INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

A COMPLETE SECURED CRYPTOGRAPHIC FRAMEWORK FOR CLOUD COMPUTING ENVIRONMENTS

M. K. DESHMUKH[1], DR. H. R. DESHMUKH[2]

1. Department of Computer Engineering, Manav School of Engg. & Tech., Vyla, Akola, Maharashtra, India.
2. Department of Computer Science & Engineering, Dr. Rajendra Gode Institute of Technology & Research, Amravati, Maharashtra, India.

Accepted Date: 15/03/2016; Published Date: 01/05/2016

Abstract: From the last two decades, Governments, international organizations, private companies and individuals are investing a great deal of time, efforts and budgets to install and use various security products and solutions. However, in spite of all these needs, activities, on-going efforts, and all current solutions, it is general belief that the security in today networks and applications is not adequate. We are daily witnessing various problems – infection of computers by malware, distribution of E–mail spam, phishing of Web pages, penetrations by hackers, software bugs, stolen industrial secrets and credit cards, disclosure of sensitive documents, and so on. There are two general approaches to network applications security, one approach is based on isolation, that is protecting them by isolating operational environments at their periphery using firewalls, port scanners, intrusion–detection tools, spam and phishing filters, "demilitarized zones", Email spam filters, etc. and the another approach is called software security which is based on methodology to create secure, robust and protected applications, bug–free and not vulnerable to attacks, by using well–established methodology for design of applications, software tools for their development, and testing methodology and environments for their debugging and testing. Our new security framework is based on our generic security objects, well established secure technologies and security standards. Therefore, the components designed based on this methodology will also be generic and compliant with security standards. The core components of our security system are Security Provider, Secure Execution Environment, and Security Protocols. They contain security engines of our security system, where each component provides the same set of tested security services. These components are complete with respect to their functionality, so developers can use these components to extend their applications with security features.

Keywords: Security, Cloud Computing, Cryptography, Web System

PAPER-QR CODE

Corresponding Author: MR. M. K. DESHMUKH

Access Online On:

www.ijpret.com

How to Cite This Article:

M. K. Deshmukh, IJPRET, 2016; Volume 4 (9): 467-479

INTRODUCTION

There were many activities and contributions in the last two decades to protect data, messages and other resources in computer networks, to provide privacy of users, reliability, availability and integrity of resources, and to provide other security properties for network environments and applications. Governments, international organizations, private companies and individuals are investing a great deal of time, efforts and budgets to install and use various security products and solutions. However, in spite of all these needs, activities, on-going efforts, and all current solutions, it is general belief that the security in today networks and applications is not adequate. We are daily witnessing various problems – infection of computers by malware, distribution of E–mail spam, phishing of Web pages, penetrations by hackers, software bugs, stolen industrial secrets and credit cards, disclosure of sensitive documents, and so on. All these interests and current problematic situations justify efforts and activities towards creating effective security solutions for network applications and environments. At the moment, there are two general approaches to network applications security:

- One approach is based on isolation, that is protecting them by isolating operational environments at their periphery using firewalls, port scanners, intrusion–detection tools, spam and phishing filters, “demilitarized zones”, Email spam filters, etc. and also using virus/malware scanners, virus signatures, encrypted disk files, etc.

- Another approach is called software security which is based on methodology to create secure, robust and protected applications, bug–free and not vulnerable to attacks, by using well–established methodology for design of applications, software tools for their development, and testing methodology and environments for their debugging and testing.

Although both approaches give some degree of security and protection, current situation in open networks indicates that in principle none of the current approaches is effective and does not produce secure, reliable and protected network applications and cloud environments. This means that current, mainly single point–solutions and approaches, reactive to emerging problems, have limited scope and effectiveness. Therefore, in the current situation, new approach and new thinking towards creating strongly and guaranteed secure network environments and applications are needed. This new approach is the focus and the concept of our research.

- AIM

To design a security system for network applications and cloud environments that will completely protect their resources, attributes, messages, software modules and execution environment against various attacks. Our methodology to create complex security systems and to apply total cryptographic protection is based on the design of security components in the form of generic security objects. Those objects can be data object, functional object or composite object. Each object is completely secure means its data attributes are protected, its functions can only be accessible to authenticated and authorized users, and its executable binaries are also protected. Furthermore, for our individual security objects and larger security systems, in order to prove their structural and functional correctness, we apply deductive scheme for verification and validation of security systems. So, verification of complex security systems is based on the principle that is: if individual objects are verified and proven to be secure, if their instantiation, combination and operations are secure, and if protocols between them are secure, then the complete system, created from such objects, is also verifiably secure. Our new security framework is based on our generic security objects, well established secure technologies and security standards. Therefore, the components designed based on this methodology will also be generic and compliant with security standards. The core components of our security system are Security Provider, Secure Execution Environment, and Security Protocols. They contain security engines of our security system, where each component provides the same set of tested security services. These components are complete with respect to their functionality, so developers can use these components to extend their applications with security features

- Literature Review and Related Work

Analyzing existing security solutions, products and architectures currently available for protection of resources and messages in network applications. Based on the analysis, we identify the shortcomings and problems with existing solutions.

○ *Security Providers (Libraries and Middleware)*

Security providers are usually crypto libraries or middleware modules exporting their Functionality through the set of Application Programming Interfaces (APIs). Generic Security Services Application Programming Interface (GSS-API) one of the first efforts to standardize cryptographic security platform was Generic Security Services Application Programming Interface (GSS-API) standard [1]. GSS-API itself does not provide security services; it is only a framework that offers security services to callers in a generic fashion, supported by a range of underlying mechanisms and technologies, such as Kerberos or public key cryptography.

Microsoft Security Support Provider Interface (SSPI) one of the implementations of GSS-API was by Microsoft in the form of Security Support Provider Interface (SSPI) [2]. SSPI is a set of interfaces between transport level applications and network security service providers and it is commonly known as Security Support Provider (SSP) or Cryptographic Service Provider (CSP) [3][4]. CSP is collection of providers: Microsoft Base Cryptographic Provider, Microsoft Enhanced Cryptographic Provider, Microsoft DSS Cryptographic Provider, Microsoft Base DSS and Difie-Hellman Cryptographic Provider, and Schannel Cryptographic Provider. Some of these providers are available with Windows 2000 and later versions and some enhanced are only available at selected locations due to export restriction policies. All these providers are proprietary solutions, so they cannot be used in open source projects and even for extensibility. Furthermore, Microsoft CSP is platform-dependent provider and is digitally signed only by Microsoft.

- *Security Management Protocols*

Client-server paradigm is widely used in distributed applications. Various security protocols for authentication, authorization and secure communications have been designed and developed. For authentication, the most popular protocol is Password Authentication Protocol. This protocol is based on a use of a secret password, which is known only by the end-user and the server. This protocol is considered weak protocol, because a password can be cracked by using various techniques, for example, guessing password, dictionary attack, and brute force attack. A comparatively secure protocol is Challenge Handshake Authentication Protocol. In this protocol, random numbers are exchanged between a client and a server for authentication. This protocol is also not secure; because it does not provide source authentication and replay attack can be used for impersonation. A modified version of this protocol uses asymmetric-key cryptographic functions. Secure Shell (SSH) and Strong Authentication protocol are examples of such protocols. SSH uses self-generated asymmetric keys, while Strong Authentication is based on X.509 certificates. Another authentication protocol is Extensible Authentication Protocol (EAP), described in RFC 3748 [5]. EAP is used for authentication of wireless LANs and most of operating systems support it. The extended version of EAP is Protected Extensible Authentication Protocol (PEAP) [6], which is based on TLS for certificates-based authentication and secure communications. Secure Socket Layer (SSL) protocol is widely deployed in most of commercial products for secure communications between clients and servers. SSL uses X.509 certificates and hash functions for confidentiality and integrity of messages. Most of companies integrated this protocol in their products with authentication and authorization protocols. Open SSL [7] is one of them. This library provides the set of cryptographic functions and security protocols. Open SSL is an open source implementation of SSL, which can be used with other applications for secure communications. Another product is eToken [8]. It provides USB smart-card-based strong user authentication and password management for enterprises. This solution is compliant with industry regulations and internal security policies. This solution also provides SSO services where a user can store more than one account information in a single token. Lexar® Jump Drive SAFE S3000 [9] is another products which provides hardware based encryption and smart card based authentication for multiple operating systems. In this solution, smart card securely generates and stores a cryptographic key which is eventually used to encrypt and decrypt user's stored data.

Authentication, encryption, decryption, and generates smart card-based digital signature for application data. Gemalto in collaboration with IBM also developed solution for web based Single-Sign-On protocol based on smart cards for physical and logical access control. This product supports public key cryptography and is fully compliant with FIPS- 201 and Europe Identification Authentication Signature standards. CryptoNET: Generic Security Framework for Cloud Computing Environments Smart Card Alliance [10] is an organization which provides platform to different member's organizations for smart card manufacturing, middleware development and smart card-based applications. The core objective of Smart Card Alliance is to promote smart card technologies for identification, payment and other user applications to ensure user privacy, data security and integrity.

- *Generic Security Server*

Currently available most important APIs and libraries for rapid development of application servers are available in the form of Eclipse plug-ins. They provide basic structure and functions like start server, strop server, publishing, targeting projects, adding and removing modules, etc. This is a generic framework, so new servers can also be added. It addressed scalability issues and provides solutions to handle multiple clients and requests. Therefore, it provides session management and client authentication. Each default message contains a header and a block of request specific data. The header includes packet size and a request type identifier, while data contains the actual information. In addition, the header may include security information, such as an authentication token, checksum, etc. IBM provided the concept of a generic server which is managed in the Web Sphere Application Servers administrative domain. WebSphere Application Server provides features to define a generic server as an instance of application server within the WebSphere Application Server administration domain. In current situation, the most important deployed generalized server is Web server [11]. It is a container for Web objects which are accessible to clients using HTTP protocol. Web server provides various services to Web modules. It provides transportation facilities, SSL based security features, basic access control services, and local and remote administration. Another application server is Lotus Domino [12]. It is a generalized server, but it expands services horizontally. Lotus Domino provides Web services, mail and newsgroup services. It supports Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP3), Internet Message Access Protocol (IMAP), Network News Transfer Protocol (NNTP), Lightweight Directory Access Protocol (LDAP) and Domino Internet Inter-ORB Protocol (DIIOP). This server provides access control using Access Control List (ACL). This mechanism determines whether or not users have access to the database and which levels of permissions are granted to users. This server provides transport level security services to all the components using SSL protocol. Furthermore, it provides Username and password-based authentication.

- *Secure Web System*

In this section, we analyzed some of the solutions which are being used for protection and authorization of Web content. Existing solutions are categorized as follows: Web Shields Attackers primarily target workstations for exploitation and spreading malicious code by devising various sophisticated techniques. These techniques [13] can be categorized in two groups: pull-based and push-based. The purpose of both techniques is to download and execute malicious code on workstations via E-mails or insecure Web contents. Drive-by-download attack is one of them. It uses pull-based techniques to download malware binaries [14]. It uses HTTP protocol as a carrier and Web mobile components for hiding and obfuscation purposes. Examples of Web mobile components are ActiveX [15], Java Applets [16], Flash scripts [17], plug-ins, etc. Some of these spyware exploit network connection of compromised workstation with the attacker to reveal weaknesses of the targeted for further exploitation. Most effective tools to combat against such types of attacks are Web shields, which are normally bundled with antivirus software. Some examples are Symantec Web Security Monitoring [18], Norton Internet Security [19], AVG [20], Avast [21], etc. These tools use virus/threat pattern and signature database to effectively detect the latest viruses and threats.

1) Intrusion Detection Systems

Various Intrusion Detection Systems (IDS) are developed to monitor network traffic and system activities. The purpose of such software is to detect malicious activities or policy violations in workstations or in a network, which are eventually reported to the administrator of a management station, but these systems do not prevent workstations from various Web threats. SNORT [22] is an example of such an IDS, which is an open source cross-platform lightweight network intrusion detection tool used for network traffic monitoring in order to detect suspicious network activities. It has rules-based logging to perform content pattern matching and to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, etc. Another IDS, nCircle [23], provides security risk and compliance management solutions. Reflex Security's Intrusion Prevention™ [24] solutions provide end-to-end enterprise network protection. Reflex IPS applies packet inspection with signature, anomaly and rate-based algorithms to inspect and control network traffic flows. This detection methodology already proved to produce either high rate of false positives or false negatives. Nessus™ [25] is another vulnerability scanner that provides a couple of good features, like efficient discovery of vulnerabilities, network configuration and auditing, asset profiling, etc. However, the major problem with Nessus is that it requires significant involvement of security administrators.

2) Protection and Authorization of Web Content

Most of Web sites provide SSL-based connections (HTTPS) in order to protect communication channels. SSL protocol uses certificate-based security solution for authentication and content protection. Normally, SSL-enabled Web sites dynamically download certificates into the client browser. However, in some cases, user may select a wrong option and browser overrides certificates verification, what increases the probability of man-in-the-middle attack over HTTPS. The same weakness of SSL is pointed out by T. Burg [26], which can be prevented by properly handling and verification of certificates. Along with Web contents protection, some Web servers implement access control mechanisms in order to restrict provision of Web contents to authorization users. Password-based access control and Access Control Lists are representative examples of such mechanisms.

○ *Secure Documents System*

The concept of storing data in an encrypted form was suggested in 1990s by Blaze92 [27] and designed as the Cryptographic File System (CFS) for UNIX operating system. Later, it was extended by Cattaneo97 [28], Zadok98 [29], and Hughes99 [30]. The system encrypts every file before storing it in a local repository or before sending it to remote servers. It decrypts the requested file before presenting it either to the client or to the intended server. Furthermore, the system provides security for communication channel between the sending host and networked file server. CFS transparently manages Crypto NET: Generic Security Framework for Cloud Computing Environments protection of documents based on symmetric keys, while access control is handled by applying key-level access control mechanisms. The author proposed a secure document distribution system based on Public-Key Cryptography [31] and secure socket layer.

1) Secure Socket Layer

(SSL) [32] For secure distribution of documents. Oracle Beehive is one of the products which provides unified single platform for all communication and collaboration services and documents sharing. This product uses user name /password-based authentication, access control lists for authorization, Secure Hypertext Transfer Protocol (HTTPS) [33], and SSL for secure communication. This product also provides audit capabilities, so that system administrator processes audit trail on log files generated by the system during day-to-day actions.

IBM Lotus Notes [34] is another popular tool for collaborative and team work. Similar to Oracle's Beehive, it is also a combination of different applications and document sharing is one of them. This product supports certification protocol, username/password authentication, Single-Sign-On, SSL with support of AES, and it is complaint with FIPS 140-2 [35] standard. It uses access control lists for access control services. This product uses .ID file to store user credentials which are protected by user's password. In addition, this product applies different levels of encryption according to the sensitivity of documents stored in a local workstation. Protection of documents using advanced cryptographic techniques was explained in [36]. That research addressed security issues of documents stored at a local station and shared in group environments. In addition, the solution structures documents in sections accessible only by authorized group members. The enforcement of authorization

Policies and protection of sections are achieved by using Role–Based Access Control and symmetric key cryptography. The system was implemented as an extension of Open Office using XACML [37], XACML Policy Server, and PEP. Another major example is Xerox DocuShare [38]. This solution implements security features like user name/password authentication, SSL to secure communication, different level of access policies to access contents, different roles to access encrypted contents, etc. Adobe [39] solution for trusted document sharing brings a new product for exchange of documents in government organizations. This product uses asymmetric key cryptography to encrypt and digitally sign documents and distribute them securely in a grouped environment.

- Proposed Work

The area of this research is security in distributed environment such as cloud computing and network applications. Specific focus was design and implementation of high assurance network environment, comprising various secure and security-enhanced applications. "High Assurance" means that our system is guaranteed to be secure, it is verifiable to provide the complete set of security services, we prove that it always functions correctly, and we justify our claim that it cannot be compromised without user neglect and/or consent.

The Principle is that if all resources of an application are always encrypted, i.e. "enveloped in a cryptographic shield", then Its software modules are not vulnerable to malware and viruses, Its data are not vulnerable to illegal reading and theft, All messages exchanged in a networking environment are strongly protected, and All other resources of an application are also strongly protected. Thus, we strongly protect applications and their resources before they are installed, after they are deployed, and also all the time during their use.

This research work will try to achieve some of these research objectives, to improve the security measures of the system,

- To create a Cloud Environment in a system making it as a server storing local files, database table, documents, applications etc. for testing the existing security modules and comparing the implemented proposed module.
- To study and investigate the existing generic security providers and examine it on our created cloud environment.
- To develop new proposed cryptographic security providers, examine it on our created cloud environment and comparing it with the existing generic security providers.
- To study and investigate the existing generic security management protocols and examines it on our created cloud environment.
- To develop new proposed cryptographic security management protocols, examine it on our created cloud environment and comparing it with the existing generic security protocols.
- To study and investigate the existing generic security servers and examine it on our created cloud environment.
- To develop new proposed cryptographic security servers, examine it on our created cloud environment and comparing it with the existing generic security servers.

- To study and investigate the existing execution environment and create it on our cloud environment.
- To create new proposed cryptographic execution environment and create it on our cloud environment and comparing it with the existing execution environment.
- To study and investigate the existing integrated workstation on our created cloud environment.
- To develop new proposed cryptographic workstation and comparing it with the existing environment on the cloud.
- To study and investigate the existing emails system on our created cloud environment.
- To develop new proposed cryptographic email system and comparing it with the existing email system on the cloud.
- To study and investigate the existing secure web system on our created cloud environment.
- To develop new proposed cryptographic web system and comparing it with the existing web system on the cloud.
- To study and investigate the existing secure document system on our created cloud environment.
- To develop new proposed cryptographic document system and comparing it with the existing document system on the cloud.

REFERENCES

1. J. Linn, “Generic Security Service Application Program Interface”, RFC-2743, RSA Laboratories, January 2000.
2. K. Brown, “Explore the security support provider interface using the SSPI workbench Utility”,at http://msdn.microsoft.com/msdnmag/issues/0800/ Security/Security0800.asp, 2000.
3. Denis Piliptchouk, “Java vs. .NET Security, Part 2 Cryptography and Communication”http://www.Onjava.com/pub/a/onjava/2003/12/10/javavsdotnet ml?page=1, [visited: October 2008].
4. Microsoft Inc.,“Microsoft CryptoAPI and Cryptographic Service Providers”, http://www.microsoft.com/technet/prodtechnol/windows2000serv/reskit/distrib/dscj_mcs _xxgl.mspx?m fr =true build date on11/19/2009.

5. B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, H. Levkowetz, Request for Comments: 3748, Extensible Authentication Protocol (EAP), June 2004.

6. Urien P, Badra M, Dandjinou M., "EAP-TLS smartcards, from dream to reality" published in the Proceedings of the Fourth Workshop on Applications and Services in Wireless Networks, ASWN 2004, Boston, MA, 2004.

7. http://www.openssl.org/docs/OpenSSL, http://www.openssl.org/docs/, [visited: January 2009].

8. White Paper, Aladdin Knowledge Systems Ltd., "Authentication Tokens: The Key toSecure PCs and Data", http://www.aladdin.com/, [visited: July 2010].

9. Lexar Media, Inc., Lexar® JumpDrive SAFE S3000, http://www.lexar.com/, [visited:July 2010].

10. Smart Card Alliance, http://www.smartcardalliance.org/, [visited: July 2010]. [43]. McAfee SECURE, "Data Sheet", downloaded form http://www.mcafee.com/us/local_content/datasheets/ds_endpoint_ encryption.pdf downloaded on September, 2009.

11. Jakarta Slide, http://jakarta.apache.org/slide/architecture.html, [visited: July 2009].

12. W. Tworek, G. Chiesa, F. Dahm, D. Hinkle, A. Mason, M. Milza, A. Smith, "Lotus Security Handbook", First Edition, April 2004.

13. Niels Provos, Dean McNamee, Panayiotis Mavrommatis, Ke Wang, and Nagendra Modadugu, "The Ghost In The Browser Analysis of Web-based Malware", published in the Proceedings of the 1st Workshop on Hot Topics in Understanding Botnets (HotBots '07), Cambridge, MA, pp. 4-13, April, 2007.

14. White paper, "Making Sense Of Man-In-The-Middle, Strategies For Mitigating A Menacing Threat", RSA, The Security Division of EMC, MITB WP 1009, January, 2010.

15. Roger A. Grimes, "Malicious Mobile Code, Virus Protection for Windows", ISBN: 1-56592-682-X, Chapter 11, August, 2001.

16. Vincenzo Ciaschini and Roberto Gorrieri, "Contrasting Malicious Applets by Modifying The Java Virtual MACHINE, Security and Protection in Information Processing Systems", Published in the proceedings of the 18th World Computer Congress, Toulouse, France, pp. 47-64, August, 2004.

17. Jianwei Zhuge, Thorsten Holz, Chengyu Song, Jinpeng Guo, Xinhui Han, and Wei Zou,"Studying Malicious Websites and the Underground Economy on the Chinese Web", Published by Springer in Managing Information Risk and the Economics of Security, pp. 225-244, December, 2007.

18. Symantec, White Paper, "Critical System Protection and Endpoint Encryption for thePCI Data Security Standard", [Online], http://www.symantec.com

/business/products/whitepapers.jsp?pcid=pcat _security&pvid=endpt_encryption_1#, [visited: February 2010].

19. Norton 360, “All in one Security”, [Online], http://www.symantec.com/norton /360, [visited: February 2010].

20. AVG Internet Security, [Online], http://www.avg.com/us-en/homepage, [visited: February 2010].

21. avast, [Online], http://www.avast.com/free-antivirus-download#tab2, [visited: February2010].

22. Snort, “The defacto standard for intrusion detection/prevention”, http://www.snort.org/,[visited: February 2010].

23. nCircle, “Proactive Network Security”, [Online], http://www.ncircle.com/ index.php, [visited: February 2010].

24. “Network Security Switch, Intrusion Prevention System and Policy”, REFLEX. http://www.reflexsecurity.com/, [visited: February 2010].

25. NESSUS. [Online],http://www.tenablesecurity.com/nessus/, [visited: February 2010].

26. Thomas Burg, “SSL Certificates and PKI in the NonStop World - and Other Worlds”, Published in The Connection, pp. 17-20, June, 2004.

27. [27]. M. Blaze, “A cryptographic file system for UNIX”, published in the proceedings of 1st ACM Conference on Communications and Computing Security, 1993.

28. G. Cattaneo, G. Persiano, A. Del Sorbo, A. Cozzolino, E. Mauriello and R. Pisapia, “Design and implementation of a transparent cryptographic file system for UNIX”, Technical Report, University of Salerno, 1997.

29. J. Hughes and D. Corcoran, “A universal access, smart-card-based, secure file system”, Atlanta Linux Showcase, October 1999.

30. E. Zadok, I. Badulescu and A. Shender. “Cryptfs: A stackable vnode level encryption file system”, Technical Report CUCS-021-98, 1998.

31. RSA laboratories, “What is public-key cryptography”, http://www.rsa.com/RSALABS/node.asp? id=2165, [visited: November 2009].

32. T. Dierks, E. Rescorla, RFC No. 5246, “The Transport Layer Security (TLS) Protocol”,SSL v3, August 2008.

33. E. Rescorla, A. Schiffman, RFC: 2660 “The Secure HyperText Transfer Protocol”, August 1999.

34. W. Tworek, G. Chiesa, F. Dahm, D. Hinkle, A. Mason, M. Milza, A. Smith, “Lotus Security Handbook”, First Edition, April 2004.

35. FIPS PUB 140-2, “Security Requirements for Cryptographic Modules”, Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899-8900, May 25, 2001.

36. M. Alhammouri, S. Muftic, “Management of Groups and Group Keys in Multi-level Security Environments”, 26th International Conference Computer Safety, Reliability, and Security (SAFECOMP 2007), Nuremberg, Germany, ISBN: 0302-9743, pp. 75-80, September 2007.

37. M. Alhammouri, S. Muftic, “A Model for Creating Multi-level-security Documents and Access Control Policies”, published in the proceeding of SSI´2006 - 8th Intl Symposium on System and Information Security, Sao Jose Dos Campos, Sao Paulo, Brazil, November, 2006.

38. White Paper: “Xerox DocuShare Security Features”, August 2009.

39. “Digital Signatures & Rights Management in the Acrobat Family of Products”, A guide for administrators and advanced users for Acrobat® Family of Products 9.x, Modification date: August 26, 2009.