

# Survey on Secure Web Mail System and File Sessions

Chethan K S<sup>1</sup>, Ravi Kumar V G<sup>2</sup>

<sup>1,2</sup> Department of CSE

<sup>1,2</sup> GSSSIETW, Mysuru

**Abstract-** *In this paper we study the evolution of C programming in the Unix operating system. We extract, aggregate, and synthesize metrics from snapshots obtained from an artificial software configuration management repository tracking the evolution of the Unix operating system over four decades. This paper describes a new technique for analyzing dynamic file usage patterns based upon classification of file sessions. A file session is defined to be the set of operations on a given file from the moment it is opened until the moment it is closed. A safe and stable webmail is a major component of office automation. However because of the low security, its development in E-commerce and E-government is limited greatly. This text introduces the technical framework and working principles of a secure webmail system based on FreeBSD UNIX together with the problems of stability and security of an enterprise mail system.*

**Keywords-** FreeBSD UNIX, webmail

## I. INTRODUCTION

Tracking long-term progress in engineering allows us to take stock of things we have achieved, appreciate the factors that led to them, and set realistic goals for where we want to go. Specific factors that drive long term progress in programming practices include the affordances and requirements of computer architecture, programming languages, development frameworks, and compiler technology, the ergonomics of interfacing devices, programming guidelines, processing memory and speed, and social conventions. The objective of this work is to study, in view of these factors, the evolution of programming practice in the context of the Unix operating system. Given the hypothesis that the structure and internal quality attributes of a working, non-trivial software artifact will represent first and foremost the engineering requirements of its construction [1], the results can also indicate areas where developers rationally

allocated improvement effort and areas where developers did not see a reason to invest. This paper

describes a technique for organizing file system trace data in a way that highlights the distinctions between different styles of file use. The technique revolves around the concept of a file session. A file session is defined as the entire set of

operations on a given file between the open operation and the corresponding close operation. By analyzing this set of operations as a group, it is possible to discover patterns which are not otherwise obvious. For example, a given session might open a file for writing, position the write head at the current end of the file, write some data sequentially, and then close the file. The rapid development of modern network technology, in particular, the increasing popularity of the Internet promotes all kinds of network application services [2]. The email is an important service of Internet, and it being a threat to security is also growing. In recent years, Webmail has good usability and manageability, but its safety is not high, therefore it has not been massively applied. This paper will introduce a high safety, high stability, easy management, easy to use and has the anti-spam and anti-virus function of large security Web Mail system [3].

## II. METHODS

Our study is based on a synthetic software configuration management repository tracking the long term evolution of the Unix operating system[16]. At successive time points of significant releases we process the source code with a custom-developed tool to extract a variety of metrics for each file. We then synthesize these metrics into values that are related to the internal code quality of the whole system, and analyze the results over time using established statistical techniques. The primary sources of the material include source code snapshots of early released versions, which were obtained from the Unix Heritage Society archive, the CD-ROM images containing the full source archives of Berkeley's Computer Science Research Group (CSRG), the Old Linux site, and the FreeBSD archive. These snapshots were merged with the CSRG SCCS repository, the FreeBSD 1 CVS repository, and the Git mirror of modern FreeBSD development. This material formed the basis for constructing a synthetic Git repository, which allows the efficient retrieval and processing of the Unix source code covering a period of 44 years [4]. We addressed the difficulty of parsing C source code without access to the original compilation environment by extending and using our `cmcalc1` open source tool, which efficiently calculates a variety of C code quality metrics, without requiring full access to the compilation environment's parameters. The tool's operation is based on state machine logic [5], and will therefore produce reasonably accurate results without requiring access to header files and the like. The `cmcalc` tool calculates size, language

feature, code style, and commenting metrics; see the tool's documentation and reference [6] for more details.

Analysis was performed on three traces of UNIX system activity collected by Ousterhout and his students at Berkeley in the spring of 1985 [7]. The traces were produced by logging every system call which affected the file system.

The following log entries were of interest to us [15].

- File opens and creates, which mark session beginnings. There are also two operations which may be performed as side effects when the file is opened: the file may be truncated to zero length (truncate mode) or the write head may be positioned to the end of the file (append mode).
- File closes, which mark session ends. The read/write head position is recorded when the file is closed.
- Seek operations, which explicitly change the position of the file's read/write head; both the old and new head positions are recorded
- Read and write data transfer operations were not recorded on the log, but could be inferred by examining changes in the file's read/write head position.
- Truncate operations on open files, which change the file's length.

Unlike Ousterhout, we were not interested in the file reads corresponding to program loading (e.xecs), since we assumed that frequently executed programs would be resident on the workstation's disk. More detailed information about the data maintained in the trace log may be found in Ousterhout's paper[10]. The three machines from which the traces were taken were used primarily for document preparation and program development (the machines named A5 and E3) and computer-aided design (machine C4). Analysis was performed on the same three traces analyzed by Ousterhout to allow easy comparison of our results with his. Each trace covers approximately three weekdays and contains between 733 000 and just over a million event records which constitute between 233 000 and 358 000 complete file sessions [11]. Because the system was not quiescent when tracing was started and stopped, each trace also contains a tiny number of incomplete sessions, which were ignored. Before analysis is begun, our technique requires that one postulate a set of access-style categories, using intuition and observations. It may be necessary to repeat this process several times to develop an appropriate set of categories.

The final set of categories used for this analysis was as follows[15].

- Read Only: The file is not modified
- NewData: The file is created from scratch or by completely overwriting an existing file. The latter may occur

if the file is written sequentially from the beginning past its previous end-of-file, or if it is truncated to zero length and before data is written into it. Either way, none of the old contents of the file is retained.

- Modified: The file is modified in some arbitrary way. Database updates would fall into this category. All sessions in which the file is read as well as written were placed in this category.
- Flag: No data is written. The file starts empty, ends empty, and is empty in between.
- Append: New data is added to the end of the file. The old contents of the file remain untouched.
- Delete Body: The file is truncated to zero length and left empty.
- Temp: The file starts empty and ends empty but some data resides in the file in between. (No sessions of this type were encountered.)

Flag sessions result from the use of the file system for synchronization. Older versions of UNIX did not provide file locks as a primitive, and some applications use the existence of a file with a certain name as a lock. (No attempt is ever made to read the flag file itself; it exists solely to cause certain operations on the directory to fail.) Although we were familiar with this locking convention, we did not expect to find that it was used so frequently[12]. We made flag sessions a separate category when we discovered a large number of files with zero length in the Modified category. The opposite occurred with Temp sessions: sessions matching this pattern were expected but not observed. It is probable that applications which create a temporary file close the file with data still in it and then delete the file. Since no temp sessions were encountered they will not be mentioned again[13].

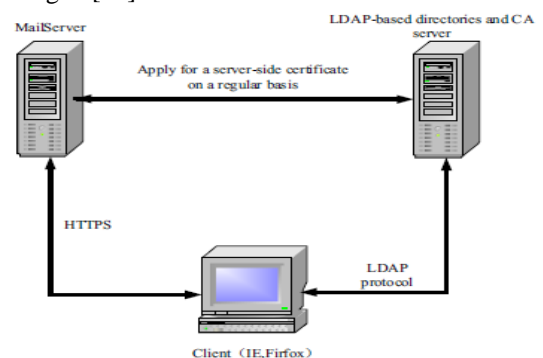


Fig 1: WebMail System

WebMail is a mail system software technology that allows users through a ordinary browser (such as IE, Firefox) to access their mailboxes, its mail processing functions are done on the server side [8], and it is provided to the mail user by the form of WEB services as shown in fig 1,[14] with convenient installation and maintenance features. The e-mail system that

we have developed has the following technical characteristics: using SSL protocol for encryption and decryption in the data transmission course to ensure the security of user information and the message in the transmission [9]; using digital signature mechanism to ensure message integrity, confidentiality, and non-repudiation; using artificial intelligence technology to realize the anti-spam and antivirus; With the convenience of centralized management function, including the domain name management, user management, certificate management etc.; mailbox user database and system user database separate management to improve the security of operating system. The system mainly uses B / S model, its biggest advantage is easy to install, easy to maintain, and compatible with C / S mode, which can still ensure the normal use of other client tools, such as Foxmail, Outlook.

## VI. CONCLUSION

A safe and stable webmail is a major component office automation. Here we analyze technical framework and working principles of a secure webmail system based on Free BSD UNIX and examine the overall pattern of file to analyze file system dynamics.

## REFERENCES

- [1] D. Spinellis, "A tale of four kernels," in ICSE '08: Proceedings of the 30th International Conference on Software Engineering, W. Schaefer, M. B. Dwyer, and V. Gruhn, Eds. New York: Association for Computing Machinery, May 2008, pp. 381–390.
- [2] Kaushik, S., Wijesekera, D., and Ammann, P.: 'BPEL Orchestration of secure WebMail', in Editor (Ed.) (Eds.): 'Book BPEL Orchestration of secure WebMail' (ACM, 2006, edn.), pp. 85-94
- [3] Taylor, B.: 'Sender reputation in a large webmail service', in Editor (Ed.) (Eds.): 'Book Sender reputation in a large webmail service' (2006, edn.), pp.125-127.
- [4] "A repository with 44 years of Unix evolution," in MSR '15: Proceedings of the 12th Working Conference on Mining Software Repositories. IEEE, 2015, pp. 13–16.
- [5] "Tools and techniques for analyzing product and process data," in The Art and Science of Analyzing Software Data, T. Menzies, C. Bird, and T. Zimmermann, Eds. Morgan-Kaufmann, 2015, to appear.
- [6] S. H. Kan, Metrics and Models in Software Quality Engineering, 2nd ed. Boston, MA: Addison-Wesley, 2002.
- [7] Ousterhout, J. K., Da Costa, H., Harrison, D., Kunze, J. A., Kupfer, K. and Thompson, J. G. "A Trace-Driven Analysis of the UNIX 4.2 BSD File System". Proc. 10th ACM Symp. on Operating System Prm., December 1985, pp 1 5-24.
- [8] Schatzmann, D., Mühlbauer, W., Spyropoulos, T., and Dimitropoulos, X.: 'Digging into HTTPS: flow-based classification of webmail traffic', in Editor (Ed.) (Eds.): 'Book Digging intoHTTPS: flow-based classification of webmail traffic' (ACM, 2010, edn.), pp. 322-327
- [9] Yi, Y.H.G.C.S.: 'An Email System Based on Linux and Qmail', Microcomputh Applications, 2001, 6, pp. 109
- [10] Burriss, C. H. "Selection Matrices: An Algebraic System for Representing File Versions " MS Thesis (in preparation), University of Washington, Computer Science Dept.
- [11] Lazowska, E . D., ZahOJjan, J . , Cheriton, D. R. and Zwaenepoel, W. "File Access Performance of Diskless Workstations" . Trans. Computer Systems 3, Nr 3 (August 1986), pp238-268.
- [12] Morris, J. H., Satyanarayanan, M., Conner, M., Howard, J., Rosenthal, D. and Smith, F. D. "Andrew: A Distributed Personal Computing Environment". Comm. of the ACM 28, Nr 3 (March 1986), pp 1 84-201.
- [13] Ousterhout, J. K., Da Costa, H., Harrison, D., Kunze, J. A., Kupfer, K. and Thompson, J. G. "A Trace-Driven Analysis of the UNIX 4.2 BSD File System". Proc. 10th ACM Symp. on Operating System Prm., December 1985, pp 1 5-24.
- [14] Zhang, Yan, and Yuping Yang. "A Secure Webmail System Based on FreeBSD UNIX." 2013 6th International Conference on Intelligent Networks and Intelligent Systems (ICINIS). IEEE, 2013.
- [15] Maloney, John H., and Andrew P. Black. "Sessions: A technique and its application to the UNIX file system." Data Engineering, 1987 IEEE Third International Conference on. IEEE, 1987.
- [16] Spinellis, Diomidis, Panagiotis Louridas, and Maria Kechagia. "An exploratory study on the evolution of C programming in the Unix operating system." 2015 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM). IEEE, 2015.