International Journal of Future Innovative Science and Engineering Research (IJFISER)
Volume - 2, Issue - II ISSN (Online): 2454- 1966

Research Manuscript Title

DETECTION AND PREVENTION OF VAMPIRE ATTACK AND SESSION HIJACKING IN WIRELESS NETWORK

[1] M.Shanmathi B.Tech.,(M.E.), [2] Mrs.X.Anitha saraffin M.E.,

P.G. Scholar, Associate.professor,
Dept of CSE MNM Jain Engineering College Chennai.

E-Mail: shanmathi.it@gmail.com, readyanita@gmail.com

JUNE – 2016

www.istpublications.com

DETECTION AND PREVENTION OF VAMPIRE ATTACK AND SESSION HIJACKING IN WIRELESS NETWORK

[1] M.Shanmathi B.Tech.,(M.E.), [2] Mrs.X.Anitha saraffin M.E.,

P.G. Scholar, Associate.professor,
Dept of CSE MNM Jain Engineering College Chennai.

E-Mail: *shanmathi.it@gmail.com, readyanita@gmail.com*

ABSTRACT

As attack techniques evolve, cyber systems must also evolve to provide the best continuous defense. This paper proposes a scheme to detect and prevent resource depletion attacks, called Vampire Attacks at the routing protocol layer, which permanently disable networks by quickly draining nodes battery power. The Session Hijacking attack compromises the session token by stealing or predicting a valid session token to gain unauthorized access to the Web Server. This extensive experimental analysis shows that OTC introduces a latency of less than 6 ms when compared to cookies - a negligible overhead for most web applications. Moreover, this paper shows that OTC can be combined with HTTPS to effectively add another layer of security to web applications. In so doing, we demonstrate that One-Time Cookies can significantly improve the security of web applications with minimal impact on performance and scalability.

Index terms *– Resource depletion attack, Session Hijacking attack, One-Time Cookies, Session Token, Magic Cookie, Latency, Scalability.*

1. INTRODUCTION

Wireless networks are computer networks that are not connected by cables of any kind. The use of a wireless network enables enterprises to avoid the costly process of introducing cables into buildings or as a connection between different equipment locations. The basis of wireless systems is radio waves, an implementation that takes place at the physical level of network structure. Ad-hoc low-power wireless networks are a high price research direction in sensing and pervasive computing.
Wireless network use radio waves to connect devices such as laptops to the Internet, the business network and applications. When laptops are connected to Wi-Fi hot spots in public places, the connection is established to that business's wireless network. There are four main types of wireless networks:

- Wireless Local Area Network(LAN)
- Wireless Metropolitan Area Network(MAN)
- Wireless Wide Area Network(WAN)
- Wireless Personal Area Network(PAN)

In a wireless network, each node observes physical phenomena in its sensing range. Node processing quantizes and combines, or fuses, the observations to produce aggregate information, processing occurs along an intermediate sequence of wirelessly linked nodes that ultimately reaches the sink (destination) node. Wireless networks (WN) are networks of small resource constrained devices which sense the environment and report the results via wireless networks. They allow spatial or temporal measurements of phenomenon previously difficult to analyze.

One of the current challenges in the WN field is the development of management systems which allow WN to be easily deployed in various application domains as different WN application domains often have different management requirements. However given that WSN are very restricted in terms of resources and usually battery powered, overheads involving communication are to be avoided as much as possible. Such a network is highly vulnerable to Vampire Attack, which are not protocol specific and which does not use the loop holes in the routing protocols.

The Session Hijacking attack consists of the exploitation of the web session control mechanism, which is normally managed for a session token. Because http communication uses many different TCP connections, the web server needs a

method to recognize every user's connections. The most useful method depends on a token that the Web Server sends to the client browser after a successful client authentication. A session token is normally composed of a string of variable width and it could be used in different ways, like in the URL, in the header of the http requisition as a cookie, in other parts of the header of the http request, or yet in the body of the http requisition.

The Session Hijacking attack compromises the session token by stealing or predicting a valid session token to gain unauthorized access to the Web Server. The session token could be compromised in different ways; the most common are:

- Predictable session token;
- Session Sniffing;
- Client-side attacks (XSS, malicious JavaScript Codes, Trojans, etc);
- Man-in-the-middle attack
- Man-in-the-browser attack

HTTP is a stateless protocol. Requests to a web server are treated as independent transactions with no relation to each other. While simple and scalable, this design is not adequate for web applications that require sessions - the association of multiple transactions to a single user (e.g., online banking and e-commerce applications). HTTP cookies, small pieces of data that keep session state information in the browser, were designed to address this limitation and rapidly became the dominant mechanism for HTTP session management. Although cookies are a practical and efficient mechanism for session management, they introduce a number of security risks, especially when employed as session authentication tokens - a function for which they were not specifically designed.

For example, most web applications rely on the security provided by HTTPS to protect the user's password during the login process. During this step, the web application generates cookies that the user can later employ as lightweight session authentication tokens. However, due to performance concerns, many web applications switch to HTTP after the user logs in and cookies are transmitted "in the clear". As a result, cookies are exposed to any adversary eavesdropping on the communication. Because cookies are static, an adversary can use them to gain unauthorized access to the user's session. While these session hijacking or "side jacking" attacks are not new, a significant number of web applications are still vulnerable.

Several factors such as the proliferation of open wireless networks and the release of automated attack tools have increased the risk of this threat. The most recommended defense is to use HTTPS to protect all communications with the web application ("always on HTTPS"). However, deploying always-on HTTPS can be challenging due to performance and financial concerns, particularly for distributed systems. More importantly, always-on HTTPS is not a complete solution; cookies can still be exposed due to configuration errors or by attacks against HTTPS and the browser. In short, always-on HTTPS does not address the root cause of the problem: cookies are weak session authenticators.

More robust alternatives to authentication cookies have been proposed . However, they have not been adopted due to their additional requirements and complexity. Specifically, most of these alternatives require state in the web server. This is a problem for highly distributed web applications because this state needs to be synchronized among servers in different geographic locations. Thus, the effect of network latency will not only make synchronization operations more expensive, but will also cause valid requests to be denied due to "out-of-sync" state. Web 2.0 applications are particularly affected by this problem due to their higher request concurrency. In short, proposed alternatives to authentication cookies fail to address the operational requirements of highly distributed Web 2.0 applications and, as result, have not been deployed.

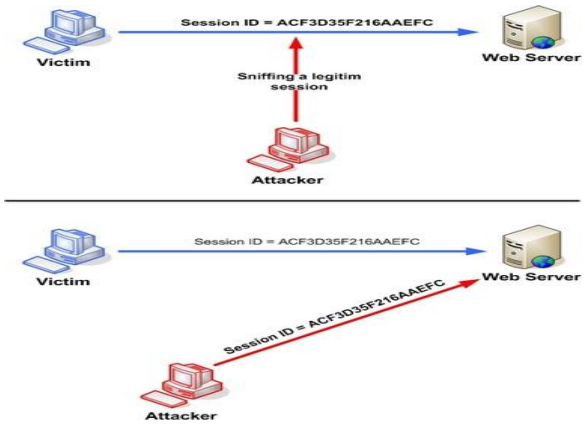

Figure 1 Manipulating the token session executing the session hijacking attack.

In this paper, we present One-Time Cookies (OTC), a more secure alternative to authentication cookies that does not require state in the web application. Instead of using a single, static token to authenticate each request, OTC generates a unique token per request based on a session key. Each OTC token is tied to a particular request by using a Hash-based Message Authentication Code (HMAC); hence, an adversary cannot reuse OTC tokens to illicitly redirect a session. To avoid state in the web application, OTC borrows the concept of Kerberos service tickets.

Like in Kerberos, an OTC session ticket contains the information the web application needs to verify an OTC token (i.e., session key), encrypted with a master key only known by the web application. Thus, any web application's server can verify OTC tokens without keeping any volatile data, one of the main barriers for deploying alternatives to cookies in highly distributed systems. Unlike cookies, OTC credentials are also securely stored and isolated from other browser components. We evaluate our proposed mechanism and demonstrate an overhead similar to the insecure traditional cookie approach.

We strongly believe that OTC raises the bar against real threats, but are careful not to over claim the guarantees that OTC can provide. Specifically, while our approach efficiently eliminates session hijacking attacks by ensuring session integrity (i.e., the integrity of navigation requests), it does not provide confidentiality or full integrity protection for the information exchanged between the browser and the web application. If these additional security guarantees are required, OTC can be used together with always-on HTTPS; OTC and HTTPS are complementary security mechanisms. OTC's main goal is to replace cookies as session authenticators, with a performance-conscious solution that can be deployed across traditional and highly distributed web applications.

2. RELATED METHODS

[1] A low-rate distributed denial of service (DDoS) attack has significant ability of concealing its traffic because it is very much like normal traffic. It has the capacity to elude the cur- rent anomaly-based detection schemes. Information metric can quantify the differences of network traffic with various probability distributions. In this paper, we innovatively propose using two new information metrics such as the generalized entropy metric and the information distance metric to detect low-rate DDoS attacks by measuring the difference between legitimate traffic and attack traffic. The proposed generalized entropy metric can detect attacks several hops earlier (three hops earlier while the order) than the traditional Shannon metric. The proposed information distance metric outperforms (six hops earlier while the order) the popular Kullback–Leibler divergence approach as it can clearly enlarge the adjudication distance and then obtain the optimal detection sensitivity. The experimental results show that the proposed information metrics can effectively detect low-rate DDoS attacks and clearly reduce the false positive rate. Furthermore, the proposed IP trace back algorithm can find all attacks as well as attackers from their own local area networks (LANs) and discard attack traffic.

[2] A new distributed approach to detecting DDoS (distributed denial of services) is the flooding attacks at the traffic flow level. The new defense system is suitable for efficient implementation over the core networks operated by Internet service providers (ISP). At the early stage of a DDoS attack, some traffic fluctuations are detectable at Internet routers or at gateways of edge networks. We develop a distributed change-point detection (DCD) architecture using change aggregation trees (CAT). The idea is to detect abrupt traffic changes across multiple network domains at the earliest time. Early detection of DDoS attacks minimizes the flooding damages to the victim systems serviced by the provider each ISP domain has a CAT server to aggregate the flooding alerts reported by the routers. CAT domain servers collaborate among themselves to make the final decision. To resolve policy conflicts at different ISP domains, a new secure infrastructure protocol (SIP) is developed to establish the mutual trust or consensus. Experimental results show that 4 network domains are sufficient to yield a 98% detection accuracy with only 1% false-positive alarms. Based on a 2006 Internet report on AS (autonomous system) domain distribution, we prove that this DDoS defense system can scale well to cover 84 AS domains. This security coverage is wide enough to safeguard most ISP core networks from real-life DDoS flooding attack.

[3] The growing popularity and application of Web services have led to increased attention regarding the vulnerability of software based on these services. Vulnerability testing examines the trustworthiness and reduces the security risks of software systems. A worst-input mutation approach for testing Web service vulnerability based on Simple Object Access Protocol (SOAP) message is proposed. Based on characteristics of SOAP messages, the proposed approach uses the farthest neighbor concept to guide generation of the test suite. The corresponding automatic test case generation algorithm, namely, the Test

Case generation based on the Farthest Neighbor (TCFN), is also presented. The method involves partitioning the input domain into sub-domains according to the number and type of SOAP message parameters in the TCFN, selecting the candidate test case whose distance is the farthest from all executed test cases, and applying it to test the Web service. A prototype Web service vulnerability testing tool is implemented and described. The tool was applied to the testing of Web services on the Internet. The experimental results show that the proposed approach can find more vulnerability faults than other related approaches.

[4] Software vulnerabilities are the root cause of computer security problem. How people can quickly discover vulnerabilities existing in a certain software has always been the focus of information security field. This paper has done research on software vulnerability techniques, including static analysis, Fuzzing, penetration testing. Besides, the authors also take vulnerability discovery models as an example of software vulnerability analysis methods which go hand in hand with vulnerability discovery techniques. The ending part of the paper analyses the advantages and disadvantages of each technique introduced here and talks about the future direction of this field.

[5] Recently many prominent web sites face so called Distributed Denial of Service Attacks. While former security threats could be faced by tight security policy and active measures like using firewalls, Vendor paths, etc. A click jacking attack to prevent DDoS is proposed. , click jacking vulnerability can use the browser to exploit weaknesses in cross domain isolation and the same origin policy. Although there are protections available for click jacking, the web applications implementing these mitigations are far and in between. Additionally, although the possibility for an attacker to frame a page is easy to detect, it is much more difficult to demonstrate or assess the impact of a click jacking vulnerability than more traditional client-side vectors.

3.PROPOSED SYSTEM

The goal of proposed system is to add some new attack detection with addition of existing system. It is performed as close to attack sources as possible providing a protection to subscribed customers and saving valuable network resources. The distinguish packets that contain genuine sources IP address from those that contain spoofed addresses were used. First, a web page that accepts a single parameter denoting a URL that should be embedded in an IFRAME is prepared. Once the page and all contents (i.e., the IFRAME) finished loading and rendering, then it is verified that the IFRAME was still present. Pages that perform frame busting would substitute the whole content in the browser window, thus removing the IFRAME.

To automate this experiment, a Firefox extension that takes a list of URLs to be visited is implemented. Once a page is loaded, the extension waits for a few seconds and then verifies the presence of the IFRAME. If the IFRAME is not part of the document's DOM-tree anymore, then conclude that the embedded page performed frame-busting. This may lead to a message DENY. The frame busting practices of the top 500 websites are surveyed. Using both known and novel attack techniques, all of the click jacking defenses were encountered could be circumvented in one way or another.

The proposed system includes number of modules which describes about the system in detail. The overall system is thus explains about the network formation and to detect the affected nodes in it. With the help of various techniques these network is formed and thus the affected nodes are avoided from the network and also it is blocked from the network to prevent from further intrusions or attacks. It is now widely recognized that traditional approaches to cyber defense have been inadequate. Boundary controllers and filters such as firewalls and guards, virus scanners, and intrusion detection and prevention technologies have all been deployed over the last decade.

The vampire attack was successfully detected and avoided from this network with the help of above discussed techniques. As the attacked node is removed from the network then the needed session is carried out. Session cookies allow users to be recognized within a website so any page changes or item or data selection you do is remembered from page to page. The most common example of this functionality is the shopping cart feature of any e-commerce site. When you visit one page of a catalog and select some items, the session cookie remembers your selection so your shopping cart will have the items you selected when you are ready to check out. Without session cookies, if you click CHECKOUT, the new page does not recognize your past activities on prior pages and your shopping cart will always be empty.

Without cookies, websites and their servers have no memory. A cookie, like a key, enables swift passage from one place to the next. Without a cookie every time you open a new web page the server where that page is stored will treat you like a completely new visitor. To meet this, have to go for magic cookies. A magic cookie, or just cookie for short, is a token

or short packet of data passed between communicating programs, where the data is typically not meaningful to the recipient program. The contents are opaque and not usually interpreted until the recipient passes the cookie data back to the sender.

To prevent session hijacking, a special technique is proposed under which, using magic Cookie to prevent this Session hijacking attack. Magic cookie is not like a normal cookie which gets the MAC address of the machine and it convert the MAC address into some encrypted format and with enables the session cookie. Any attacker or intruder may steal the cookie/session as like normally but in this case even when attacker steal the cookie/session he/she not able to access the webpage without user credential.

4. ARCHITECTURE

In this paper we propose an architecture that detects the vampire attack in a wireless network to provide a network without any kind of attack. In a network if any misbehaving node is detected then its IP address is checked to validate or to block them from a continuous request in a network. Thus it is a measure of how well a proposed system solves the problems, and takes advantages of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development and it automatically block the weak nodes.

A. PROXY FINDER

An Internet Protocol address (IP address) is a numerical label assigned to each device (e.g., computer, printer) participating in a computer network that uses the Internet Protocol for communication. The IP address is checked because every domain having the ip address it was registered when they were launching the website. **Proxy server** is a server (a computer system or an application) that acts as an intermediary for requests from clients seeking resources from other servers. A client connects to the proxy server, requesting some service, such as a file, connection, web page, or other resource available from a different server and the proxy server evaluates the request as a way to simplify and control its complexity. Proxies were invented to add structure and encapsulation to distributed systems.

B. VERIFYING IP

Analyze the IP request to server

In computer networks, a proxy server is a server (a computer system or an application) that acts as an intermediary for requests from clients seeking resources from other servers. A client connects to the proxy server, requesting some service, such as a file, connection, web page, or other resource available from a different server and the proxy server evaluates the request as a way to simplify and control its complexity. Proxies were invented to add structure and encapsulation to distributed systems.

Reverse IP

A reverse proxy (or surrogate) is a proxy server that appears to clients to be an ordinary server. Requests are forwarded to one or more proxy servers which handle the request. The response from the proxy server is returned as if it came directly from the origin server, leaving the client no knowledge of the origin servers. Reverse proxies are installed in the neighborhood of one or more web servers. All traffic coming from the Internet and with a destination of one of the neighborhood's web servers goes through the proxy server. The use of "reverse" originates in its counterpart "forward proxy" since the reverse proxy sits closer to the web server and serves only a restricted set of websites.

C. BLOCKING IP ADDRESS

IP address blocking prevents the connection between a server or website and certain IP addresses or ranges of addresses. IP address blocking effectively bans undesired connections from hosts using affected addresses to a website, mail server, or other Internet server. IP address blocking is commonly used to protect against brute force attacks. Here the blocking of request if more than 7 continuous requests from the client side to server then that IP user can’t access the original page of the server. So the Vampire attack happened is stopped.

Server setup

The VMware is installed on various machines so that many virtual machines can be created. From these machines one machine as the server can be set up, which keeps track of the number of requests made for a service by a particular client. This helps in showing the vampire attack of a network. Thus the IP is automatically blocked but the unblocking must be done manually by the server to send requests again.

If the user to an IP changes and he also makes the same request then one have to make use of the cookies or the session id's of the particular user or the browser to authenticate the service. IP address blocking prevents the connection between a server or website and certain IP addresses or ranges of addresses. IP address blocking effectively bans undesired connections from hosts using affected addresses to a website, mail server, or other Internet server. IP address blocking effectively bans undesired connections from hosts using affected addresses to a website, mail server, or other Internet server. IP address blocking is commonly used to protect against brute force attacks.

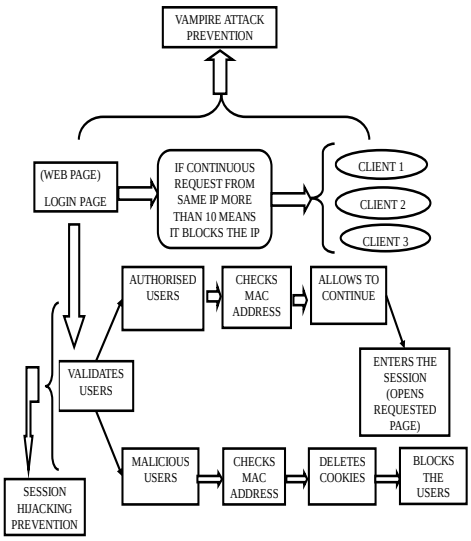

Figure 2. System Architecture.

D. COOKIE MANAGEMENT

Where the attacker uses packet sniffing to read network traffic between two parties to steal the session cookie. Many web sites use SSL encryption for login pages to prevent attackers from seeing the password, but do not use encryption for the rest of the site once authenticated. This allows attackers that can read the network traffic to intercept all the data that is submitted to the server or web pages viewed by the client. Since this data includes the session cookie, it allows him to impersonate the victim, even if the password itself is not compromised. Unsecured Wi-Fi hotspots are particularly vulnerable, as anyone sharing the network will generally be able to read most of the web traffic between other nodes and the access point.

E. MAGIC COOKIE

A magic cookie, or just cookie for short, is a token or short packet of data passed between communicating programs, where the data is typically not meaningful to the recipient program. The contents are opaque and not usually interpreted until the recipient passes the cookie data back to the sender or perhaps another program at a later time. The cookie is often used like a ticket – to identify a particular event or transaction.

F. IDLE TIMEOUT

The other type of session attack is session fixation. Here, instead of stealing/hijacking the victim's session, the attacker fixes the user's session ID before the user even logs into the target server (that is, before authentication), thereby eliminating the

need to obtain the user's session ID afterwards. Before going into detail of session fixation attacks, we must classify two types of sessions managed on Web servers:

1. Permissive sessions allow the client's browser to propose any session ID, and create a new session with that ID if one does not exist. After that, the server continues to authenticate the client with the given ID.
2. Strict sessions allow only server-side-generated session ID values.

A successful session fixation attack is generally carried out in three phases:

1. Phase I or session set-up: In this phase, the attackers set up a legitimate session with the Web application, and obtain their session ID. However, in some cases the established trap session needs to be maintained (kept alive) by repeatedly sending requests referencing it, to avoid idle session time-out.
2. Phase II or fixation phase: Here, attackers need to introduce their session ID to the victim's browser, thereby fixing the session.
3. Phase III or entrance phase: Finally, the attacker waits until the victim logs into the Web server, using the previous session ID

4. CONCLUSION

It is crucial to detect the Vampire flooding attacks at their early launching stage before widespread damages done to legitimate applications on the victim system. Internet worms that previously took days or weeks to spread now take minutes. Service providers and vendors are quickly adapting to the new landscape. Defense in depth must be practiced by service providers as zero day exploits are released. The solution utilizes user feedback to create dynamic black and white lists and overcome limitations posed by previous solutions. Here we have discussed about how we can block an IP but if the user changes their IP then the attack must not happen, so we must make use of cookies or the session id along with the IP to block a node.

After going through all the aspects of session hijacking, it can be concluded that it is successful because of unawareness in users about their security. Systems are compromised as of insecure handling, weak session IDs and mostly no account lockout. All in order to prevent this must apply the countermeasures in their daily routine of internet access. For the future work one as to prevent session hijacking attack against attacker by implementing the software like RSA ID generator that helps communication between server and client machine will be safe attacker not able to perform any sort of attack.

VI REFERENCES

[1]. Yang Xiang, Member, Ke Li, and Wanlei Zhou (2014), 'Low-Rate DDoS Attacks Detection and Traceback by using new information metrics' IEEE Transaction, VOL. 6, NO.2, JUNE, Digital Object Identifier 10.1109/TIFS.2014.2107320.

[2]. Yu Chen, Kai Hwang, and Wei-Shinn Ku (2007), 'Collaborative Detection of DDoS Attacks over Multiple Network Domains' IEEE Transactions, accepted April 10, 2007; published online June 2007.

[3]. Jinfu Chen, Huanhuan Wang, Dave Towey, Chengying Mao, Rubing Huang, and Yongzhao Zhan (2014), 'Worst-Input Mutation Approach to Web Services Vulnerability Testing Based on SOAP Messages' Volume 19, Number 5, October, ISSN 1007-0214 02/13 pp429-44.

[4]. Liu, Bingchang; Shi, Liang; Cai, Zhuhua; Li, Min; "Software vunerability Discovery Techniques : A Survey" IEEE Conference Publication, DOI : 10.1109/MINES.2012.202, Page(s) 152-156, 2012.

[5]. Chonka *et al.*, (2010), 'Cloud security defence to protect cloud computing against HTTP-DoS and XML-DoS attacks', Jun. 23.

[6]. Smith, Yurick, Doss "Ethical Hacking" IEEE Conference Publication, DOI: 10.1147/sj.403.0769, Page(s): 769-780.

[7]. Bradley, Rubin "Computer Security Education and Research: Handle with care" IEEE Conference Publication, DOI : 10.1109/MSP.2006.146, Page(s): 56-59.

[8]. Wilbanks "When Black Hats are really white" IEEE Conference Publication, DOI: 10.1109/MITP.2008.146, Page(s): 64.

[9]. X.-S. Yang, S. Deb, "Cuckoo search via Levy flights", in: World Congress on Nature and Biologically Inspired Computing (NaBIC 2009). IEEE Publication, USA. 2009, pp. 210–214.

[10]. X.-S. Yang, S. Deb, S. Fong, "Accelerated Particle Swarm Optimization and Support Vector Machine for Business Optimization and Applications", The Third International Conference on Networked Digital Technologies (NDT 2011), Springer CCIS 136, 11-13 July 2011, Macau, pp.53–66.

[11]. X.-S. Yang, "A New Metaheuristic Bat-Inspired Algorithm", in: Nature Inspired Cooperative Strategies for Optimization (NISCO 2010), Eds. J.R. Gonzalez et al., Studies in Computational Intelligence, Springer Berlin, 284, Springer.

[12]. M. Prandini, M. Ramilli, W. Cerroni, and F. Callegati. Splitting the HTTPS Stream to Attack Secure Web Connections. IEEE Security and Privacy, 8:80–84, 2010.

[13]. T. Choi and M. G. Gouda. HTTPI: An HTTP with Integrity. In in Proceedings of International Conference on Computer Communications and Networks (ICCCN), 2011.

[14]. A. Bortz, A. Barth, and A. Czeskis. Origin Cookies: Session Integrity for Web Applications. In Web 2.0 Security and Privacy Workshop (W2SP), 2011.

[15]. B. Adida. Beamauth: two-factor web authentication with a bookmark. In Proceedings of the ACM Conference on Computer and Communications Security (CCS), 2007.

[16]. B. Adida. Sessionlock: securing web sessions against eavesdropping. In Proceeding of the ACM international conference on World Wide Web (WWW), 2008.

[17]. C. Blundo, S. Cimato, and R. D. Prisco. A Lightweight Approach to Authenticated Web Caching. In Proceedings of the The Symposium on Applications and the Internet, 2005.