BogoSec:
Source Code Security Quality Calculator

Dustin Kirkland
dustin.kirkland@us.ibm.com
Loulwa Salem
loulwa@us.ibm.com

May 2006

Abstract

BogoSec is a source code security quality metric tool. It wraps multiple source code scanners, invoking them on its target code, and produces a final score that approximates the security quality of the code. BogoSec aims to increase awareness of source code vulnerablites by identifying greatly offending code and charting security quality trends over time. For additional information on BogoSec, please refer to the Web site http://bogosec.sourceforge.net/.

1 Background

The CERT Coordination Center (CERT/CC) reported 5,990 vulnerabilities in 2005 compared to 171 in 1995. Many software security vulnerabilities occur because of poor programming practices. Some vulnerabilities are algorithmically detectable by static source code scanners designed for identifying potential security issues. As the number and severity of potential security holes per line of code increase, it is reasonable to believe that the overall quality of the source code in terms of security decreases. BogoSec metrics are computed values that attempt to reflect relative ratings of source code security quality for comparative purposes.

The motivation behind BogoSec is to influence developers to produce more secure source code over time. Various scanners exist that point developers to potentially insecure sections of code, however, developers are often reluctant to use such scanners because of a seemingly high degree of "false positive" output as well as the difficulties associated with use. BogoSec attempts to reduce the penalty of false positives while broadening the scope of the source scan by using multiple independent scanners. This produces high-level metrics that allow developers and users alike to comparatively judge the quality of the source code in terms of security.

2 Methodology

Several source code scanners exist that identify numerous vulnerabilities with varying accuracy and success. BogoSec parses the output of any number of source code scanners and computes its metric based on the number, severity, and frequency of potential bugs found as per number of lines scanned. BogoSec currently supports the following scanners:

- Flawfinder
- ITS4
- RATS

Support for additional scanners is easily extended by creating plugins that understand the input parameters and parse the output of the new scanners. This is useful for incorporating support for proprietary or internal scanning tools or newer public tools.

BogoSec requires that at least one of the scanners listed in this paper is installed on the system and can be found in the path. These scanners are not distributed as part of BogoSec. However, BogoSec does include plugins that interface with each scanner.

The basic methodology of BogoSec is as follows:

1. Execute each scanner present on target source code or tree
2. Parse output of each scanner, determining the filename, line number, severity, description of each possible vulnerability
3. Interpret the severity indicator and adjust to a common scale (by default, 10 being most severe, 1 being least severe) to calculate "severity points"
4. Report the total number of vulnerability severity points, as well as the total number of lines analyzed by each scanner
5. Calculate and report the BogoSec final score

$$BogoSecFinalScore = \frac{TotalVulnerabilityPointsFromAllScanners}{TotalLinesOfCodeAnalyzedByAllScanners}$$

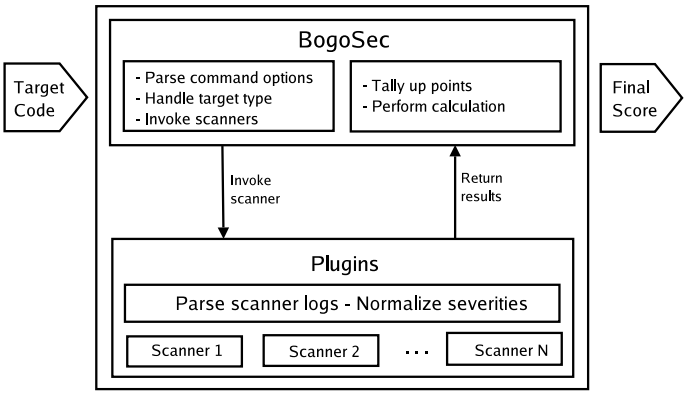

Figure 1: BogoSec Implementation Diagram

The algorithm above considers weighted vulnerabilities. The total number of vulnerability severity points accumulates as scanners identify potential vulnerabilities. The ratio of points per line is the indicator by which users of BogoSec are able to compare source code in terms of security quality. BogoSec operates under the assumption that as the number of weighted severity points per line increases, the overall security quality of the compiled code decreases.

3 Implementation

BogoSec is implemented as a Perl script and accompanying Perl modules. You can add support for additional scanners by creating a module that executes the scanner against the source code, interprets the output, and normalizes severity ratings to BogoSec's scale. The other modules can be used as templates for new modules.

Initially, BogoSec sets up its execution environment by parsing configuration files and reading command line parameters. (See the manpage for an extensive description of command line options.) The final parameter specifies the target; BogoSec handles the following:

- a single source file
- an entire source tree
- an archive (*.tar.gz, *.tgz)
- a source RPM (*.src.rpm)

BogoSec handles the last two formats by creating a temporary directory and expanding the code accordingly. Additionally, the src.rpm targets are rebuilt using the `%prep` section of the spec file which applies all patches to the source code.

The script builds a list of files in the target tree, consisting of any file case insensitively matching (*.c, *.h, *.cpp, *.c++). BogoSec executes scanners on each file individually. Although some scanners have the ability to recursively scan an entire tree, the stability and consistency of BogoSec favors compiling the results of individually scanned files. BogoSec is multithreaded such that all three scanners execute simultaneously on the list of files, waiting for the slowest scanner to complete execution before analyzing the results.

False positives are one of the main discourging factors against running source code scanners. BogoSec implements a mechanism to exclude vulnerabilities in an effort to reduce false positives. Using command line options, users may specify a list of vulnerabilities to be excluded from the final calculation. Additionaly, while parsing each scanner's output, BogoSec keeps a running total of all the vulnerability types encountered and reports those findings if you choose to run in verbose mode.

Each scanner's plugin has a routine defined for analyzing the particular output of that scanner. This consists of scanning each line of output to determine if the line specifies an identified potential vulnerability. If it does, the filename, line number, severity, and description are parsed out according to rules defining each scanner's output conventions. The severity is scaled according to BogoSec's scale (by default, a 10-point scale). The plugin obtains the number of lines of source code scanned from the output whenever possible. Additionally, BogoSec implements a mechanism to exclude vulnerabilities in an effort to allow the user to reduce false positives. This data is stored in a structure accessible by the main program.

Finally, the main program tallies the number of points accumulated by all scanners on all files, as well as the number of lines of code scanned by all scanners in all files. The BogoSec final score is this quotient, which the script reports and then exits.

4 Output Samples

BogoSec's benefit lies mainly in its ability to simplify the process of understanding other scanner results and distilling the information to metrics. For the purpose of this demonstration, a sample of each scanner's output against the Sendmail-8.13.5 package is shown, followed by BogoSec's results and a snapshot of the BogoSec wrapper against a list of packages.

4.1 Flawfinder

Developed by David Wheeler and released under GPL version 2, Flawfinder is written in Python and uses a built-in database of C/C++ functions with well-known problems to produce a sorted list of hits or potential security flaws. Flawfinder uses a scale of 1-5, with 5 being the maximum level of vulnerability to categorize its hits. The following is an excerpt from Flawfinder's output:

```
Flawfinder version 1.26, (C) 2001-2004 David A. Wheeler.
/tmp/bogosec.temp_target.DTtf4m/sendmail-8.13.5/include/sm/io.h:141:  [2]
(buffer)
char:  Statically-sized arrays can be overflowed.  Perform bounds checking,
use functions that limit length, or ensure that the size is larger than the
maximum possible length.
...
/tmp/bogosec.temp_target.DTtf4m/sendmail-8.13.5/include/sm/string.h:53:  [4]
(buffer)
strcpy:  Does not check for buffer overflows when copying to destination.
Consider using strncpy or strlcpy (warning, strncpy is easily misused).
...
/tmp/bogosec.temp_target.WOEdpn/sendmail-8.13.5/mailstats/mailstats.c:74:
[3] (buffer)
getopt:  Some older implementations do not protect against internal buffer
overflows .  Check implementation on installation, or limit the size of all
string inputs.
...
/tmp/bogosec.temp_target.WOEdpn/sendmail-8.13.5/libsmutil/safefile.c:495:
[5] (race)
readlink:  This accepts filename arguments; if an attacker can move those
files or change the link content, a race condition results.  Also, it does
not terminate with ASCII NUL. Reconsider approach.
```

4.2 RATS

Rough Auditing Tool for Security was developed by a team from Secure Software Solutions and is licensed under the GPL. RATS is capable of scanning C, C++, Perl, PHP and Python code. The tool uses a High, Medium, and Low rating to categorize its hits. The following is an excerpt from RATS' output:

```
Analyzing /tmp/bogosec.temp_target.DTtf4m/sendmail-8.13.5/libsm/b-strl.c
/tmp/bogosec.temp_target.DTtf4m/sendmail-8.13.5/libsm/b-strl.c:70:  High:
fixed size local buffer
Extra care should be taken to ensure that character arrays that are allocated
on the stack are used safely.  They are prime targets for buffer overflow
attacks.
/tmp/bogosec.temp_target.DTtf4m/sendmail-8.13.5/libsm/b-strl.c:73:  High:
getopt
Truncate all input strings to a reasonable length before passing them to
this function
/tmp/bogosec.temp_target.DTtf4m/sendmail-8.13.5/libsm/b-strl.c:103:  Low:
strlcpy
Double check that your buffer is as big as you specify
...
/tmp/bogosec.temp_target.WOEdpn/sendmail-8.13.5/libsm/test.c:83:  High:  fprintf
Check to be sure that the non-constant format string passed as argument 2
to this function call does not come from an untrusted source that could have
added formatting characters that the code is not prepared to handle.
...
/tmp/bogosec.temp_target.WOEdpn/sendmail-8.13.5/libsm/sscanf.c:85:  Low:  strlen
This function does not properly handle non-NULL terminated strings.  This
does not result in exploitable code, but can lead to access violations.
```

4.3 ITS4

It's The Software Stupid Source Scanner (ITS4) is developed by Cigital and it scans C and C++ source code. Note that ITS4 is not released under an OSI-approved open source license. ITS4 reads a vulnerability database from a text file at startup, which allows for additions of new vulnerabilites easily. ITS4 uses a Low Risk, Some risk, Risky, Very Risky, and Urgent scale to categorize its hits. The following is an excerpt from ITS4's output:

```
/tmp/bogosec.temp_target.DTtf4m/sendmail-8.13.5/include/sm/string.h:59:(Very
Risky) strcpy
This function is high risk for buffer overflows
Use strncpy instead.
...
/tmp/bogosec.temp_target.DTtf4m/sendmail-8.13.5/libsm/b-strl.c:198:(Urgent)
printf
Non-constant format strings can often be attacked.
Use a constant format string.
...
/tmp/bogosec.temp_target.DTtf4m/sendmail-8.13.5/libsm/stdio.c:132:(Some risk)
read
Be careful not to introduce a buffer overflow when using in a loop.
Make sure to check your buffer boundries.  ...
/tmp/bogosec.temp_target.WOEdpn/sendmail-8.13.5/libsm/ungetc.c:72:(Low Risk)
memcpy
Low risk of buffer overflows.  Make sure that your buffer is really big enough
to handle a max len string.
/tmp/bogosec.temp_target.WOEdpn/sendmail-8.13.5/libsm/stdio.c:352:(Risky)
fstat
Can lead to process/file interaction race conditions (TOCTOU category C)
Manipulate file descriptors, not symbolic names, when possible.
```

4.4 BogoSec

The scanners discussed previously in this paper give valuable insights into potential security vulnerabilities. Some of them even offer suggestions on how to eliminate those vulnerabilities. It is necessary to look over these outputs if you need to understand the vulnerabilities in detail in order to fix them. On the other hand, developers often need a quick gauge of their code, or administrators need a simple comparison mechanism to aid them in making a software choice. BogoSec provides concise output, indicating the scanners used, the severity points, and lines scanned for each scanner separately. BogoSec calculates the total points, total lines scanned and finally the BogoSec score. The following is a sample of the BogoSec output:

```
bogosec sendmail.8.13.5.tar.gz
Running flawfinder...
Running rats...
Running its4...
flawfinder
4210 points
88100 lines
rats
7943 points
117742 lines
its4
4386 points
128906 lines
>>> Using scanners:  (flawfinder rats its4 )
>>> 16539 total severity points
>>> 334748 total lines of code scanned
>>> final score = 0.0494083111275749
```

4.5 BogoSec Wrapper

Early detection of security vulnerabilities helps with their timely resolutions before they pose a potential threat. System administrators often perform routine checks of their systems by running overnight scripts and analysis tools to find and remedy any potential problems. BogoSec wrapper is designed to aid in this effort by providing a mechanism for running BogoSec on a large sum of packages automatically. The wrapper is capable of handling a large number and mixture of target types supported by BogoSec. Results of the wrapper tool are saved in a tabular, easy-to-read format. One file contains only the total points, lines, and score for each scanned target, and a second file contains detailed BogoSec output showing the breakup of each scanner. For added flexibility, BogoSec wrapper accepts command line options that provide output file names, as well as command options specific to BogoSec. The following is a sample of the wrapper output file:

```
START : Mon Jan 09 15:57:00 CST 2006
=====================================
 Package                    SevPoints  Lines Of Code  Final Score
 4Suite-1.0-3.src.rpm       20377      133664         0.1524519192
 acpid-1.0.3-2.src.rpm      896        4269           0.2098852190
 alsa-lib-1.0.6-3.src.rpm   10862      227617         0.0477205129
 am-utils-6.0.9-10.src.rpm  10149      129569         0.0783263486
 anacron-2.3-32.src.rpm     617        4617           0.1337087574
 mingetty-1.07-3.src.rpm    451        1194           0.377442769402568
 apmd-3.0.2-24.src.rpm      2410       9250           0.260540540540541
 rwho-0.17-22.src.rpm       1399       3859           0.362615530793815
 cracklib-2.7-29.src.rpm    1900       6997           0.271544947834786
 inn-2.3.5-12.src.rpm       77187      249509         0.309354238390866
 .  .  .
```

5 Testing

To verify the operation of BogoSec and the reliability of its results, test cases have been executed, documented, and studied. BogoSec was executed on several popular packages against all released versions available for download. These tests demonstrate BogoSec's use against a given package to indicate the general trend of the quality of source code over subsequent releases. Also, several equivalent open source software packages (when available) were tested to compare in an absolute sense which of the packages have better BogoSec scores.
The following popular packages tested across released versions:

- Web server
  - Apache
- Secure Shell
  - OpenSSH
- FTP servers
  - vsftpd
  - wu-ftpd
- Mail transfer agents
  - Sendmail
  - Qmail
  - Postfix
- Scripting languages
  - Perl
  - PHP
  - Python
  - Ruby

5.1 Web Servers

Apache is the world's most popular Web server, accounting for over 60% of all internet Web sites. The Apache 1.3 tree has been under constant development since 1998 and up to the present. In that time, the Apache team has added many features and fixed many bugs. Apache 2.0 is the next generation Web server from the Apache development team.

5.1.1 Results

These test results show consecutive runs of BogoSec against all Apache 1.3 (in maroon) and 2.0 (in blue) released versions available.

Absolute Points: The overall scores of 1.3 are very slowly, but smoothly, increasing. On the other hand, the tremendous spike between 2.0.18 and 2.0.28 (both beta releases) should cause some concern. Also, it looks like the 2.0.44 release fixed some security problems present in previous releases.

Points / Line: Both versions exhibit BogoSec ratios that are generally improving over time, though the 2.0 releases have better scores than the 1.3 branch.

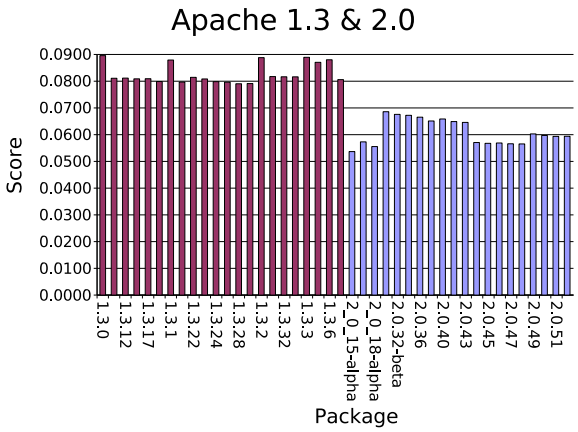

Figure 2: Web Servers Score

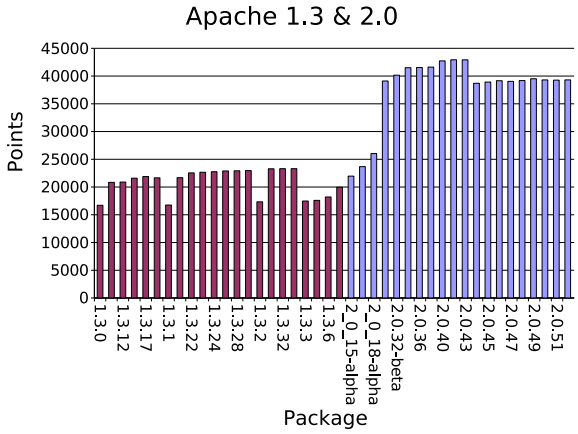

Figure 3: Web Servers Points

Package	Version	Score	LOC	Points	Package	Version	Score	LOC	Points
apache	1.3.0	0.0896	186578	16711	httpd	2_0_15	0.0537	409238	21965
apache	1.3.11	0.0811	256983	20841	httpd	2_0_16	0.0573	413056	23673
apache	1.3.12	0.0812	257446	20895	httpd	2_0_18	0.0555	468691	26035
apache	1.3.14	0.0809	267005	21595	httpd	2_0_28	0.0686	570302	39112
apache	1.3.17	0.0809	270501	21889	httpd	2.0.32	0.0676	594210	40162
apache	1.3.19	0.0799	271083	21654	httpd	2.0.35	0.0672	617261	41503
apache	1.3.1	0.0879	190542	16753	httpd	2.0.36	0.0665	624265	41540
apache	1.3.20	0.0797	272119	21681	httpd	2.0.39	0.0651	639129	41620
apache	1.3.22	0.0814	276845	22546	httpd	2.0.40	0.0659	648812	42737
apache	1.3.23	0.0808	280312	22660	httpd	2.0.42	0.0649	661539	42933
apache	1.3.24	0.0798	285126	22761	httpd	2.0.43	0.0646	664596	42925
apache	1.3.27	0.0796	287594	22900	httpd	2.0.44	0.0571	678224	38701
apache	1.3.28	0.0790	290025	22923	httpd	2.0.45	0.0568	685602	38918
apache	1.3.29	0.0791	290240	22965	httpd	2.0.46	0.0569	688417	39160
apache	1.3.2	0.0888	195124	17329	httpd	2.0.47	0.0566	689951	39044
apache	1.3.31	0.0817	284938	23287	httpd	2.0.48	0.0565	693187	39191
apache	1.3.32	0.0816	285458	23305	httpd	2.0.49	0.0603	655544	39517
apache	1.3.33	0.0816	285501	23305	httpd	2.0.50	0.0597	658322	39305
apache	1.3.3	0.0889	196542	17479	httpd	2.0.51	0.0594	661497	39268
apache	1.3.4	0.0871	202112	17596	httpd	2.0.52	0.0594	661740	39308
apache	1.3.6	0.0880	206900	18209					
apache	1.3.9	0.0806	247724	19963					

5.2 Secure Shell Servers

OpenSSH provides an encrypted command shell, usually for remote network access to systems. OpenSSH is primarily developed by members of the OpenBSD project, a group of developers known for security-conscious code. OpenBSD conducts extensive manual audits of source code to identify and fix security vulnerabilities.

5.2.1 Results

The following graphs demonstrate admirable models for secure software development. Both sets of data seem to approach asymptotes, with Absolute Points gradually increasing, and Points/Line scores gradually decreasing, and neither have significant spikes. The false positives reported by the tools could well form the asymptote base of these graphs.

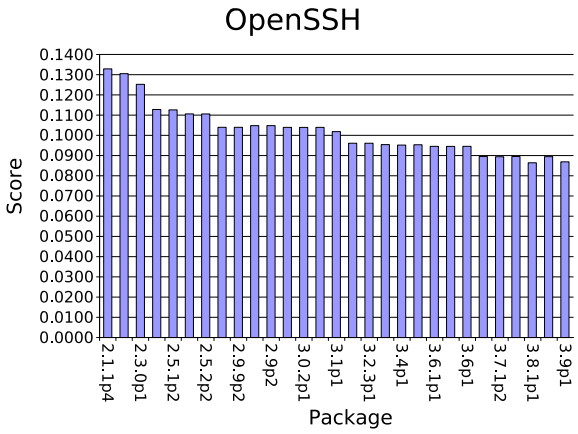

Figure 4: Shell Servers Score

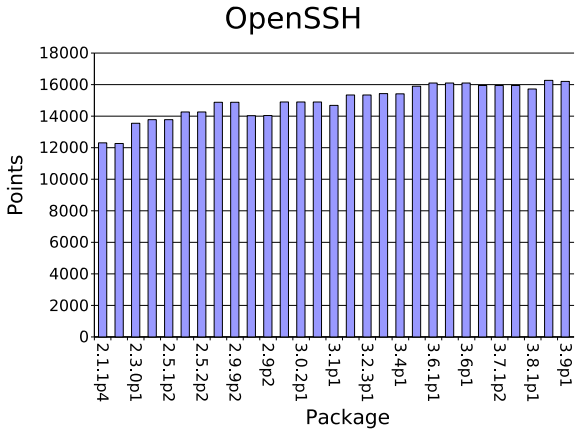

Figure 5: Shell Servers Points

Package	Version	Score	LOC	Points	Package	Version	Score	LOC	Points
openssh	2.1.1p4	0.1329	92619	12307	openssh	3.2.2p1	0.0961	159646	15342
openssh	2.2.0p1	0.1305	93999	12267	openssh	3.2.3p1	0.0961	159660	15342
openssh	2.3.0p1	0.1252	108205	13549	openssh	3.3p1	0.0954	161720	15430
openssh	2.5.1p1	0.1128	122159	13778	openssh	3.4p1	0.0952	161964	15416
openssh	2.5.1p2	0.1126	122387	13778	openssh	3.5p1	0.0953	166756	15899
openssh	2.5.2p1	0.1106	128989	14266	openssh	3.6.1p1	0.0945	170301	16101
openssh	2.5.2p2	0.1106	129019	14266	openssh	3.6.1p2	0.0945	170316	16101
openssh	2.9.9p1	0.1040	143107	14880	openssh	3.6p1	0.0946	170270	16101
openssh	2.9.9p2	0.1040	143107	14880	openssh	3.7.1p1	0.0895	178101	15943
openssh	2.9p1	0.1048	133867	14029	openssh	3.7.1p2	0.0894	178288	15943
openssh	2.9p2	0.1048	133972	14040	openssh	3.7p1	0.0895	178084	15943
openssh	3.0.1p1	0.1039	143342	14899	openssh	3.8.1p1	0.0864	181919	15722
openssh	3.0.2p1	0.1039	143351	14899	openssh	3.8p1	0.0895	181852	16269
openssh	3.0p1	0.1039	143319	14897	openssh	3.9p1	0.0869	186467	16203
openssh	3.1p1	0.1018	144189	14683					

5.3 FTP Servers

Vsftpd and wu-ftpd are two major open source FTP servers. The first, vsftpd, was written with security as a primary objective, with the "vs" meaning "very secure". The second, wu-ftpd, is Washington University's FTP server, which pre-dates vsftpd and has a history of security vulnerabilities.

5.3.1 Results

These charts show a drastic difference in source-code security between vsftpd and wu-ftpd. Both BogoSec scores of vsftpd appear orders of magnitude better than wu-ftpd. This is consistent with the popular opinion regarding the security of these two FTP servers.

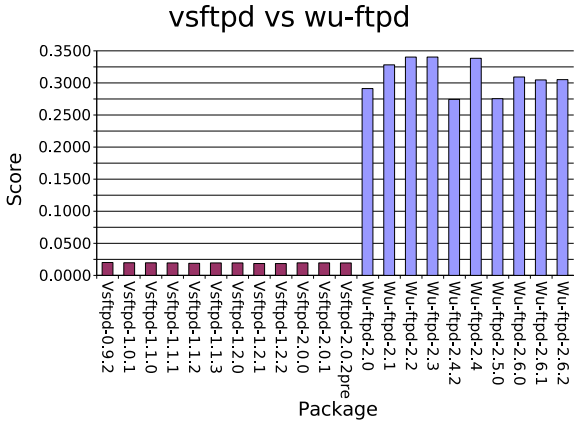

Figure 6: FTP Servers Score

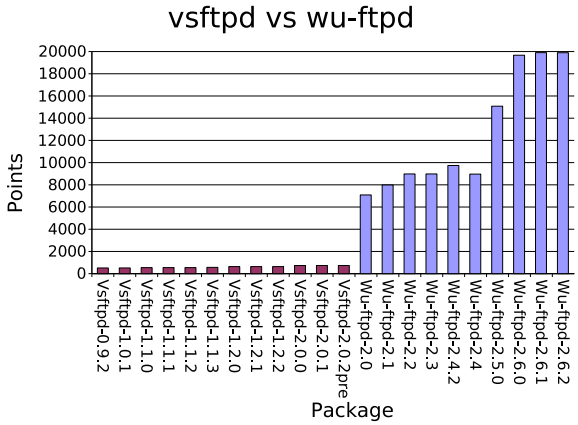

Figure 7: FTP Servers Points

Package	Version	Score	LOC	Points	Package	Version	Score	LOC	Points
vsftpd	0.9.2	0.0201	25656	515	wu-ftpd	2.0	0.2912	24345	7089
vsftpd	1.0.1	0.0197	26149	515	wu-ftpd	2.1	0.3282	24334	7986
vsftpd	1.1.0	0.0196	27784	544	wu-ftpd	2.2	0.3403	26388	8980
vsftpd	1.1.1	0.0194	28029	544	wu-ftpd	2.3	0.3403	26385	8980
vsftpd	1.1.2	0.0188	28868	544	wu-ftpd	2.4.2	0.2743	35509	9741
vsftpd	1.1.3	0.0194	29228	568	wu-ftpd	2.4	0.3384	26495	8966
vsftpd	1.2.0	0.0194	32849	637	wu-ftpd	2.5.0	0.2754	54752	15081
vsftpd	1.2.1	0.0185	34409	637	wu-ftpd	2.6.0	0.3093	63603	19670
vsftpd	1.2.2	0.0185	34540	637	wu-ftpd	2.6.1	0.3047	65322	19903
vsftpd	2.0.0	0.0195	37607	734	wu-ftpd	2.6.2	0.3051	65212	19898
vsftpd	2.0.1	0.0195	37657	734					
vsftpd	2.0.2	0.0194	37883	734					

5.4 Mail Transfer Agents

Postfix, Qmail, and Sendmail are major open source mail transfer agents (MTAs). Sendmail has long served as the primary mailer for UNIX environments, but Sendmail's history is riddled with exploitable security vulnerabilities. Alternative MTA's have emerged, such as Postfix and Qmail, whose purpose was to address Sendmail's security shortcomings. Postfix was written and is maintained by an expert member of IBM Research in secure computing. Qmail is another option with a notably small code base designed in the interest of security.

5.4.1 Results

These results are interesting in that the package that has the lowest absolute scores (Qmail) does not have the lowest points/line ratios. This is because of the significantly smaller code base of Qmail, and perhaps identifies an unfair BogoSec bias toward larger projects. However, this is exactly why both metrics must be considered. The Postfix scores are very good, as expected. A concerted effort by the Sendmail development team between releases 8.11 and 8.12 demonstrates a marked BogoSec score improvement. Since that time, Sendmail has maintained approximately the same status.

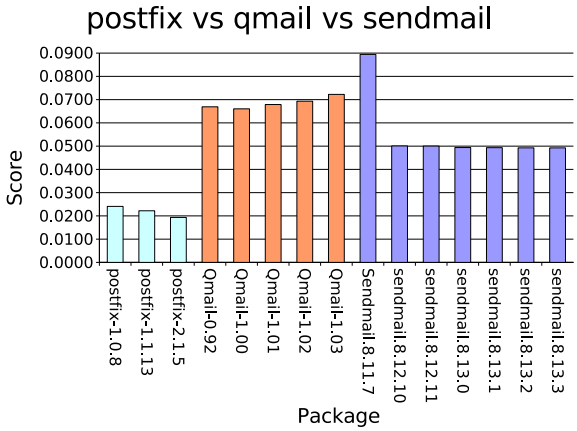

Figure 8: Mail Servers Score

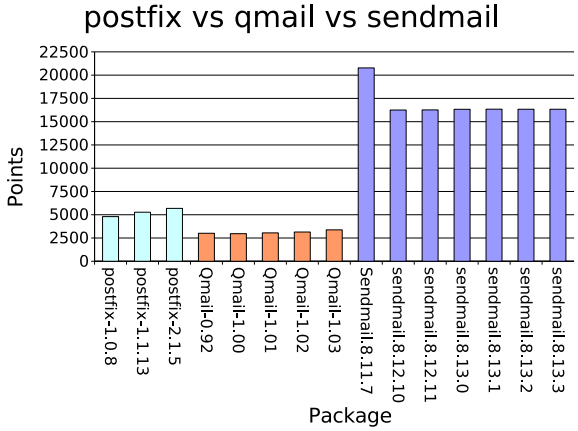

Figure 9: Mail Servers Points

Package	Version	Score	LOC	Points	Package	Version	Score	LOC	Points
postfix	1.0.8	0.0241	199281	4800	sendmail	8.11.7	0.0894	232285	20775
postfix	1.1.13	0.0222	237491	5273	sendmail	8.12.10	0.0501	324450	16251
postfix	2.1.5	0.0193	293908	5683	sendmail	8.12.11	0.0500	325152	16265
qmail	0.92	0.0669	44957	3008	sendmail	8.13.0	0.0494	330588	16330
qmail	1.00	0.0660	44913	2966	sendmail	8.13.1	0.0493	331157	16341
qmail	1.01	0.0679	44918	3050	sendmail	8.13.2	0.0493	331607	16335
qmail	1.02	0.0693	45339	3143	sendmail	8.13.3	0.0493	331647	16335
qmail	1.03	0.0722	46737	3377					

5.5 Scripting Languages

Open source scripting languages include Perl, PHP, Python, and Ruby. Each of these provide a higher level programming language easily used by developers to easily and quickly accomplish software tasks. A binary interpreter is needed by each of these languages to execute the scripted source code.

5.5.1 Results

Ruby and Python have the lowest absolute and ratio scores. PHP, by far, has the largest code base and the highest absolute points. In all cases, the latest release shows marked improvement over the earliest release.

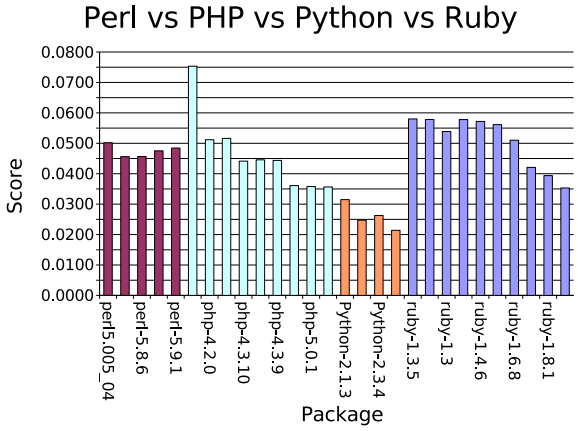

Figure 10: Scripting Languages Score

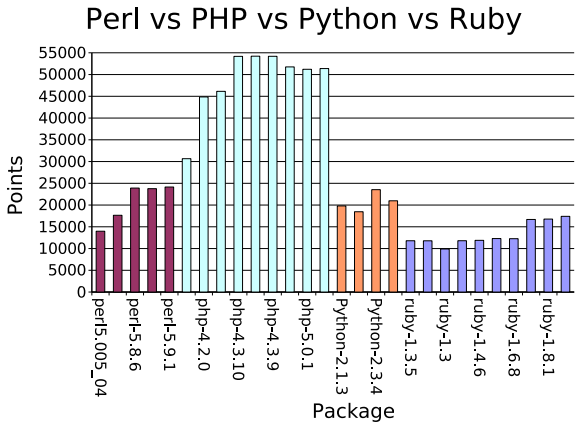

Figure 11: Scripting Languages Points

Package	Version	Score	LOC	Points	Package	Version	Score	LOC	Points
perl	5.005	0.0502	278627	13982	Python	2.1.3	0.0315	628654	19797
perl	5.6.1	0.0456	386938	17649	Python	2.2.3	0.0247	746199	18461
perl	5.8.6	0.0457	523662	23908	Python	2.3.4	0.0263	895668	23539
perl	5.9.0	0.0475	500034	23765	Python	2.4	0.0214	980119	20976
perl	5.9.1	0.0484	498403	24145	ruby	1.3.5	0.0580	203286	11792
php	3.0.18	0.0753	406914	30655	ruby	1.3.7	0.0578	203739	11781
php	4.2.0	0.0512	876430	44854	ruby	1.3	0.0539	183631	9893
php	4.2.3	0.0516	893882	46141	ruby	1.4.0	0.0578	204141	11799
php	4.3.10	0.0441	1227476	54180	ruby	1.4.6	0.0572	207843	11887
php	4.3.8	0.0446	1216570	54224	ruby	1.6.0	0.0561	219104	12299
php	4.3.9	0.0444	1221208	54194	ruby	1.6.8	0.0510	240446	12267
php	5.0.0	0.0361	1433945	51758	ruby	1.8.0	0.0421	396570	16687
php	5.0.1	0.0358	1431329	51212	ruby	1.8.1	0.0394	426031	16778
php	5.0.3	0.0357	1441051	51379	ruby	1.8.2	0.0353	492828	17405

5.6 All Packages Tested

The following data shows an absolute comparison of the latest release of all packages tested above. Be somewhat cautious of this comparison, because it's difficult to compare vastly different packages, especially when one package consists of a few hundred lines of code and another comprises millions of lines of code. Still, this chart teaches some valuable lessons about the advantages of BogoSec as well as, perhaps, some of its shortcomings.

5.6.1 Results

Absolute Points: It seems that vsftpd, Qmail, and Postfix are clear winners–these being packages designed and implemented by experts to be secure software. On the other end of the spectrum, httpd and PHP totaled the highest number of points–probably because of the fact that these are two of the largest packages that were tested.
Points / Line: Again, Postfix and vsftpd exhibited the best scores, while wu-ftpd scored several orders of magnitude worse than the rest of the field–consistent with the popular opinion of its state of security.
All four of the scripting languages (Python, PHP, Ruby, Perl) appear to have relatively similar BogoSec scores. It is somewhat surprising to see Apache, httpd, and OpenSSH near the upper end of the spectrum. This also deserves further investigation as it is possible that these packages are yielding an abnormally high number of false positives, thereby driving their scores disproportionately higher.

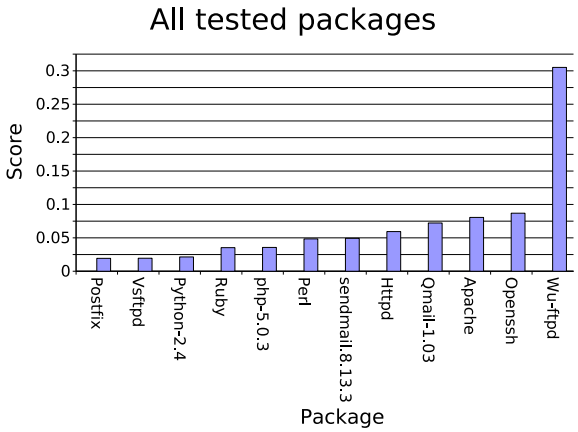

Figure 12: All Packages Score

Figure 13: All Packages Points

Package	Version	Score	LOC	Points
postfix	2.1.5	0.0193	293908	5683
vsftpd	2.0.1	0.0195	37657	734
ython	2.4	0.0214	980119	20976
ruby	1.8.2	0.0353	492828	17405
php	5.0.3	0.0357	1441051	51379
perl	5.9.1	0.0484	498403	24145
sendmail	8.13.3	0.0493	331647	16335
httpd	2.0.52	0.0594	661740	39308
qmail	1.03	0.0722	46737	3377
apache	1.3.9	0.0806	247724	19963
openssh	3.9p1	0.0869	186467	16203
wu-ftpd	2.6.2	0.3051	65212	19898

6 Applications

We hope that BogoSec drives developer awareness of insecure code by providing a higher level interface to numerous scanners. One way of encouraging developers to use these existing tools is by integrating BogoSec into some common development tools and processes, such as:

- source code repositories
- integrated development environments
- text editors
- build processes
- package installation managers

7 Conclusion

BogoSec aims to reduce source code security vulnerabilities by simplifying the process of identifying and eliminating them. The following is a recap of the functionality and uses of BogoSec, as well as future improvements:

- BogoSec provides a convenient interface that should make developers and users more conscious of the security quality of software packages.
- Given a diverse set of software packages, BogoSec is able to accurately identify those with the highest and lowest security quality.
- Given a set of subsequent releases of a software package or a single file, BogoSec is able to chart the security quality progress over time.
- Additional work is necessary to further BogoSec and push for more widespread adoption including:
  - a complexity factor - BogoSec currently seems to prefer packages with a larger code base, which is counterintuitive to the adage that "simplicity is the ally of security."
  - tool integration - BogoSec metric calculation could be integrated into additional tools, such as package installation managers.
  - reduction of "false positive" effects - More skillful use of the actual source-code scanners could produce more accurate vulnerability output.
  - incorporation of more and new scanners.

8 References

- CERT/CC (http://www.cert.org/stats/cert_stats.html)
- IBM
  - IBM (http://www.ibm.com)
  - Linux Technology Center (http://oss.software.ibm.com/linux/)
- BogoSec
  - BogoSec Web site (http://bogosec.sourceforge.net/)
- Scanners
  - Flawfinder (http://www.dwheeler.com/flawfinder/)
  - RATS (http://www.securesoftware.com/rats/)
  - ITS4 (http://www.cigital.com/its4/)
- Packages Tested
  - Apache (http://www.apache.org)
  - OpenSSH (http://www.openssh.org)
  - Perl (http://www.perl.org)
  - PHP (http://www.php.net)
  - Postfix (http://www.postfix.org)
  - Python (http://www.python.org)
  - Qmail (http://www.qmail.org)
  - Ruby (http://www.ruby-lang.org)
  - Sendmail (http://www.sendmail.org)
  - vsftpd (http://vsftpd.beasts.org)
  - wu-ftpd (http://www.wu-ftpd.org)

9 APPENDIX A : BogoSec Manual Page

```
BOGOSEC(1)                BogoSec User Manual                BOGOSEC(1)

NAME
   bogosec - source-code security quality metric using established static source-code
scanners

SYNOPSIS
   bogosec [-l] [--log-dir directory ] [--min-sev 0-10 ] [--nhf] [-p plugin_name [args]
] [--plugin-dir directory ] [--sev-range-max num ] [--timeout num ] [--temp-log-dir
directory ] [-v 0|1 ] [--xp plugin_name ] [--xv vuln_list ] TARGET

DESCRIPTION
   BogoSec attempts to influence developers to produce more secure source-code over
time.  Various existing scanners point developers to potentially insecure sections
of code.  BogoSec broadens the scope of source-code scans by utilizing multiple independent
scanners and compiling the results into high level calculated metrics.  These metrics
can help developers and users alike to comparatively judge the security quality of
source-code.

OPTIONS
   -l
   Turn on scanner output logging.  Log will be called <scanner_name>.log and created
in current working directory, unless --log-dir is used to specify a different location.

   --log-dir directory
   Specify a directory for scanner output logs (only makes sense if -l is also used).
Default is current working directory.

   --min-sev minimum_severity_level
   Specify a minimum severity level.  Any vulnerabilities reported by the scanners
whose score falls below this number will be ignored.  The argument must be a number
0-10.  Default is 0.

   --nhf, --no-header-files
   Do not scan header files.  Useful if the scanners being used do not support scanning
header files.

   -p, --plugin plugin_name [args]
   Specify a plugin to use.  If no plugins are defined on the command-line, all of
the plugins in the plugins_dir will be used.  This option can be passed more than once,
to specify a set of scanners to use.  Each scanner requires a separate instance of
the --plugin flag (please see examples).  Optionally, a set of command-line arguments
can be passed to the scanner - this feature must be used with care.  Keep in mind that
the plugin requires a certain formatting of the scanner output (for example, -SQ is
always passed to flawfinder, and -w 3 is always passed to rats).  You can pass additional
command-line arguments using this option, but be aware of the effect it might have
on the formatting of the scanner output, and the effect that will have on the plugins
ability to parse it correctly.  If you must change the defaults (-SQ, -w 3, etc.)  you
```

must edit the plugin directly.

--plugin-dir directory
Specify the directory where the plugins are stored. Default is /usr/local/bogosec/plugins.

--sev-range-max number
Specify the maximum severity value to be used in calculating the severity value range. The default is 10. For example, setting --sev-range-max to 50 would mean that the severity results would now be on a scale of 0-50 instead of on a scale of 0-10. This can be used to scale the result if more granularity is required. NOTE: -v 1 will not work if this option is used.

--timeout number
Specify the cpu time limit in seconds. Some scanners might hang, in order to overcome this problem you may choose to set the timeout to an appropriate period to kill the scanner process. For example setting --timeout 60, will kill any remaining scanner processes after 60 seconds, and return control to the main bogosec process. This option uses the ulimit command, please refer to ulimit manpage for additional information.

--temp-log-dir directory
Specify a directory where you want the temporary files used by BogoSec to be stored (scanner output logs, etc.) The default is /tmp/.

-v, --verbosity 0|1
Specify verbosity level (default is 0). If 1, then a graph of the severity points is shown, which breaks the results down by severity levels. This option does not work if the --sev-range-max is changed from 10.

--xp, --exclude-plugin plugin_name
Do not run plugin defined by plugin_name.

--xv, --exclude-vuln vuln_list
Exclude the vulnerabilites in the vuln_list from the final bogosec calculation. vuln_list is a ":" separated list of vulnerability identifiers.

TOOLS
bogosec_wrapper provides a method to run bogosec automatically on a directory containing multiple targets. Please refer to bogosec_wrapper man page for additional information.

FILES
/etc/bogosec.conf
Global configuration file. The settings here are overwritten by any settings in users /.bogosecrc file.

/.bogosecrc
Default user configuration file (overrides the settings in /etc/bogosec.conf). This file is not created during an installation, you must create it yourself.

/usr/local/bogosec/plugins/

Default plugins directory. Can be changed with --plugin-dir option. Plugins must be executable, and must end in .pm as per convention.

/usr/local/bogosec/documents/
Directory of BogoSec documentation and other germane documents.

SCANNERS

FlawFinder : http://www.dwheeler.com/flawfinder
ITS4 : http://www.cigital.com/its4
RATS : http://www.securesoftware.com/resources/tools.html

BUGS

Not all input validated. Not all environmental variables checked. This program expects to be run by trusted users.

AUTHORS

Developed by Dustin Kirkland, Agoston Petz, and Loulwa Salem at the IBM Linux Technology Center.

http://sourceforge.net/projects/bogosec/

Linux Jan 25 2005 BOGOSEC(1)

10 APPENDIX B: BogoSec Wrapper Manual Page

```
BOGOSEC(1)              BogoSec Wrapper User Manual              BOGOSEC(1)

NAME
   bogosec_wrapper - Wrapper script for BogoSec source-code security quality metric
tool

SYNOPSIS
   bogosec_wrapper [OPTIONS] TARGET-DIRECTORY

DESCRIPTION
   bogosec_wrapper automates the process by running bogosec on a directory containing
different file formats supported by bogosec and collecting the results.  Results by
default are collected in the following files:  /tmp/bogosec-results.<timestamp> and
/tmp/bogosec-details.<timestamp> (where timestamp is the current system time in HourMinSec
format).  Users can specify different destination files on the command line.
   bogosec_wrapper accepts options to be passed on to bogosec, options are passed in
"" with the flag --bo.  (example:  bogosec_wrapper --bo "--nhf --timeout 60" /Target_Directory).


WRAPPER OPTIONS

   --rf, results_file file-name
   Specify results file

   --df, details_file file-name
   Specify detailed results file

   --bo, bogo_opts bogosec options
   Specify bogosec options (must be included in "")

BOGOSEC OPTIONS
   The following is a brief explanation of bogosec options; for additional information,
please refer to bogosec man page.
   -l
   Turn on scanner output logging.

   --log-dir directory
   Specify a directory for scanner output logs

   --min-sev minimum_severity_level
   Specify a minimum severity level.

   --nhf, --no-header-files
   Do not scan header files.

   -p, --plugin plugin_name [args]
   Specify a plugin to use.
```

```
   --plugin-dir directory
   Specify the directory where the plugins are stored.  Default is /usr/local/bogosec/plugins.

   --sev-range-max number
   Specify the maximum severity value to be used in calculating the severity value
range.  The default is 10.

   --timeout number
   Specify the cpu time limit in seconds.

   --temp-log-dir directory
   Specify a directory where you want the temporary files used by BogoSec to be stored
(scanner output logs, etc.)  The default is /tmp/.

   -v, --verbosity 0|1
   Specify verbosity level.  The default is 0.

   --xp, --exclude-plugin plugin_name
   Do not run plugin defined by plugin_name.

   --xv, --exclude-vuln vuln_list
   Exclude the vulnerabilites in the vuln_list

FILES
   Default:  /tmp/bogosec-results.<timestamp>
   Default:  /tmp/bogosec-details.<timestamp>


AUTHORS

Developed by Dustin Kirkland, Agoston Petz, and Loulwa Salem at the IBM Linux Technology
Center.

   http://sourceforge.net/projects/bogosec/
Linux               Mar 07 2005                      BOGOSEC(1)
```

Legal Statement

This work represents the view of the author and does not necessarily represent the view of IBM.

IBM, IBM (logo), e-business (logo), pSeries, e (logo) server, and xSeries are trademarks or registered trademarks of International Business Machines Corporation in the United States and/or other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product, and service names may be trademarks or service marks of others.