Exploitation of Vulnerabilities in Cloud Storage

Narendran Calluru Rajasekar
School of Computing and Technology,
University of East London,
London, U.K.
crnarendran@gmail.com

Chris Imafidon
School of Computing and Technology,
University of East London,
London, U.K.
chris12@uel.ac.uk

Abstract - The paper presents the vulnerabilities of cloud storage and various possible attacks exploiting these vulnerabilities that relate to cloud security, which is one of the challenging features of cloud computing. The attacks are classified into three broad categories of which the social networking based attacks are the recent attacks which are evolving out of existing technologies such as P2P file sharing. The study is extended to available defence mechanisms and current research areas of cloud storage. Based on the study, simple cloud storage is implemented and the major aspects such as login mechanism, encryption techniques and key management techniques are evaluated against the presented attacks. The study proves that the cloud storage consumers are still dependent on the trust and contracts agreed with the service provider and there is no hard way of proven defense mechanisms against the attacks.

Keywords-cloud storage; security; architecture; vulnerabilities

I. INTRODUCTION

Computers has evolved from small computing devices such as abacus to super computers, and the computing has changed from stand alone computers to centralised computing and then to distributed computing. Current era is cloud computing era where all the software, platform and infrastructure are virtualised and provided as services typically exploited using pay-per-use model. In traditional model, a company or an organisation will maintain their own Information Technology Infrastructure and hence had full control on the data and processes. But in cloud computing, the data and processes are maintained by some 3rd party vendors and hence the control is lost.

Internet is ubiquitous and its penetration rate is very high in recent years. 82.5 percent of United Kingdom residents have access to Internet [28]. It is the communication channel for cloud computing and almost everyone using Internet is involved in some form of cloud computing activities such as Gmail, Yahoo, Picasa, Facebook and so on. Since it is open to everyone, if a soft spot is identified by attackers, it could be exploited to a great extent. Hence it is the tempting target for cyber crime.

Most of the companies like to move their applications to cloud services due to the huge cost savings it provides. But they are taken aback due to one main reason "Security". There are many leading cloud storage service providers such as Google Docs, Amazon Simple Storage Service (Amazon S3), Nirvanix, Adrive and Zumo drive. The vulnerabilities of cloud storage are very high, even the leading service providers have been compromised at some point.

In this paper, under the title "Attacks", various vulnerabilities in cloud storage are identified that could be exploited. "Implementation" discusses the solution implemented to counter the attacks based on the study. Finally, implementation is evaluated and conclusion is drawn.

II. ATTACKS

The cloud storage attacks can be classified into three broad categories network / resource based, browser based and social networking based attacks.

A. NETWORK / RESOURCE BASED

1) Denial of Service

It is the most popular attack in Network Security where the user is abstained from getting the normal service from the service provider with the help of other attacks such as Internet Control Message Protocol (ICMP) Flood, SYN Flood, User Datagram Protocol (UDP) Flood and Smurf attack. It can also be performed by increasing load on Central Processing Unit (CPU), primary memory, network to slow down or eventually crash the system. Distributed Denial of Service (DDoS) attack is based on DoS, which consists of three layers, controller layer, broker layer and attacker layer. In DDoS, the actual attack is made from the broker layer by receiving commands from the controller layer. Since it involves different layers attacker information can be hidden easily [29].

Since the attack can be performed using various other attacks, both detection and prevention steps can be taken. Lee et al. [7] has proposed a DoS/DDoS intrusion detection system, which uses cumulative sum algorithm to detect the attacks. Kompella et al. [22] proposed a novel data structure called partial completion filter, which can detect claim and hold attack, which is not handled in the former system.

2) Buffer Overflow

Buffer Overflow is the most common vulnerabilities for past two decades where an attacker seeks for partial or total control of a host. This is caused when exceptions are not handled properly. For example: out of bound exceptions and type exceptions when not handle properly can be exploited to move the control to a function introduced by an attacker and the results are endless [3].

The buffer overflow vulnerabilities can be avoided by taking little extra care during development of software.

 ISBN: 978-1-61208-106-9

Further down, buffer overflow can be prevented even at the Kernel level [30] and Hardware level [32] as well.

3) Virtual Machine Based Rootkit

Virtualisation is an important aspect of cloud computing where the software, operating system and all related components are packaged together such that it is independent of the hardware [15]. This is facilitated by multiplexing the system with a small privileged kernel known as hypervisor. Virtual Machine Based Rootkit (VMBR) is a new type of malware, which is similar to hypervisor, installs underneath the operating system layer and hoist the operating system to virtual machine. Hence, it is difficult to detect VMBR's state by the software running on the operation system. Vitrol and Subvert are other rootkits that use this technique [25]. VMBR allows other malicious software or services to run on it, which is protected by the operating system. According to King et al. [25], the best way to detect VMBR is to control the layer beneath it with the help of a secure hardware or bootable media.

4) Side Channel

Although complex cryptographic algorithms are devised for security, the weakness in implementation is exploited to break the security. The side channel attack exploits the unintended data leakage such as power consumption, timing information to break the keys [19].

Lee et al. [11] proposed Lock and Key technique, which includes a test security controller that randomises the sub chains when accessed by an unauthorised user and hence reducing the predictability. An attacker has to break many layers of security in order to the access the scan chain in order to exploit it.

5) Man in the Middle

There are different variants of man in the middle attack exists. One of the variants is where an attacker having a device with two wireless cards can launch this attack. First he sends de-authentication frames of legitimate user to the service station. Legitimate user will be disconnected from the service station and start searching for access point in the same channel. Now attacker can use one of his wireless cards to act as service station and connect the legitimate user, mean while use the other wireless card to get connected to the actual service station using legitimate user Media Access Control (MAC) address [23]. Thus man in the middle attack can be launched successfully.

Once the attack is successfully launched, attacker can even attack Hyper Text Transfer Protocol Secure (HTTPS) communication. In this scenario, the user will be displayed a security certificate warning, which most of the users ignore and hence the system is compromised [6].

6) Replay Attack

Replay attack is where an attacker is able to capture the network traffic and replay it at a later time to gain access to the unauthorised resources even if it is encrypted. It has more effect on the Dynamic Rights Management (DRM) content where the rights over the resource changes over the time or number of times/ bandwidth used. If the attack is on the DRM content, the user not only looser privacy but also cost involved in the dynamic rights. For example, if user is accessing a file from the cloud storage service such as Amazon S3 where he is charged based on the amount of data transferred, the replay attack will eat up user's money. Abbadi et al. [9] have proposed a solution against the replay attack, which give users flexibility to use and manage the DRM content in any of the devices they own.

7) Resource Exhaustion

"Resource-exhaustion vulnerability is a specific type of fault that causes the consumption or allocation of some resource in an undefined or unnecessary way, or the failure to release it when no longer needed, eventually causing its depletion. [10]"

As stated in the definition, resource exhaustion vulnerability can be exploited to cause Denial of Service (DoS) attacks. This can be caused by bad design or inefficient utilization of resources on the service side and resource leakage where the resources are not released or destroyed from memory after use [10]. Hence it is difficult to observe and identify the cause unless it is closely monitored.

Antunes et al. [10] has proposed methodology to detect resource exhaustion vulnerability using which implemented Predator, a black box testing tool to identify the resource exhaustion vulnerabilities of a system. The operations of the tool involve attack generation and injection campaigns. Incorporating this methodology in software development life cycle (SDLC) can reduce this vulnerability.

8) Byzantine Failure

In cloud storage, many nodes participate to complete an activity. For instance there could be many redundant servers involved and also multiple users accessing single source. In this scenario, any of the nodes i.e., servers or users participating can fail arbitrarily as a result of crash or malicious activity, which is known as Byzantine failure [13]. System can be made robust by implementing threshold cryptography [2] to ensure the system is tolerant to Byzantine failure.

Recently, Wang et al. [24] proposed Agreement Protocol for cloud computing (APCC), which involves two processes Interactive Consistency Process and the Agreement Process. The Interactive Consistency Process is executed at the server nodes, which shares and stores the initial message among them. The server nodes then aggregate the results and transmit the message to client nodes. The Agreement Process is executed at the client nodes to receive the agreed message.

B. BROWSER BASED

1) XSS

Cross Site Scripting (XSS) can be used to inject malicious code into the client machine by exploiting the client side script vulnerability in the website [1]. Thus an attacker can introduce his own script and impersonate user credentials to perform malicious activity in the website such as session Hi-jacking and also craft phishing sites. A user called Samy added more than 1 million buddies to his My Space account by exploiting XSS Vulnerability [5]. Even Google has suffered from XSS vulnerability in their online spreadsheet application using which the user cookie can be stolen, which is valid for other sub-domains also. From the server point of view detecting and preventing XSS attack is a

 ISBN: 978-1-61208-106-9

difficult task as it is done at the client end for which server has not much control. Wurzinger et al. [20] has proposed a server side solution SWAP (Secure Web Application Proxy) to detect and prevent XSS. It has a reverse proxy, which intercepts the HTML response from client and validates it for XSS attack.

2) SQL Injection

Similar to XSS, SQL Injection is also a vulnerability, which can be used to inject malicious database scripts when user inputs are not properly validated. This can generally be prevented by passing user inputs as parameters and avoiding query building based on the user input. At times there will be scenarios where building queries based on user inputs are unavoidable. In these cases, vulnerable user input validation must be performed to prevent SQL Injection. SQL Injection is very dangerous as it can be used to change values of multiple records and can even be used to delete the whole table [31].

For example, consider the following SQL statement, which updates a value based on email id.

UPDATE [User Information] SET [Credit] = '£1000000' WHERE [Email Id] = '$email'

If user passes value ['xyz@abc.com' OR 'x'='x] for $email. Then 'x'='x' condition is always true and hence credit will be set to '£1000000' for all users.

3) Malware

A program designed to damage the machine is called malware [14]. The web browsers are more susceptible to malwares as it supports extension of 3rd party programs through "Add-on" or "Plug-in" capability. Louw et al. [17] has demonstrated a malware program, which is capable of capturing sensitive information such as passwords even when the communication is done using Secure Socket Layer (SSL). This is possible because the information is captured even before the communication begins when the data is encrypted for submission. Some of the Internet security program may detect and warn some of the malicious activities of malware programs such as submitting hidden data to remote server, but not many users are not aware of the technical details and tend to ignore the warning.

4) XML Wrapping

Web Services is a key technology to implement Service Oriented Architecture (SOA) especially is very useful for implementing interoperable and platform independent services. Extensible Markup Language (XML) is the underlying mark up language used to communicate between server and client. XML signatures facilitate the unauthorised modification and origin authentication for the XML documents [16].

A SOAP message with a signed body can be moved to different wrapper message without altering the signatures. Hence the resulting SOAP message is still valid producing valid hash [18]. This is called XML Wrapping attack, which according to Gruschka et al. [18] can be mitigated using SOAP message security validation and XML schema validation but the formal proof of safety is missing.

C. SOCIAL NETWORK

1) Sybil Attack

In a Sybil attack [8], a malicious user acquires multiple identities and pretends to be distinct users and tries to create a relationship with honest users. Even if one honest user is compromised, malicious user will gain special privileges, which can be used for attacks. Cloud storage is widely used in social networking such as Facebook, My Space, Orkut, Bebo where users can store their files such as documents, photos and videos and share it easily with their network. The relationship between the honest user and the malicious user is called attack edge, which can even be used for social engineering.

Vanish [21] is a proposed system, which increases privacy by self destructing data. Using this system, a message can be encrypted using a random key, which is stored in the distributed hash table (DHT). These keys will be destructed from DHT after user specified interval and hence the data is lost forever. User can de-encrypt the data using the key before it is destructed forever. This seems to be a solution for P2P based storages and also has been simulated for Gmail using Firefox Plug-in but in its current form it is not adequately protected against Sybil attacks and can be defeated [26].

2) Social Intersection Attacks

Social Intersection attacks can be effectively launched in social networking environment. It can be used to identify the original owner of the shared anonymous data object with just two compromised users in a group [12]. It is hard to detect this kind of attack as it is performed passively and become more powerful with increase in number of compromised users. A solution is proposed for this attack where the service provider can build a number of anonymous nodes around a user and hence highly reducing the probability of identifying the originating source.

3) Collusion Attack

Secret Key Multiplication (SKM) group re-keying scheme is used in group collaboration, where multiple users participate in a group discussion that might involve sharing of various resources such as text, files and even hardware resources. According to this scheme a subset key is generated for each group from a master key and in turn each user in a group is given a private key. It is assumed that the users in each group keep their key secret. Collusion attack is performed by combining information among two or more users and gain access to resources that the attacker is not supposed to have. Using collusion attack, it is possible to obtain even the high level key, which will give attacker access to other groups, which is not related to the attacker. Raphael [4] proved SKM group re-keying scheme to be vulnerable to collusion attacks.

III. IMPLEMENTATION

Tb drive [27], an online storage is implemented using the architecture devised based on the study. As a student it is hard to avail access to storage servers used for commercial purpose. Hence storage provided by a web hosting service was utilised to implement this project, which had limitations on server capacity and performance.

 ISBN: 978-1-61208-106-9

The application is developed using ASP.Net 3.5 with C# as server side script language, AJAX and Visual Studio 2010 IDE. MS SQL 2008 is used for storing data. IIS 7.0 is used as the web server to handle client requests and the developed application is tested using different browsers namely Fire Fox, Internet Explorer, Google Chrome and Safari under Mac OS X and Windows platforms.

This web application is designed to meet the functionality and security requirements, based on the analysis done in previous chapters. ASP.Net application service is used to implement the access controls for the web application.

Folders are stored in database using Hierarchyid, new data type available in SQL Server 2008. This facilitated displaying folder structure in Tree View control reducing number of Lines of Code (LOC).

The implementation of the application is discussed below in four major aspects namely login mechanism, storage organisation, file encryption and decryption and key management.

A. Login Mechanism

Tb drive doesn't store password to authenticate users, instead implements Open ID mechanism to authenticate user. Implementation accepts Google and Yahoo Open Ids for login to Tb drive and demonstrates simplest account creation process. Users can start using the application readily and securely without having to fill out sign up forms. User simply has to click on the Open ID provider logo. Application will be redirected to Open ID service provider. Open ID service provider will ask for user name and password for authentication mentioning the requesting domain (techbizarre.com) name in the authentication page.

Once the user enters valid user name and password, Open ID service provider will display all the details requested by the relying party (techbizarre.com) requesting approval from the user. If the user approves the details to be shared, the requested information will be sent to relying party. User can also choose to remember the association; hence this step can be skipped in future.

Open ID provides only authentication and don't support user session that has to be taken care by the relying party. Single-Sign-On generally known as SSO is another mechanism, which facilitates even the session to be taken care by the service provider and have its own advantages and disadvantages. Though Open ID doesn't facilitate session management, the service is provided free of charge where as cost is involved to avail SSO facility form 3rd party. If Open Id is implemented, application can accept services from many service providers and hence more audience where as when SSO is implemented, the service is restricted to one service provider.

B. Storage Organisation

In cloud storage, files can be stored in two different ways file system and database. Both has advantages and disadvantages; the file system requires full permission on the disk, which is difficult in a web hosting scenario than maintaining own server. The storage system requires a binding between the application layer and storage layer, which is loosely coupled when the files are stored in the file system and there is no concrete relationship between the files and the application layer where privacy and confidentiality is driven. Figure 1 depicts typical high level cloud storage architecture.

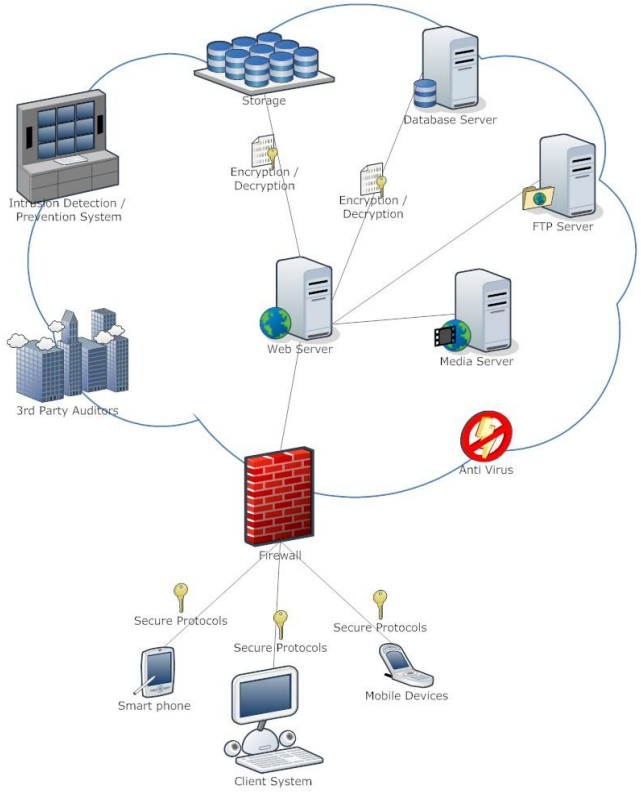

Figure 1. High Level Cloud Storage Architecture.

The binding between the application layer and storage layer can be made strong when the storage is implemented in database server. This may hinder other support such as File Transfer Protocol (FTP) services for the storage facility.

Considering the web hosting storage scenario of Tb drive, database is chosen for storage layer but storage using File system is also demonstrated without the feature of encryption.

Hierarchyid data type available in SQL Server 2008 is used to implement the folder structure in Tb drive, which allows easy mapping of folder structure to tree view control.

C. File Encryption & Decryption

Since the files are stored in 3rd party server, this might break user's privacy. Hence the ability to encrypt files can be provided to users. In security perspective, submitting encrypted files seems more appropriate; but, when considering key management, there exists risk of user using various keys to encrypt files and may get lost when retrieving data. To make it more user friendly, Tb drive uses master key concept which is explained under Key Management. Tb drive accepts a key string from user, which is internally converted to key and vector combination to

 ISBN: 978-1-61208-106-9

encrypt files, Tb drive demonstrates symmetric encryption of files using Rijndael encryption algorithm. Individual files can be encrypted using a key. Since symmetric encryption is used, user has to provide the same key to decrypt the files. If the key is lost, the data is lost forever.

D. Key Management

Since loss of key can lead to data lost forever, key management is crucial in cloud storage. To mitigate this risk, Tb drive stores one way hashed master key supplied by user in the database and uses the master key to encrypt files. Since Tb drive only stores hashed master key, unless user provides the master key data cannot be decrypted. Thus users can safely store data in Tb drive having full access with them. The other concern here is that the law-maker can be law-breaker, i.e., the key management technique logic is again provided by the 3rd party service providers who can internally change the logic. Hence the service provider should conduct regular external audit that can verify and certify the credibility of the application and thus it can be trusted.

IV. CONCLUSION AND FUTURE WORK

People are addicted to simplicity and are lazy to manage multiple passwords. One of the solutions to overcome this could be Open ID, which increases usability but at the same time increases the risks of XSS and phishing attacks. Hence general awareness is required on Do's and Don'ts of cloud computing.

ASP.Net Membership service simplified the implementation of session security and assumed to be safe from attacks. If any of the vulnerability in the Membership assembly is exploited then the assumption is flawed.

Simple flaws in coding such as not destroying objects in memory that is no longer required can be easily exploited to launch DoS attack and buffer over flow attack, which decrease the performance of the system or even crash the server. Coding flaws such as lack of input validations can lead to SQL injection through which an attacker can gain access full database.

Service provider has full access and in most of the cases they have ability to impersonate a customer and have full access over his data. An attack by a disgruntle employee of the service provider can easily break in to consumers data. Encryption can protect sensitive data but still vulnerable to Man in the Middle attack.

Attacks through social networks are recent ones, which are being exploited as this doesn't require any extra burden for an individual to initiate. Most of the social network users are naive enough to give away sensitive and personal information in social network websites, which can then be used to break the password using password recovery facility that every service provider provides.

A famous phrase exists "Words spoken cannot be taken back"; similarly Data given away to cloud cannot be taken back. No one knows how many backup exists for the data stored in cloud. A user can upload unencrypted data assuming to encrypt it after uploading into the cloud. Even before the data is encrypting, it could have been backed up and user is left unaware.

Cloud service relies on the network, which is not always secure. Network sniffers can easily gain sensitive information by monitoring network payloads.

Security should be implemented at each and every layers defined in International Standards Organisation – Open System Interconnect (ISO-OSI) Model and at very granular level. The attacks such as denial of service attack, buffer overflow attack, man in the middle attack exists for ages, still there is no concrete mechanisms to counter these attacks. Even the strongest encryption available is vulnerable to side channel attacks.

Even the leading service such as Google, Zoho, Nirvanix has failed at some point exposing customer data and even losing the data. Hence users should have their own backup mechanisms for critical data.

ACKNOWLEDGEMENT

The success of this work largely depends on the motivation and encouragement provided by all my tutors in the University of East London. I also extend my thanks to all my friends who have been supportive achieving this work and special thanks to Arul Arasu Ganesan, Brindha Sheshadri and Madhuvanthi Dayalan who reviewed this work.

I also owe a big thanks to University of Cambridge for the security seminar held at William Gates building. This helped me to learn aspects of security oriented languages.

I owe a special thanks to MSDN Academic Alliance, which made possible for me to gain access to recent version of Microsoft Window Server 2008, Visual Studio 2010 and Microsoft SQL Server 2008.

REFERENCES

[1] A. Kieyzun, P. J. Guo, K. Jayaraman, and M. D. Ernst. "Automatic creation of SQL injection and cross-site scripting attacks", Proceedings of the 2009 IEEE 31st International Conference on Software Engineering, pp. 199-209, 2009.

[2] C. Cachin and S. Tessaro. "Optimal resilience for erasure-coded Byzantine distributed storage." Distributed Computing, pp.497-498, 2005.

[3] C. Cowan, P. Wagle, C. Pu, S. Beattie, and J. Walpole. "Buffer Overflows: Attacks and Defenses for the Vulnerability of the Decade," oasis, pp.227, Foundations of Intrusion Tolerant Systems (OASIS'03), 2003.

[4] C. Raphael. "Collusion attacks on secret keys multiplication (skm) group re-keying scheme proposed at CITA03.", unpublished.

[5] E. Levy and I. Arce. "New threats and attacks on the world wide web." IEEE Security & Privacy, pp. 234-266, 2006.

[6] F. Callegati, W. Cerroni, and M. Ramilli. "Man-in-the-Middle Attack to the HTTPS Protocol," IEEE Security and Privacy, vol. 7, no. 1, pp. 78-81, Jan./Feb. 2009, doi:10.1109/MSP.2009.12

[7] F. Leu and Z. Li. "Detecting DoS and DDoS Attacks by Using an Intrusion Detection and Remote Prevention System", IEEE Conference and Exposition, pp. 1-15, 2009.

[8] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. "SybilGuard: Defending Against Sybil Attacks via Social Networks." IEEE/ACM TRANSACTIONS ON NETWORKING, vol. 16, no. 3, pp. 576-589, 2008.

[9] I. M. Abbadi and M. Alawneh. "Replay Attack of Dynamic Rights within an Authorised Domain," securware, pp.148-154, 2009 Third International Conference on Emerging Security Information, Systems and Technologies, 2009.

[10] J. Antunes, N. F. Neves, and P. J. Ver. "Detection and Prediction of Resource-Exhaustion Vulnerabilities" issre, pp.87-96, 2008 19th International Symposium on Software Reliability Engineering, 2008.

[11] J. Lee, M. Tehranipoor, C. Patel, and J. Plusquellic. "Securing Designs against Scan-Based Side-Channel Attacks." IEEE transactions on dependable and secure computing, vol. 4, no. 4, pp. 325-336, 2007.

[12] K. Puttaswamy, A. Sala, and B. Y. Zhao. "StarClique: guaranteeing user privacy in social networks against intersection attacks", Proceedings of the 5th international conference on Emerging networking experiments and technologies, pp. 157-168, 2009.

[13] K., Driscoll, B. Hall, H. Sivencrona, and P. Zumsteg. "Byzantine fault tolerance, from theory to reality." Computer Safety, Reliability, and Security, vol. 2788, pp. 235-248, 2003, doi: 10.1007/b12002.

[14] M. D. Preda, M. Christodorescu, S. Jha, and S. Debray. "A semantics-based approach to malware detection.", ACM Transactions on Programming Languages and Systems (TOPLAS), vo. 30, no. 5, pp. 25, 2008.

[15] M. F. Mergen, V. Uhlig, O. Krieger, and J. Xenidis. "Virtualization for high-performance computing.", ACM SIGOPS Operating Systems Review, vol. 40, no. 2, pp. 11, 2006.

[16] M. McIntosh and P. Austel. "XML signature element wrapping attacks and countermeasures", Proceedings of the 2005 workshop on Secure web services, pp. 20-27, 2005.

[17] M. T. Louw, J. S. Lim, and V. N. Venkatakrishnan. "Enhancing web browser security against malware extensions." Journal in Computer Virology, vol. 4, no. 3, pp. 179-195, 2008.

[18] N. Gruschka and L. Iacono. "Vulnerable Cloud: SOAP Message Security Validation Revisited", IEEE International Conference on Web Services, pp. 625-631, 2009.

[19] N. R. Potlapally, A. Raghunathan, S. Ravi, N. K. Jha, and R. B. Lee. "Aiding side-channel attacks on cryptographic software with satisfiability-based analysis.", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 15, no. 4, pp. 465-470, 2007.

[20] P. Wurzinger, C. Platzer, C. Ludl, E. Kirda, and C. Kruegel. "SWAP: Mitigating XSS attacks using a reverse proxy", ICSE Workshop on Software Engineering for Secure Systems, pp. 33-39, 2009.

[21] R. Geambasu, T. Kohno, A. Levy, and H. M. Levy. "Vanish: Increasing data privacy with self-destructing data" Unisex Security Symposium, vol. 18 pp. 299-350, 2009.

[22] R. Kompella, S. Singh, and G Varghese. "On Scalable Attack Detection in the Network." IEEE/ACM TRANSACTIONS ON NETWORKING vol. 15, no. 1, 2007.

[23] R. Syahputri and M. Hasibuan. "Security in Wireless LAN Attacks and Countermeasures", SNATI, pp.54-78, 2009.

[24] S. C. Wang, K. Q. Yan, S. S. Wang, and C. P. Huang. "Achieving high efficient agreement with malicious faulty nodes on a cloud computing environment", Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, pp. 468-473, 2009.

[25] S. King and P. Chen. "SubVirt: Implementing malware with virtual machines", IEEE Symposium on Security, pp. 1-14, 2006.

[26] S. Wolchok, O. S. Hofmann, N. Heninger , E. W. Felten , J. A. Halderman , C. J. Rossbach, et al. "Defeating Vanish with Low-Cost Sybil Attacks Against Large DHTs", Citeseer, 2010.

[27] Tb Drive. http://www.techbizarre.com/tbdrive/ seen: 29.08.2010

[28] United Kingdom Internet Usage Stats and Market Report. http://www.internetworldstats.com/eu/uk.htm seen: 25.08.2010

[29] W. Liu. "Research on DoS Attack and Detection Programming", IITA, pp. 207-210, 2009.

[30] W. Speirs. "Making the kernel responsible: a new approach to detecting & preventing buffer overflows", Proceedings of the Third IEEE International Workshop on Information Assurance, pp. 21-32, 2005.

[31] X, Fu and K. Qian. "SAFELI: SQL injection scanner using symbolic execution", Proceedings of the 2008 workshop on Testing, analysis, and verification of web services and applications, pp. 34-39, 2008.

[32] Z. Chen and X. Yan. "Hardware Solution for Detection and Prevention of Buffer Overflow Attacks in CPU Micro-architecture." RESEARCH AND PROGRESS OF SSE, vol. 26, no. 2 pp. 214-219, 2006.

 ISBN: 978-1-61208-106-9