Global Information Assurance Certification Paper

A Survey of the Basic Functionality of SAINT

By Paul Engler

Understanding the services available on a network is intrinsic to providing a secure computing environment. Even though you may have installed the operating system and applications, you may not know all of the services that are available on a computer. Operating systems are not generally shipped with secure configurations; extra services are started and are exploitable. The same is true for applications that may run secondary services for administrative purposes. However, even if only know services are available, they may still be vulnerable to attack. One tool that can help the security administrator track down and understand these services is SAINT, the Security Administrators Integrated Network Tool. SAINT is one of a growing number of auditing tools available to security administrators. What differentiates it from the rest is its multiple modes of operation, configurable scan levels, and good reporting tools.

Modes of Operation:

SAINT has three modes of operation. The first of these is execution via the command line interface. This is most useful if you are planning to automate scans of your network via cron jobs. This method is also the most secure because a server does not have to be started to access the web based GUI. However, running only off the command line denies access to the reporting functions. You can however import the database onto another server for reporting purposes.

The Second and simplest method for running SAINT is through its web interface. The only requirement is that a web browser is available on the server that is running SAINT. The browser that is used can be configured after SAINT has been installed by editing the config/paths.pl file. Access to the web interface is gained by executing the "saint" script from the directory where SAINT was installed.

Finally, you can run SAINT as a server. In this case you specify a port for the server to listen on and hosts that are allowed to connect. If none are specified then the defaults are taken out of the config/saint.cf file. Once the server is started it can then be accessed remotely via a web browser from the defined trusted hosts. Please note that when run in remote mode SAINT does not send the password across the network. However it is still possible for the session key to be exposed over the network via the browser being used. This severity of this problem is reduced by the fact that each time the server is started new passwords can be chosen. So by limiting the run time of the server you can limit the time the session key is valid. To completely avoid the problem you can run SAINT exclusively from the command line, and the problem will not occur. For more information you can reference CVE 1999-0151 or CERT CA-1995-07.

Configurable Scan Levels:

By default SAINT performs 5 types of scans plus a custom scan that is undefined by default. The scans provided are Light, Normal, Heavy, Heavy+, and Top 10. As the names imply each the scan get progressively more intrusive. The options for each of the scans can be configured

either by editing the config/saint.cf file or through the web interface. The saint.cf file is well documented internally and further documentation can be found in the SAINT reference at the SAINT website or in the documentation provided within the software bundle. Custom probes can also be built although this can be somewhat complex. All of the scripts used to perform scans are contained in the bin directory and end in ".saint". Each of these scripts is written in perl and requires a good understanding of Perl and the SAINT database format in order to be modified.

Besides just being able to configure what ports to scan and services to look for, SAINT also allows items such as the number of concurrent threads to run, time out values, and scan speeds to be set. Most importantly it lets you define proximity values, allowed/disallowed networks and subnets, and domains not to scan. It is very important to make sure these are configured properly because when SAINT's inference engine finds trust relationships it will add new hosts to the list of hosts to probe. So if you are not careful SAINT will extend its scan outside of your authorized targets networks to areas that you may not have permission to scan. This could easily be misinterpreted as a malicious act, so ensure that SAINT stays within authorized bounds to by having the above parameters properly configures.

Reporting Tools:

Currently three methods exist for reporting out of SAINT. You can use the web interface that is provided with the software if you are running in the default mode or are connected to a remote server. If you are running from the command line a report will be dumped out at the end of the scan. Finally a set of scripts or an editor could be used to parse information out of the database. It the near future WWDSI will be providing a report writer that will allow users to build customized reports based on information in the SAINT databases.

The Web interface allows for browsing of the scan information via vulnerability, host information, and trust relationships. Through this interface each of the possible vulnerabilities is sorted via the severity of the problem. The interface will also highlight the vulnerabilities listed in the SANS TOP 10. However, it is important to note that the list of vulnerabilities may not be accurate. For example SAINT flags Windows machines as being vulnerable to DoS attacks. Even if the machines have been patched it will still list this as a possible vulnerability.

Probably the most useful part of the web interface is the tutorial sections. With this SAINT provides explanations of the security holes as well as resolutions. Also useful is the fact that the vulnerabilities are indexed using CVE (Common Vulnerabilities and Exposures). CVE functions like a dictionary allowing you to cross reference vulnerabilities across different organizations such as CERT and SANS. The list of sites that use CVE are located at http://www.cve.mitre.org/compatible/.

Because the database format is well documented in the html/docs/saint.db.html file, parsing the information via scripts is fairly simple. Each scan will produce four database files. The facts database keeps track of all vulnerabilities and services found. The all-hosts database keeps track of all the hosts located during the scan. The todo database keeps track of what scans have been performed. Finally the CVE database keeps track of all of the vulnerabilities found that can be

located in CVE or the SANS Top Ten lists. Although the databases are well documented, as of version 3.1.1 beta 1 of SAINT there are some discrepancies between the documentation and what is produced in the database files. An example of one is in the "facts" database field definitions. Field four is defined as listing the vulnerability severity. However, a severity level of "bo", "host", "gi", and "i" can now occur in this field, which are not covered in the documentation provided.

Despite the above discrepancies useful information can still be gathered from the databases. For example the following simple script will parse through the facts file looking to match a field. So to find all facts about a machine named "test" we would invoke the script "# ./search_saint.pl saint-data 0 test".

```
#!/usr/local/bin/perl
my $DB_NAME=$ARGV[0]; #Name of db to scan
my $FIELD=$ARGV[1]; #field to scan see saint.db.html for reference
my $QUERY=$ARGV[2]; #data to query for
my $DB_ROOT="/usr/local/saint-3.1.1/results/";

open (FACT_DB,"<$DB_ROOT/$DB_NAME/facts") or die "File does not exist. ";
foreach ( <FACT_DB> ) {
chomp;
@facts = split /\|/,$_;
print "@facts\n" if ($facts[$FIELD] =~ /$QUERY/ );
}
close (FACT_DB);
exit;
```

Another possibility is if you created different databases each day you could compare the all-hosts file to look for machines that have appeared or disappeared from the network. So really the only limiting factors are how much time and creativity the user wishes to put into creating customized reports.

Conclusion:

Overall SAINT is an excellent tool for the auditing of networks and hosts. Its customizability, while sometimes complex, makes for a very flexible tool. Its reporting functions allow security administrators to quickly locate problems within areas being scanned. And the tutorials provided educate and allow problems to be fixed. All of this contributes to make SAINT a valuable tool for any security administrator.

References:

Carnegie Mellon University. "CERT® Advisory CA-1995-07 SATAN Vulnerability: Password Disclosure." 23 Sept 1997 URL: http://www.cert.org/advisories/CA-1995-07.html (20 Nov 2000)

CVE. "Common Vulnerabilities and Exposures." 17 Aug 2000 URL: http://www.cve.mitre.org/about/ (19 Nov 2000)

"CVE-1999-0151." URL: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0151 (20 Nov 2000)

The SANS Institute. "SANS Resource – How To Eliminate The Ten Most Critical Internet Security Threats." Version 1.30 17 Nov 2000. URL: http://www.sans.org/topten.htm (21 Nov 2000)

World Wide Digital Security Inc. "SAINT Frequently Asked Questions." URL: http://www.wwdsi.com/demo/saint/faq.html (21 Nov 2000)

World Wide Digital Security Inc. "SAINT Introduction." URL: http://www.wwdsi.com/demo/saint/intro.html#what-is-saint (19 Nov 2000)

World Wide Digital Security Inc. "SAINT Reference." URL: http://www.wwdsi.com/demo/saint/saint_reference.html (20 Nov 2000)

World Wide Digital Security Inc. "Vulnerabilities." URL: http://www.wwdsi.com/demo/saint/vulnerability_tutorials.html (21 Nov 2000)

World Wide Digital Security Inc. "SAINT Writer" URL: http://www.wwdsi.com/saintwriter/index.html (19 Nov 2000)