSEC)
Practical Assignment
Version 1.4b (amended August 29, 2002)
Resubmission

Asset Tracking in a Large Unmanaged Intranet

Introduction

I work for a government organization that is scattered across four states, and consists of seven separately managed labs plus staff and support elements for each. The organization has around 3500 employees and around 4200 networked workstation-class computers. I was given the following assignment:

- Design a way to track, certify and document machines added to the network
- Maintain a database of these machines
- Provide tools to mine this database for use in configuring our firewall
- Provide tools to mine this database for tracking of software licenses
- Provide tools to mine this database for verification of vulnerability compliance

Before

When I was given the assignment to design ways of more effectively track assets on our network, there were several different methods of tracking assets being used.

- To count machines on our network we used discovery scans of the various networks with tools such as ISS or STAT. While this could be effective in discovering vulnerabilities and assets it was time-consuming and consumed large portions of network resources. In addition this process often missed machines that were on-line for only short periods of time.

- To discover vulnerable machines on our networks, we used the same tools to scan for various vulnerabilities. While this was also effective, it was still time-consuming, consumed large portions of network resources, and would sometimes cause problems when the vulnerability scanners would actually DOS some network assets.

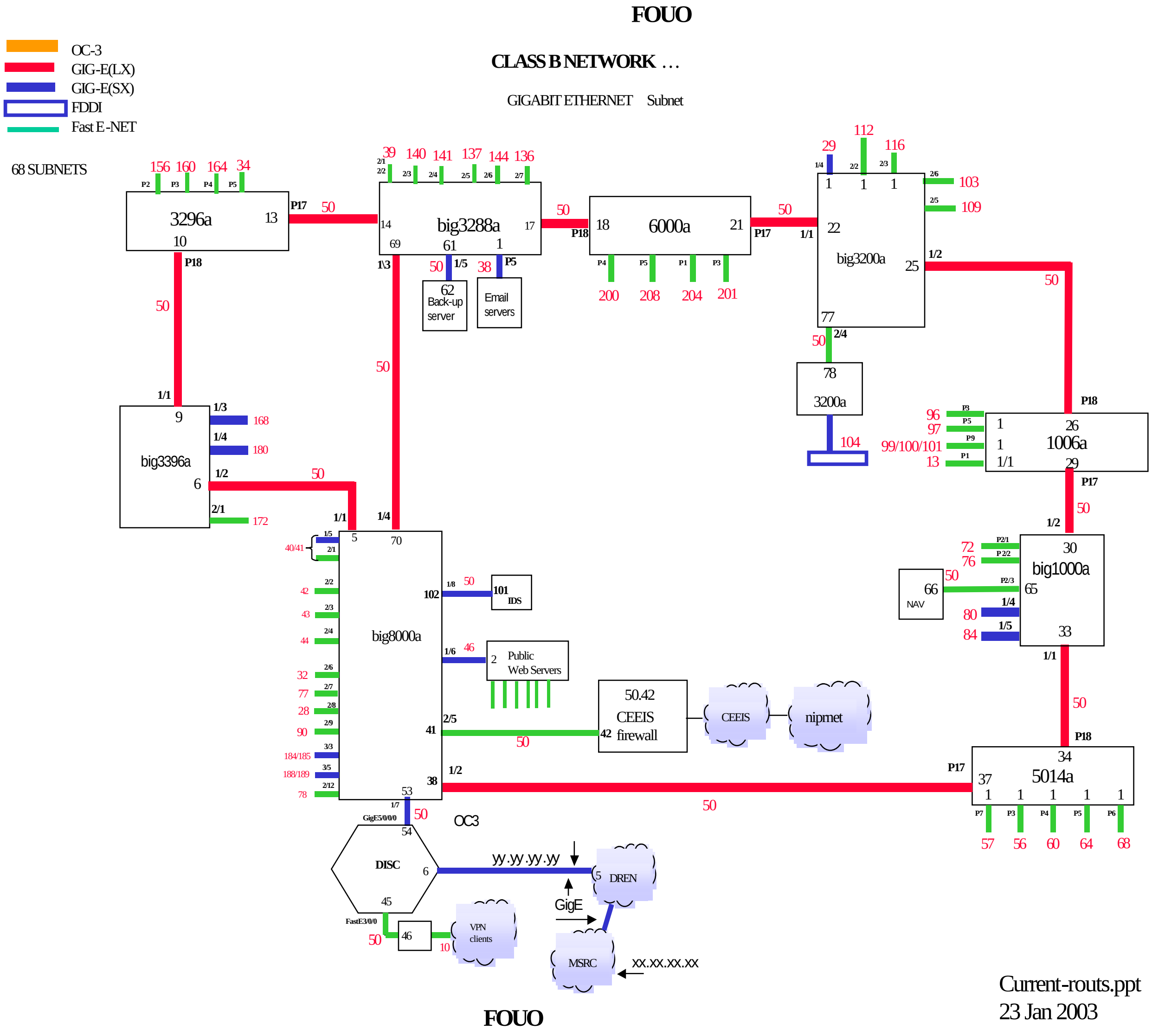
- To track software licenses we would send out data calls, basically sending email to all users asking if they used a certain piece of software, or thought they would if we provided them with a license for it. While this would give us an estimate of how many copies of each package we would need, it wasn't accurate. And, again, it was very time consuming. We

> would get less than a 20% response rates unless we tied getting the info to something like a password change where they didn't get their new password until they responded.

We knew we needed something to give us real-time access to information about systems and software on our network. Since we do not manage the individual machines or even the networks they were on, a lot of the managers of the computers expressed concern about security of data on their machines and do not trust anyone loading software on their computers. Solutions, then, involving installing clients on the machines were out. We were mandated by the DOD to provide them with lists of computers and software on their network as well as to make sure that vulnerabilities were patched on their network. We had no way to accurately do that.

Our network seems a little like the Wild West. Like the West, it's pretty much wide open, vulnerable to attack from all sides. Our network has only a few very specific protocols blocked and allowed only to specific subnets. Most of the machines are wide open to the Internet, and anyone is able to plug a machine into the network anytime they wish. We block telnet, ftp, and http to all machines except machines on our single public subnets, but leave the rest of the network wide open except for various know problems, such as KAZA, IRC, code red, etc. We are required to report new machines/applications added to our network to various authorities in our chain of command, and are under constant monitoring by them, and subject to random scans/inspections. We are under constant threat of machines/applications that we haven't reported being detected by higher ups that monitor our reporting for accuracy. Or even worse, we face the danger of having an unreported machine/application being compromised and the machine being confiscated for evidence. If a machine is confiscated, the researcher could lose a month or more of the data on the machine.

Below is a diagram of one of our class B networks. There are three other similar networks at sister sites.



During

From our use of HP Openview and knowledge of SNMP we knew that you could monitor the ARP tables in the routers on the network to discover new computers as they were added to the network, but it had no real mechanism to track new machines especially on subnets whose IP address were provided by DHCP. We noted that you could use SNMP (simple network management protocol) to query the routers for a table of IP/MAC pairs that the router knew about on each interface. Because of this, I got busy researching what it would take to develop an in-house piece of software to monitor the ARP tables in our routers and track changes over time in a database.

In our organization there is a three-level hierarchy of responsibility for each machine. The machine's SA (System Administrator), the IASO (Information Assurance Security Officer) in charge of the subnet the machine is on, and the LIASO (Lead Information Assurance Security Officer) in charge of all subnets that make up a lab. We need a system that would allow the correct people to be notified when a new machine is added. The notification would also include the actions they would need to perform to certify that the machine is compliant with security polices of our network.

Next I looked at different database options to determine what would be the best cost effective choice for a database to store our data in, and language to write the scripts and front-end in. After searching and evaluating several options, I determined that the most cost effective scalable database appeared to be IBPhoenix since it's a free open source solution that has good third party support and is ODBC capable. For the scripts I decided to use Perl and PHP. And for the front-end I decided to use HTML and PHP. My hardware choice for the project is a Dell 1.7 Ghz P4 with 40 gig of raid 5 storage and 1 gig of ram running Redhat 7.3. We also decided that due to the data being collected that we would install a standalone hardware firewall between this machine and the rest of our network. We installed rules in this firewall to restrict access to only machines at our sites that were currently listed in our database and they were only allowed access to port 443 for secure SSL transfers.

The first thing step in the project was installing Redhat 7.3 and making sure that someone had patched all known security vulnerabilities. Next we installed the latest versions of Apache, Perl, PHP, and IBPhoenix and patched all known vulnerabilities. After the hardware was ready and all the software components to be used were installed I started designing a basic structure for the database

Locating new devices is handled by 2 tables one called ARP_TABLE, the other called ROUTERS and a script called arp.php. The table ROUTERS contains a list of all the routers on our network, and the read-only SNMP community string used to access the routers. The script arp.php opens the table ROUTERS and loops through its rows doing snmpwalks on object 1.3.6.1.2.1.4.22. This returns IP/MAC pairs for each machine connected to each router. A query is then run against the table ARP_TABLE looking to see if the MAC address exists in the database. If the MAC address is found, the record is updated to show the current time as the last time the machine was detected on the network. The IP address is checked to make sure the machine hasn't moved. If the IP address has changed but to another address on the same subnet, the record is just updated with the new IP address. If the IP address is on a different subnet, the record is updated and the SA, IASO and LIASO are sent emails asking them to update the record to make sure the information is still correct, since the machine has most likely been physically moved.

Next if the record is not found at all, a new record is inserted into the database, and a quick scan is made to find open ports, DNS name, Netbios name and OS. This information is emailed to the IASO of the subnet with a request that they assign a SA to the machine.

A set of scripts is run once a day in a CRON job that does routine maintenance on the database.

- One script runs once a day and scans all machines with ISS and STAT and sends out email notifying each SA of machines that need fixing and each IASO/LIASO counts of machines with vulnerabilities. They can use this information to determine if their SA or IASO is overworked or not performing well. This list includes any new machines not scanned before, and any machines that haven't been scanned in 12 days. The SA is given 24 hours to fix his/her machines or an email is sent up the chain. This data is then imported into the database where it can be viewed or used to block machines in our firewall.

- Another script queries our LDAP servers and updates the lists of users on our network from there, keeping the user list up to date with the email system. This list is used to make it easy to assign users to machines, and is also used to find any machines that were assigned to a user when that user employment is terminated. When security is notified of a employee leaving they can run a report that emails all the IASOs involved with machines this user had access to notifying them to make sure the passwords are changed and any user accounts are deleted.

- Another script queries the database looking for any machines that need to be blocked from network access, and emails a block list to the firewall administrators who check it, and cut and paste it into the firewall acl. All machines marked as printers are blocked completely from internet access. Machines that failed any of the ISS/STAT vulnerabilities are blocked from access to the Internet by outside hosts. Machines that have been in the database for 36 hour or more and haven't been scanned or have missing information are blocked from network access. This script also sends email to the SA of the blocked machines reporting why they were blocked. A list of blocked machines goes to the appropriate IASO and LIASO.

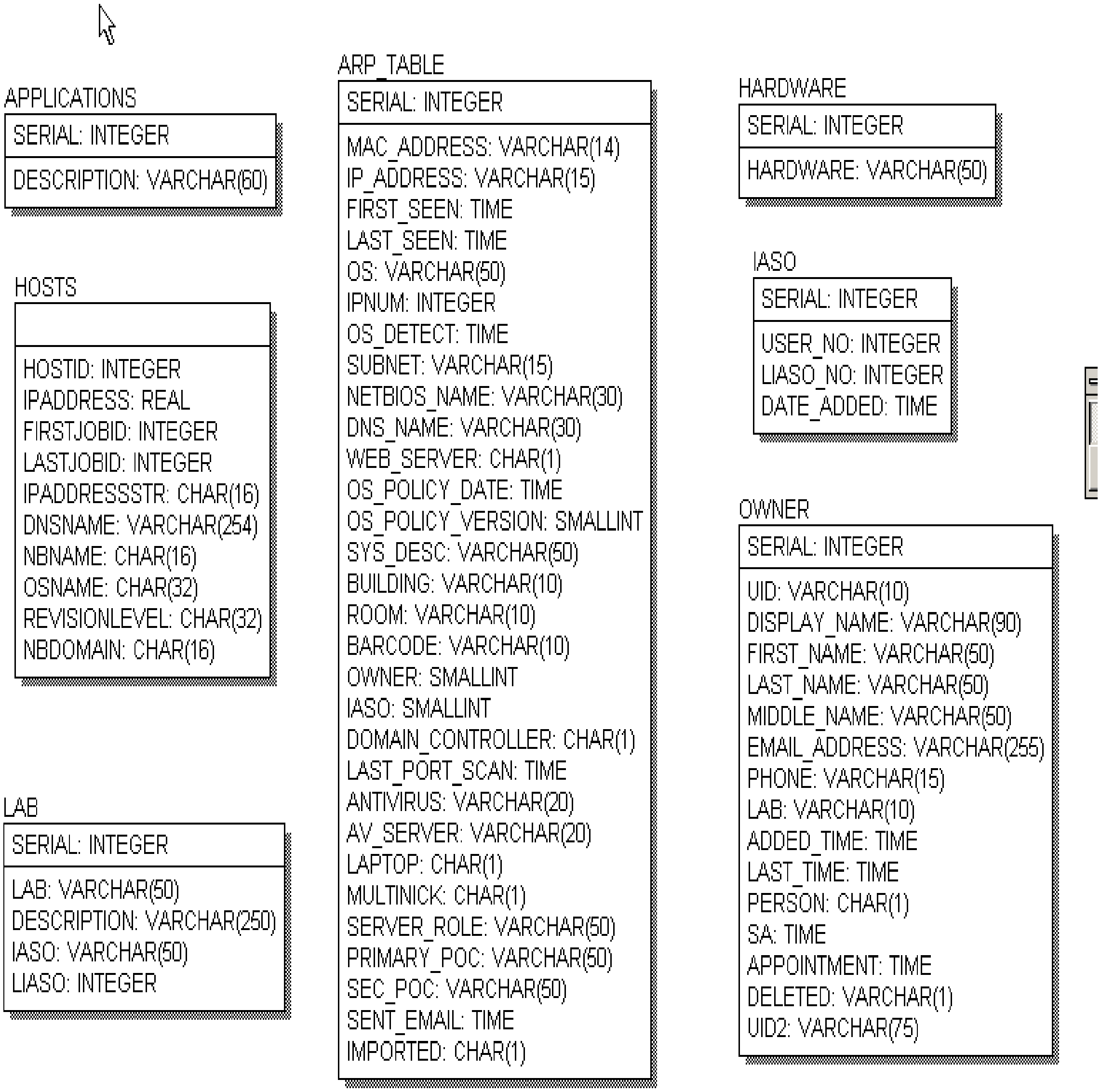
There are also a set of scripts that run hourly.

- One script checks the output logs of our IDS and emails any serious attack attempts to appropriate SA and IASO. This information has been checked, but not merged into the database yet. I am confident it will be merged soon.

- Another script does ping sweeps of our network to make sure all machines talk to the router so the ARP tables keep a complete list of all attached machines. This was added when machines it was discovered that machines like printers could sometimes stay for months on a subnet without talking to any device off their subnet so they didn't show up in the router ARP tables.

- Another script checks for machines that have been assigned an SA for more than 24 hours, but haven't had their information filled in by the SA. A second email is sent to them reminding them that they must fill in the information or their machine will be blocked from internet access.

Below are the tables used in the system.



APPLICATIONS
SERIAL: INTEGER
DESCRIPTION: VARCHAR(60)

HOSTS
HOSTID: INTEGER
IPADDRESS: REAL
FIRSTJOBID: INTEGER
LASTJOBID: INTEGER
IPADDRESSSTR: CHAR(16)
DNSNAME: VARCHAR(254)
NBNAME: CHAR(16)
OSNAME: CHAR(32)
REVISIONLEVEL: CHAR(32)
NBDOMAIN: CHAR(16)

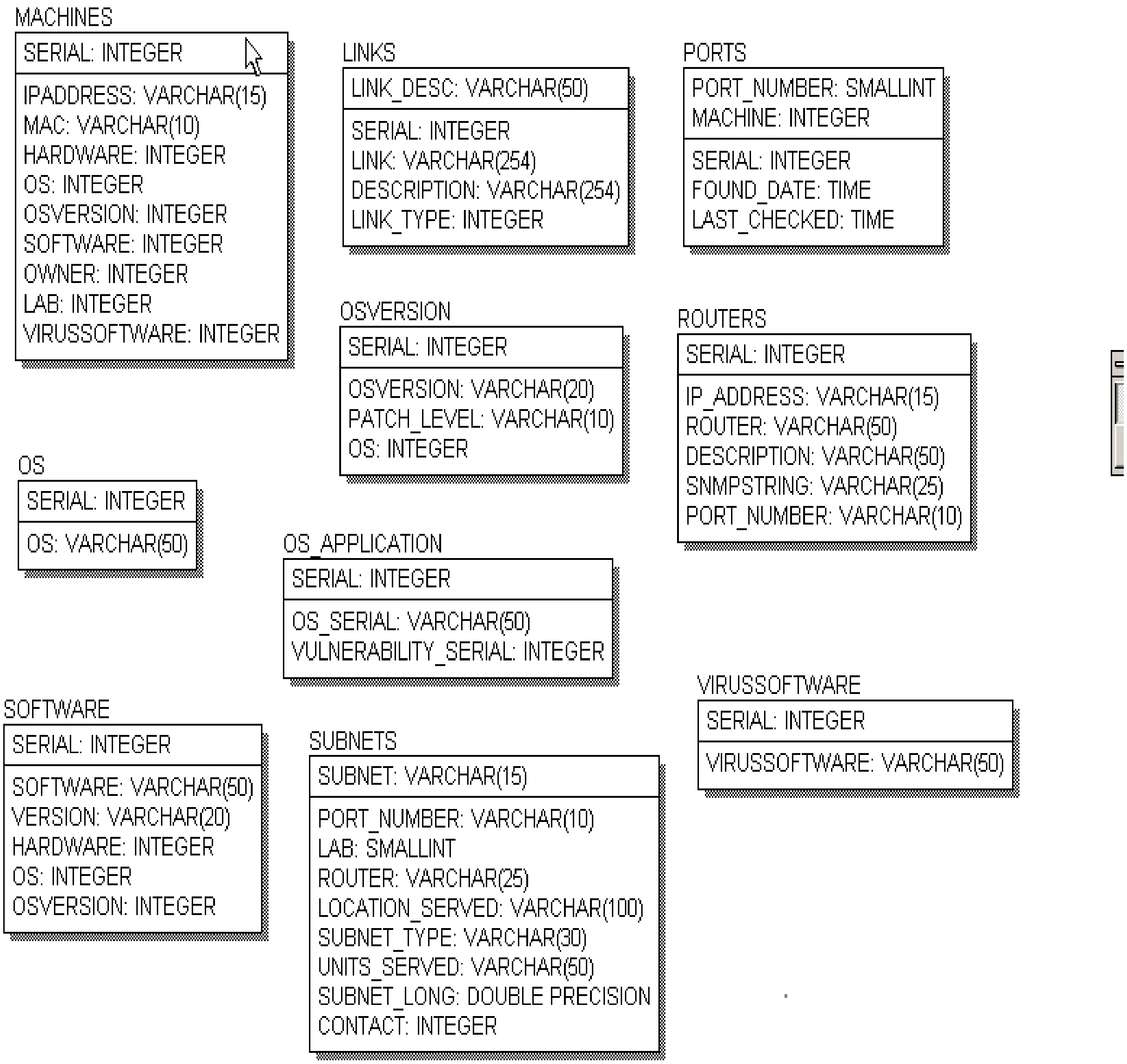
LAB
SERIAL: INTEGER
LAB: VARCHAR(50)
DESCRIPTION: VARCHAR(250)
IASO: VARCHAR(50)
LIASO: INTEGER

ARP_TABLE
SERIAL: INTEGER
MAC_ADDRESS: VARCHAR(14)
IP_ADDRESS: VARCHAR(15)
FIRST_SEEN: TIME
LAST_SEEN: TIME
OS: VARCHAR(50)
IPNUM: INTEGER
OS_DETECT: TIME
SUBNET: VARCHAR(15)
NETBIOS_NAME: VARCHAR(30)
DNS_NAME: VARCHAR(30)
WEB_SERVER: CHAR(1)
OS_POLICY_DATE: TIME
OS_POLICY_VERSION: SMALLINT
SYS_DESC: VARCHAR(50)
BUILDING: VARCHAR(10)
ROOM: VARCHAR(10)
BARCODE: VARCHAR(10)
OWNER: SMALLINT
IASO: SMALLINT
DOMAIN_CONTROLLER: CHAR(1)
LAST_PORT_SCAN: TIME
ANTIVIRUS: VARCHAR(20)
AV_SERVER: VARCHAR(20)
LAPTOP: CHAR(1)
MULTINICK: CHAR(1)
SERVER_ROLE: VARCHAR(50)
PRIMARY_POC: VARCHAR(50)
SEC_POC: VARCHAR(50)
SENT_EMAIL: TIME
IMPORTED: CHAR(1)

HARDWARE
SERIAL: INTEGER
HARDWARE: VARCHAR(50)

IASO
SERIAL: INTEGER
USER_NO: INTEGER
LIASO_NO: INTEGER
DATE_ADDED: TIME

OWNER
SERIAL: INTEGER
UID: VARCHAR(10)
DISPLAY_NAME: VARCHAR(90)
FIRST_NAME: VARCHAR(50)
LAST_NAME: VARCHAR(50)
MIDDLE_NAME: VARCHAR(50)
EMAIL_ADDRESS: VARCHAR(255)
PHONE: VARCHAR(15)
LAB: VARCHAR(10)
ADDED_TIME: TIME
LAST_TIME: TIME
PERSON: CHAR(1)
SA: TIME
APPOINTMENT: TIME
DELETED: VARCHAR(1)
UID2: VARCHAR(75)



MACHINES
SERIAL: INTEGER
IPADDRESS: VARCHAR(15)
MAC: VARCHAR(10)
HARDWARE: INTEGER
OS: INTEGER
OSVERSION: INTEGER
SOFTWARE: INTEGER
OWNER: INTEGER
LAB: INTEGER
VIRUSSOFTWARE: INTEGER

LINKS
LINK_DESC: VARCHAR(50)
SERIAL: INTEGER
LINK: VARCHAR(254)
DESCRIPTION: VARCHAR(254)
LINK_TYPE: INTEGER

PORTS
PORT_NUMBER: SMALLINT
MACHINE: INTEGER
SERIAL: INTEGER
FOUND_DATE: TIME
LAST_CHECKED: TIME

OSVERSION
SERIAL: INTEGER
OSVERSION: VARCHAR(20)
PATCH_LEVEL: VARCHAR(10)
OS: INTEGER

ROUTERS
SERIAL: INTEGER
IP_ADDRESS: VARCHAR(15)
ROUTER: VARCHAR(50)
DESCRIPTION: VARCHAR(50)
SNMPSTRING: VARCHAR(25)
PORT_NUMBER: VARCHAR(10)

OS
SERIAL: INTEGER
OS: VARCHAR(50)

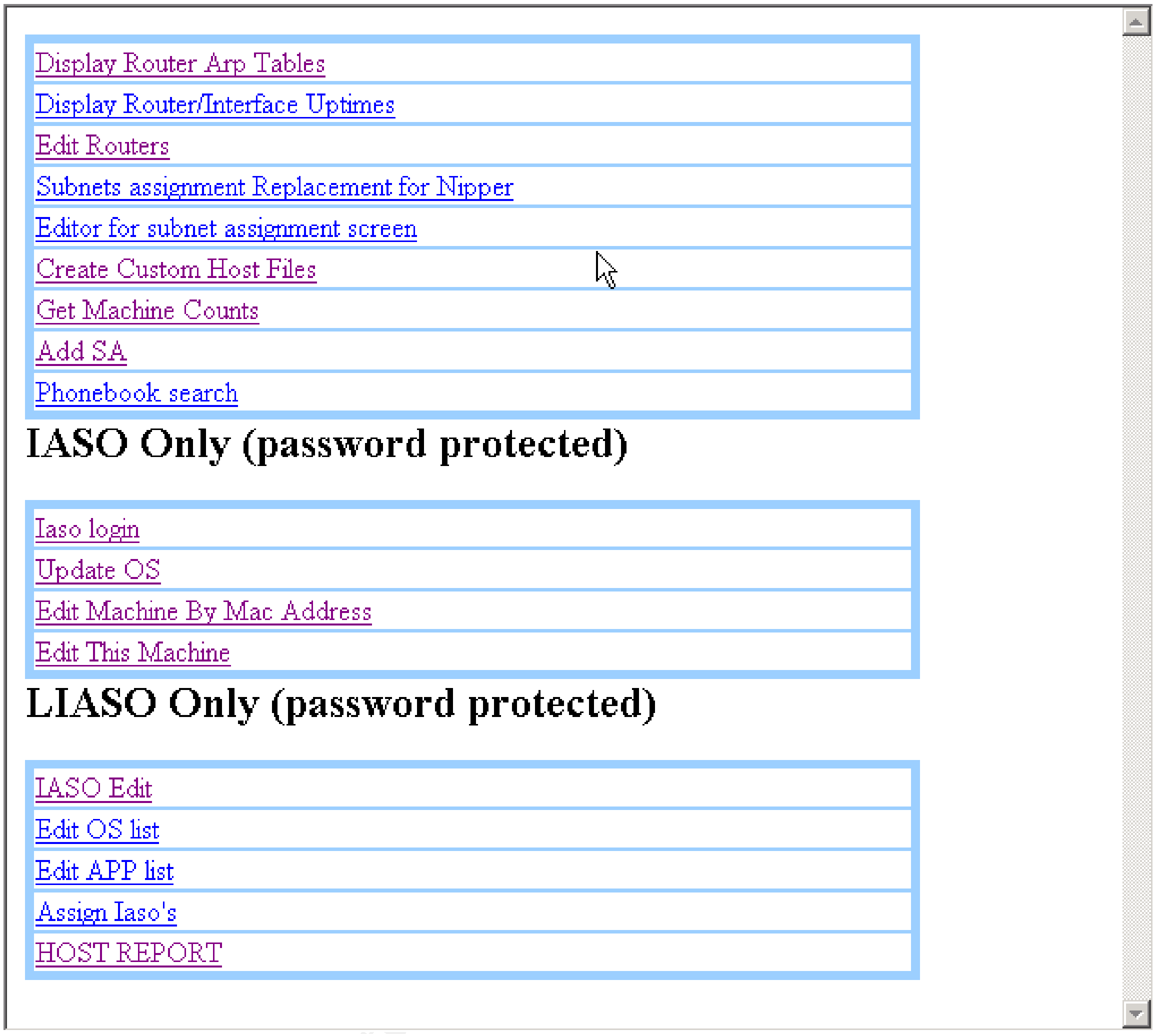
OS_APPLICATION
SERIAL: INTEGER
OS_SERIAL: VARCHAR(50)
VULNERABILITY_SERIAL: INTEGER

VIRUSSOFTWARE
SERIAL: INTEGER
VIRUSSOFTWARE: VARCHAR(50)

SOFTWARE
SERIAL: INTEGER
SOFTWARE: VARCHAR(50)
VERSION: VARCHAR(20)
HARDWARE: INTEGER
OS: INTEGER
OSVERSION: INTEGER

SUBNETS
SUBNET: VARCHAR(15)
PORT_NUMBER: VARCHAR(10)
LAB: SMALLINT
ROUTER: VARCHAR(25)
LOCATION_SERVED: VARCHAR(100)
SUBNET_TYPE: VARCHAR(30)
UNITS_SERVED: VARCHAR(50)
SUBNET_LONG: DOUBLE PRECISION
CONTACT: INTEGER

Below is the Maintenance page used to update database information.



Display Router Arp Tables
Display Router/Interface Uptimes
Edit Routers
Subnets assignment Replacement for Nipper
Editor for subnet assignment screen
Create Custom Host Files
Get Machine Counts
Add SA
Phonebook search

IASO Only (password protected)

Iaso login
Update OS
Edit Machine By Mac Address
Edit This Machine

LIASO Only (password protected)

IASO Edit
Edit OS list
Edit APP list
Assign Iaso's
HOST REPORT

The above page is used to manage the database as well as to maintain maintenance tables such as list of current OS's used on base, Applications used on base, and Assign IASOs to specific subnets.

Display Router Arp Table gives a quick on the fly list of all IP/MAC pairs currently in use on base. This allows for quick reference to the ARP tables in all the routers at all our sites. Can be very handy when trying to track down routing problems.

Display Router/Interface Uptimes lists each router and gives its uptime as well as the uptime of each interface on that router. Also displays some error counts on the routers. Can be hand a handy reference when trying to track down router problems.

Edit Routers is used to add new routers into the system or remove old routers from the system. Each routers IP address, name of the router, description of the router and snmp community string are collected. Router name and description are used in reports and network diagrams.

Subnet assignment Replacement for Nipper used to display information about each subnet at all sites. You can see what router and port each subnet is own, which lab uses that subnet, contact for the subnet as well as some statistics about number of machines currently on the subnet. You can also drill down on each subnet and get detailed information on machines on each subnet.

Editor for subnet assignment used to maintain the list of subnets at all sites.

Create Custom Host Files allows lists of IP addresses to be pulled for various reasons. If someone needs a list of all machines running NT4 servers running MS SQL 6.0 to block at the firewall due to some new vulnerability that just came out they can come here and produce a quick list.

Get Machine Counts gives a quick count of machines that meet a certain criteria. If someone needs to know how many laptops each lab has running Windows ME they can come here to get a count.

Add SA allows System Administrators to be added to the system/deleted from the system. Also allows any training that the SA has received to be tracked.

Phonebook Search Allows quick phone number/email address searches across all sites.

IASO login can be used by each IASO to get a quick list of all machines they currently have assigned to them that aren't compliant with current pc standards for a networked pc on base.

Update OS is used to update machines without having to login as a specific IASO. If you know the ip address of a machine you can change its data directly here.

Edit Machine By Mac Address if someone knows the Mac address of a machine, but doesn't know the current IP (machines on DHCP subnets) this allows you to quickly find/update them.

Edit This Machine allows you to quickly update the information about the currently logged into machine.

IASO Edit assign IASOs to a lab.

Edit OS List add a new OS or delete a no-longer supported OS from the database. There are constraints on the database to not allow an OS to be deleted/edited if there are currently machines assigned to it.

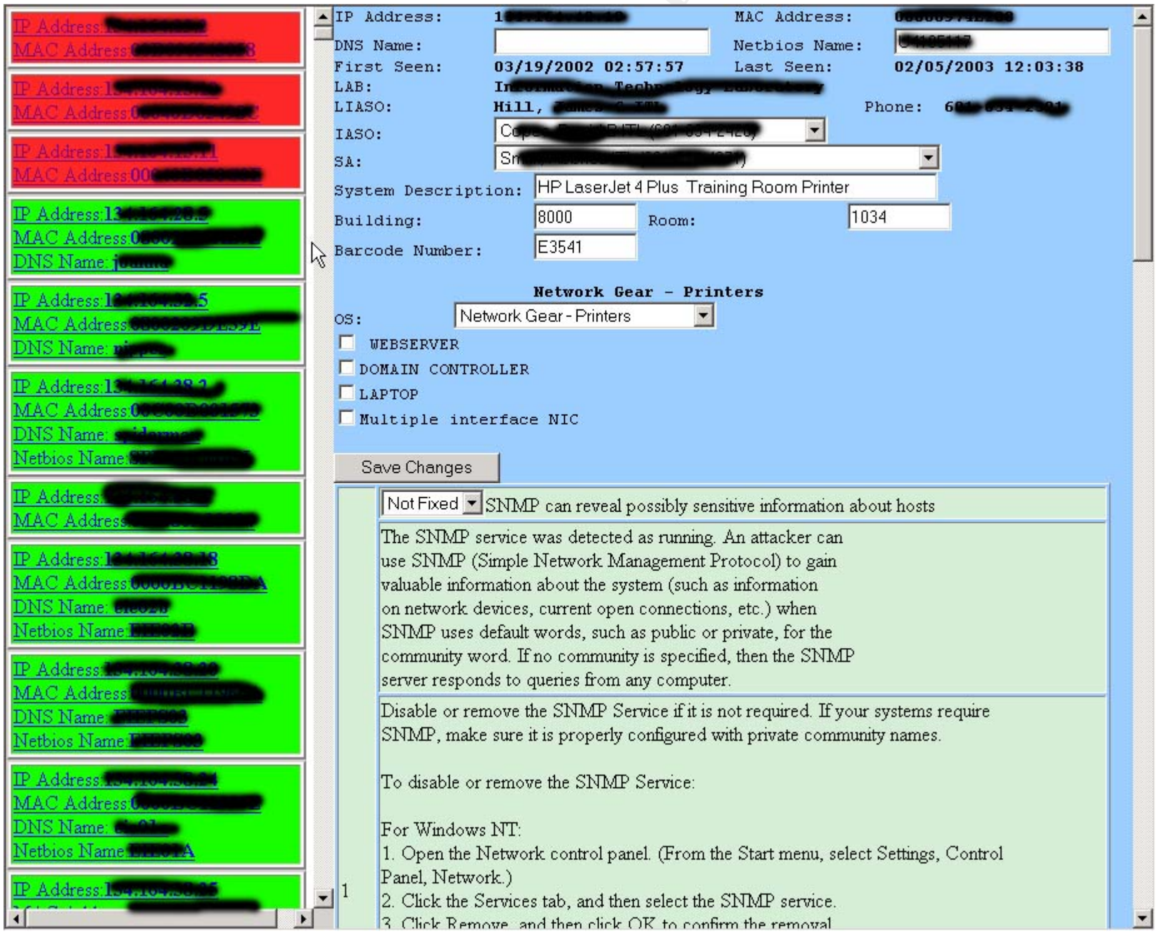
telnet	23	TCP
Tftp	69	UDP
httpd	80	TCP
3com-tsmux	106	UDP
SNMP Server	161	UDP
snmp	161	UDP

Edit APP List add a new Application or delete a no-longer supported Application from the database. There are constraints on the database to not allow an application to be deleted/edited if there are currently machines assigned to it.

Assign Iaso is used by LIASOs to assign IASOs to specific machines/subnets of machines.

Host Report gives print reports about machines or sets of machines in a format easily exportable to access/excel so users who aren't familiar with odbc can still use the data we collect. Also allows an IASO to print out al list of all their machines that need attention along with building, room number, barcode number, and a list of problems. This makes it easy for the IASO to go to each machine and fix the problems.

Below is an example of the data collected for each machine:



IP Address:[redacted]
MAC Address:[redacted]

IP Address:[redacted]
MAC Address:[redacted]

IP Address:[redacted]
MAC Address:[redacted]

IP Address:[redacted]
MAC Address:[redacted]
DNS Name:[redacted]

IP Address:[redacted]
MAC Address:[redacted]
DNS Name:[redacted]

IP Address:[redacted]
MAC Address:[redacted]
DNS Name:[redacted]
Netbios Name:[redacted]

IP Address:[redacted]
MAC Address:[redacted]

IP Address:[redacted]
MAC Address:[redacted]
DNS Name:[redacted]
Netbios Name:[redacted]

IP Address:[redacted]
MAC Address:[redacted]
DNS Name:[redacted]
Netbios Name:[redacted]

IP Address:[redacted]
MAC Address:[redacted]
DNS Name:[redacted]
Netbios Name:[redacted]

IP Address:[redacted]
[redacted]

IP Address: [redacted]
MAC Address: [redacted]
DNS Name:
Netbios Name: [redacted]
First Seen: 03/19/2002 02:57:57
Last Seen: 02/05/2003 12:03:38
LAB: [redacted]
LIASO: Hill, [redacted]
Phone: [redacted]
IASO: [redacted]
SA: [redacted]
System Description: HP LaserJet 4 Plus Training Room Printer
Building: 8000
Room: 1034
Barcode Number: E3541

Network Gear - Printers

OS: Network Gear - Printers
☐ WEBSERVER
☐ DOMAIN CONTROLLER
☐ LAPTOP
☐ Multiple interface NIC

1

Not Fixed SNMP can reveal possibly sensitive information about hosts

The SNMP service was detected as running. An attacker can use SNMP (Simple Network Management Protocol) to gain valuable information about the system (such as information on network devices, current open connections, etc.) when SNMP uses default words, such as public or private, for the community word. If no community is specified, then the SNMP server responds to queries from any computer.

Disable or remove the SNMP Service if it is not required. If your systems require SNMP, make sure it is properly configured with private community names.

To disable or remove the SNMP Service:

For Windows NT:
1. Open the Network control panel. (From the Start menu, select Settings, Control Panel, Network.)
2. Click the Services tab, and then select the SNMP service.
3. Click Remove, and then click OK to confirm the removal

The above forms shows some of the data we collect for each machine. Most of it is automatically collected, so the actual data entry takes just a few seconds for each machine. When a machine is determined to have vulnerability, we try to provide enough information to the Information Assurance Security Officer (IASO)/System Administrator (SA) to enable him/her to fix it without having to go to another source. For example if a Microsoft SQL vulnerability is detected on a machine we give them an explanation of how to fix it and a link to the patch.

We use the system to document vulnerabilities on systems by scanning systems discovered with ISS and STATS. These vulnerabilities are then included in the machine summary where the IASO/SA can quickly determine what's wrong with all the machines they are in charge of, and to make sure they fix the most critical ones first. Machines with major security holes turn red, machines with minor security problems show up yellow, while correctly configured machines show up green. Machines are also moved to the top of the list the scanners detect that they have major security problems that need to be patched.

After

Now that the system is in place and running, when a new machine is introduced to our network for the first time, or reintroduced to the network after an extended absence, the LIASO for the subnet is notified of its addition to the network by a PHP script monitoring the ARP tables in our routers. A record is also inserted into an Interbase database, which will be monitored to make sure the rest of the process of securing/documenting the machine is achieved. This record is monitored to see if an IASO is assigned to the machine by close of business the following business day. If the machine is assigned to an IASO that IASO is emailed asking him/her to assign a System Administrator to the machine. If the machine is not assigned to an IASO, the LIASO is sent a reminder that the machine needs to be assigned. If two days pass and no IASO has been assigned, the reminder goes out again with a carbon copy to the entire information assurance team (IAT) to make sure the machine does get assigned an IASO.

Once the IASO is assigned the record is monitored again to make sure that an SA is assigned within 2 days, with reminder email sent each day, and reminders up the chain in 2 days. During this process initial scans of the machine are made using NMAP and other tools to determine open ports, the best guess of the OS on the machine, and a generic vulnerability scan is run with ISS.

Once the SA is assigned, the SA can pull up the record of the machine and see any vulnerabilities we have found so far and correct them. They are requested to fill in all the manual information like building, room number, and barcode. They also are requested to verify the OS, indicate if it's a server, laptop, or has multiple nicks.

After the information is entered, the record in the database is marked for scan. Before the scan script is run a script runs through the database and marks all old records that haven't been scanned in 12 days as needing to be scanned. The scan script runs every workday and takes all the machines marked and does an OS specific scan with ISS and STATS to make sure that all known vulnerabilities are detected. The record is only updated in the database if the machine is successfully scanned so machines that are *off* or on TDY will get retried in the next scan. If all is going correctly, it means that a machine added to the network should be scanned and patched within a week and follow-up scans should be done at least twice a month.

It has only been a couple of months since we got the system fully implemented at all our sites, but it seems to be running smoothly. We currently are tracking 4153 machines. Last year before we got the system up and running, it would take three of us at least a week to handle each IAVA release. The ones released after this system went into place have taken less than 20 minutes for us to verify compliance.

We have received praise from Managers as well as end users because we now know exactly who to inform of new security concerns. We also have the ability to quickly report compliance up the chain of command. After implementation of the new system for tracking assets at our site we are able to do the following:

- Know when a new machine is added to our network
- Know when a machine was removed from our network
- Know when a machine is moved from one subnet to another
- Know which machines need to be scanned where we don't waste time and resources scanning a Windows machine for a Linux vulnerability
- Know that all machines on our network have been accounted for
- Able to provide custom reports on short notice

We have been contacted by a couple of other agencies about using our system to help track their assets. One is going to start using it within the next month, while another is still evaluating it for their sites.

The next step in the process will be to start collecting information from our IDS systems and our Anti-Virus servers. This will give us a single page resource on a machine to see who is in charge of the machine, what patches it has, what anti virus definitions are on it, a list of viruses detected, and a list of exploits attempted on the box. Eventually I would also like to get to the point where the database is used in a completely automatic way to drive our firewall. So when an uncertified machine is plugged into the network it is blocked in the firewall, or better have the firewall block all access until they are in the certification database.

References:

Noel Davis, “Buffer Overflow; Secure PHP Coding”, Feb 20, 2001 URL: http://linux.oreillynet.com/lpt/a/632

Douglas F Gray, “Inprise to release Interbase 6 as open-source”, Jan 5, 2002: URL: http://archive.infoworld.com/articles/ec/xml/00/01/04/000104ecinprise.xml

K. McCologhrie, “RFC2011, SNMPv2 Management Information Base for the Internet Protocol using SMIv2”, November, 1996: url: http://www.faqs.org/rfcs/rfc2011.html

Richard Sigle, “Building a secure Redhat Apache Server”, Feb 6, 2001 URL: http://www.linuxdocs.org/HOWTOs/SSL-RedHat-HOWTO.html&e=912

Wrox Press, “LDAP”, Feburary 8, 2002 URL: http://www.wdvl.com/Authoring/Languages/PHP/Pro/prophp1_1.html