2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing (HPSC), and IEEE Intl Conference on Intelligent Data and Security (IDS)

A Survey on Deception Techniques for Securing Web Application

Mohd Efendi, A.I[1,b], Ibrahim, Z. [2,a,b], Ahmad Zawawi, M.N[3,a,b], Abdul Rahim, F. [4,a,b]
[a]Institute of Informatics and Computing in Energy, Universiti Tenaga Nasional, 43000, Malaysia
[b]College of Computing & Informatics, Universiti Tenaga Nasional, 43000, Malaysia
[1]CS0100985@utn.edu.my, [2]zulazri@uniten.edu.my, [3]mdnabil@uniten.edu.my, [4]fiza@uniten.edu.my

Mohamad Pahri,N.A [4,a], Ismail, A.[5,a],
[c]ASK-Pentest, Menara Keck Seng, 55100 Kuala Lumpur, Malaysia
[4]*azuwa@ask-pentest.com*, [5]*anuar@ask-pentest.com*

***Abstract* - Many web applications are developed to handle important and critical tasks, which may attract a large number of attackers. With various types of attacks, there is no finite solution to mitigate it's all. Deception technique is one of the area that can be explore to defend against web attack. Deception can detect, analyzed and defend against advanced web attack that cannot be done using existing anomaly-based detection and prevention techniques. Current deceptive solutions tend to be doubtful to application-layer protocols and lack of study on how deception can be applied at this level. Thus, those solutions can't properly be used to protect against application-layer attacks that are integrally based on elements from the application-layer itself. This research aims to study possible usages of deception techniques that could be incorporated in the context of application-layer traffic of web applications with the purpose of detecting web application attacks. The comparative results from this study will be used to identify which deception techniques are suitable to provide a useful layer of protection for a web application.**

Keywords—deception, security, web application

I. INTRODUCTION

The popularity of the Internet has made the growing number of web applications delivered over the HTTP protocol. Web application provide services in our everyday life in a variety of fields such as healthcare, commerce, education, critical infrastructure and etc. Many web applications are developed to handle important and critical tasks, which may attract a large number of attackers. According to Symantec, attackers usually compromise web servers and insert malicious code, and later enable to redirect victim to the exploit kit servers [1]. Other attacks aimed at the application layer such as Cross Site Scripting (XSS), SQL injection and parameter tampering.

With various types of attacks, there is no single solution to mitigate all of them. For example, injection and broken authentication, and XSS vulnerabilities have occupied the top 10 issues of the OWASP Top 10 2017 edition [2]. According to Trustwave's 2017 Global Security Report [3], 99.7% of tested web applications were found at least one vulnerability. Furthermore, existing anomaly-based detection and prevention techniques unable to handle with the large number and sophisticated attacks. A large number of techniques have been proposed to secure web applications. Li et al. [4] has emphasized on the importance of securing construction of new web application and preservation of legacy applications through analysis, testing and runtime protection. However, these countermeasures still unable to equip an inclusive solution against Internet threats. There is a need to provide new additional layers of protection by helping user to anticipating threats and possibly warn users against attacks in their early stages [5].

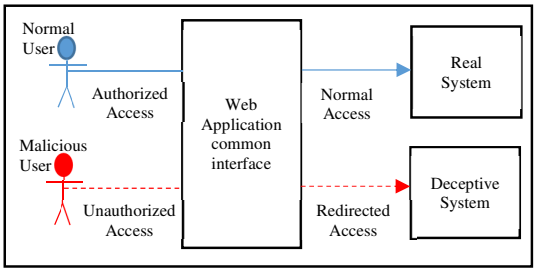

Figure 1. Basic idea of deception

Deception may provide an alternative approach to defend web applications that can deliver an advantageous additional layer of protection. Deception work like honeypot where it tried lured attacker to penetrate a system that imitate all functionality of the real system with preconfigured faults and study the behavior and methods that attacker used after manage to hack into the imitate system shown in Figure 1. In this case the real system would not be affected by the attacker as they assume they are hacking into the real system and all the attacker activity were monitored by the administrator in the fake system [6]. Still the use of deception has traditionally been limited to ad-hoc approaches realized as single tools or to repackaged entire solutions deployed as isolated honeypot machines. Indeed, deception may provide a valuable additional layer of protection that could potentially be integrated into web applications that is susceptible of suffering attacks, including application-layer protocols such as HTTP. Current deceptive solutions tend to be doubtful to application-layer protocols and lack of study on how deception can be applied at this level. Hence, this study aims to study possible usages of deception techniques that could be incorporated in the context of application-layer of web applications with the purpose of detecting web application attacks.

978-1-7281-0006-7/19/$31.00 ©2019 IEEE
DOI 10.1109/BigDataSecurity-HPSC-IDS.2019.00066

This paper is structured as follows: Section II reviews on web application attacks. Section III explains the web application attack detection. Next, section IV discusses on deception techniques to detect web application attacks. Lastly, Section V presents conclusion and future works.

II. Web Application Attacks

Web applications are client-server applications that utilizes web browsers and web technology. Web applications commonly use a combination of server-side script (ASP, PHP, etc.) and client-side script (HTML, Javascript, etc.) to develop the application. The client-side script deals with the presentation of the information while the server-side script deals with storing and retrieving the information.

As the number of businesses embracing the benefits of doing business over the web rises, so will the use of web applications and other related technologies continue to grow. With the growing number of web applications, it also raises a number of security concerns. Serious vulnerabilities may allow hackers to gain direct and public access to web system databases in order to agitate sensitive data. Many of these databases contain valuable information (e.g., personal and financial details) making them a frequent target of hackers. Some hackers, for example, may maliciously inject code within vulnerable web applications to hoax users and redirect them towards phishing sites [7][8].

According to [4] web application happen to possess three types of security weaknesses that can be exploit by attackers; Input validation, Session management and Application logic. Input validation can happen when attacker able to inject malformed inputs that can alter program executions and gain unauthorized access to system resources. Session management vulnerability happen when attacker manage to hijack the communication session between client and server. Then Application logic is an attack where attacker can exploit the hidden link in the web system that allow them to access unauthorized information. Figure 2 show the summary of web application vulnerabilities and example of attack that can happen within each vulnerability.

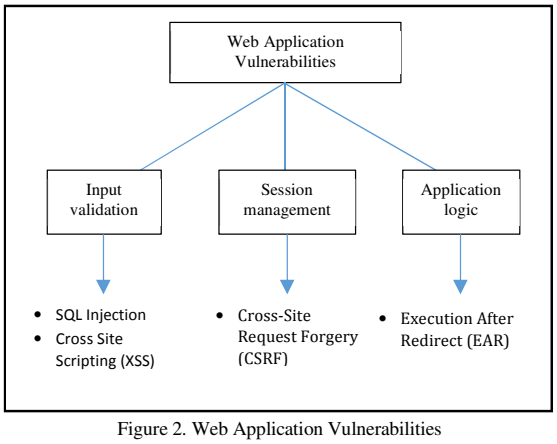

Figure 2. Web Application Vulnerabilities

Normally web system will used password based-authentication to protect against unauthorized access as it is easy to implement and did not required any additional devices to used. Even so attacker can still gain access to the system by using brute-force attack where the attacker will try to guest combination of password used by the user. To make thing worst, the lack of user awareness which used weak combination of password can make it easier for attacker to penetrate the system [9].

Another type of web application attacks that pose a severe threat to the availability of web applications is Denial-of-Service (DoS). DoS attacks may also cause significant harm by severely degrading the performance. Then come the Distributed DoS (DDoS) which an upgrade from DoS. DDoS attack happen at network layer, where bunch of compromise workstation called zombies will flood the web resources such web application bandwidth and data by making fault request to the web application. Currently this attack has become more severed as it already evolved where attacker can launch this attack in application layer which make it difficult to protect against it. DDoS in application layer will required less zombie workstation but the traffics looks more legitimate compare to the past network layer DDoS[10].

III. Web Application Attack Detection

To handle these web application threat, prevention technologies such as access controls, email gateways, intrusion detection/prevention systems (IDS/IPS), network firewalls, proxy servers, and web application firewalls (WAFs) are important foundations of cyber security [11] to detect and block the attack. However, these prevention techniques are not enough to handle with the large number and sophisticated nowadays attacks. As the method of attack evolve through time, there is a need to study the method and behavior of the current attacks, which can be done using honeypot and deception. These approach will try to deceive web-based attacks also consists of using fake information disguised as web server configuration errors [12].

Solutions to detect threats and attacks can be either anomaly-based or deception-based. Similar with prevention technologies, anomaly-based detections are not capable to handle sophisticated attacks. Such disadvantages in anomaly-based solutions are complexity, expensive and time consuming. In addition, anomaly-based detection may create a high rate of false positives, adding a significant burden to the monitoring team [11].

In contrast, deception-based detection provides an effective alternative to anomaly-based detection. Any component in an enterprise network such as a computer system, a service, a credential, a data item, and so on, which can be used for deception-based detection. The first generation of deception technologies is honeypot, demonstrated the effectiveness of deception as part of a layered security strategy [13][14]. Several deception techniques have been proposed to be integrated into the application layer of web applications such as employing two mechanisms based on parameters injection into web application traffic [15], and deploying deception proxy to detect any possible false alarm [5]. Hence, there is still a gap to fill when it comes to the development of new additional techniques to diverse types of web application attacks.

Taking everything into consideration, it reveals that there is a lack of works addressing the specific use of deception strategies that are incorporated into the application-layer protocol aiming to detect or prevent web application attacks.

IV. Deception Techniques to Protect Against Web Application Attacks

Cyber-attacks are similar with military situations in that the attacker has a direct advantage of being the initiator. In cyber environment, cyber attackers have the added advantages to become unknown with their malicious intention and might not be noticeable to the defender until they succeed [16]. To be compared with tangible valuable assets such as money, sensitive information stored in an organization tends to be leaked or breached due to low-level protection [17].

A numerous amount of research has been done in the field of deception, ranging from military domain to recent studies within the cyber domain. Deception techniques, if used correctly, can place defenders a step ahead of attackers, by modifying the system with additional traps that increase the possibility of being detected. A defender should put an effort to detect and mitigate attackers as early as possible within the cyber kill chain in order to stop the attack [18].

The cyber kill chain framework introduced by Lockheed Martin researchers used for identification and prevention of cyber intrusions activity [19]. The main objective behind this framework is to understand each of these stages, in order to identify and stop attacks at early stage. The earlier a kill chain attack can be stopped, the better for the defenders to prevent the attackers from attacking the systems.

With the same objective of the kill-chain framework, which is early detection of attackers, deception strategies should enable the defenders to deceive attackers while learning the methods and techniques performed by the attackers. Combining this with intrusion detection mechanism, the defenders could have a better position and be one step ahead of the attackers.

There are different ways to deploy deception techniques into an information system, such as through the network, system, application, and data layers [12]. In the network layer, deception techniques can be deployed over the network and not attach to any specific host. For system layer, deception techniques will be attached to the host. While, the application layer covers deception techniques that are associated to the application components, such as web applications or databases. Finally, the data layer covers deception techniques that use fake data to deceive attackers. In this paper, the deception method in the application layer, specifically on the web application will be discussed.

Various deception techniques proposed by academia, as well as practices by the industry. Whaley define and categorize deception into two main categories: dissimulation and simulation [20]. Dissimulation involves the process of hiding real information through disguised and misrepresentation of information towards possible unauthorized probing. Simulation on the other hand involves a process of making a process or service anonymous in order to detach it from being tied to a specific process. Table I list both summarized deception method key features and we elaborate it further in the following section. Although listed individually, there exists possibility that the method could be implemented in combination or successive to add to its effectiveness as countermeasures to possible attacks.

TABLE I. Deception Techniques [20]

Dissimulation	***Simulation***
Masking: Conceal relevant information about the item by making the truth hidden with cover information.	Mimicking: Imitate mechanisms by portraying other services instead of showing its real services.
Repackaging: Introduced hidden functions to decoy files, process and services	Inventing: Create a simulated service to draw attacker's resources away from actual service.
Dazzling: Introduced extra noise to information on top of already available information.	Decoying: Distract the attention by attracting the attackers' attention away from the truth

A. Masking

Masking usually involves camouflage, mislabelling and producing false associated plans to hide the actual data. In web application, defenders could use it to hide specific software and services from the attackers. For example, defenders may hide SSH demon and respond as if the service is not working or as if it is encountering an error whenever they receive an SSH connection request from a known bad IP address [21]. Another example is by placing honey-files that contain honey accounts to monitor and gather information to detect attackers [22]. By using honey accounts to draw in the attackers, it may bring the targeted user to another environment, a decoy environment that is isolated from the real service. Masking can also be used to insert hidden program in a relatively benign looking program, which allow the defenders to observe the attacker's activity once activated.

B. Repackaging

Repackaging involves adding hidden functions to decoy files, processes and services. Defenders may repackage their honey files as normal files. In this situation, the honey files is repackaged to act as silent alarms to system administrators when opened [21]. In deploying this technique, defenders may also repackage their framework by adding additional cookies, hidden input form fields additional parameters, which enables the possible attacks such as SQL injection. Repackaging may also consist of the hiding JavaScript code that make it appear as something it is not. Once the attacker activates the code, it may log the activity of the attackers, which allow the defenders to monitor possible attack.

C. Dazzling

Dazzling involves confusing attackers with additional noise of information. In the web application's database for example, defenders may confuse each user's hashed password with an extra (N – 1) hashes of other, similar, passwords dazzling the attacker who obtains the credentials database [21]. Methods to tempt confusion also include randomization of elements within the object. Taking example from the credential database extracted by attackers, password dazzling may confuse the attacker by randomization of elements within the rows. When the randomized passwords are being used by the attacker, the defender can check the record from log and able to increase the protection level of their actual credential database.

D. Mimicking

Mimicking enables process and service to appear as other services. For instance, defenders can use it to respond to attacker's request as if running on MySQL, while the application is actually running on Oracle. This will excess attackers' resources and time in trying to exploit Oracle thinking it is MySQL, as well as increase the opportunity for the defenders to discover and learn attacker's method. In another example, defenders may include a set of fake

vulnerabilities, which will maintain the attackers busy trying to exploit the vulnerabilities that does not exist [5].

E. Inventing

Inventing involves creating a simulated service and process to draw attacker's resources away from the actual service. Defenders may deploy a new protected area to invent a number of systems in their organization to deceive an attacker that the fake systems are real systems. For example, an administrative console is created that requires HTTP authentication, looks exactly similar with normal administrative console, but it contains no content and may lure the attackers to brute-force the system to gain access [5].

F. Decoying

Decoying is used to manipulate an attacker into believing they successfully found something fruitful and successfully used an exploit to access the protected resources. Defenders may use this technique to attract attackers' attention away from the most valuable parts of their web application [23]. For example, defenders may insert hidden hyperlinks (decoys) in a number of web pages that point to a decoy web page [24]. This configuration can be used to detect Web-DoS attack, which the attacker's IP address can be blocked for a period of time once the attacks are detected. The crucial part in this configuration is to hide the decoy hyperlinks between pages, hence will increase the probability of the decoy hyperlink to be clicked during the attack.

V. Conclusion and Future Works

In this paper, we have addressed various deception techniques through reviewing past literatures that could be used as mechanisms detect web application attacks and at the same time act as a decoy and protection mechanism to draw the attacker away from actual protected services and processes. The techniques, although listed individually, could also be executed a combined together to enhance its effectiveness. In the next stage of this study, a testing environment will be prepared that contain a software implementation of the deception techniques that will be used as protective mechanism to a web application service. A set of penetration testing tools will be used to launch attacks. The observation will be analyzed based on three general outputs from the use of deception techniques. The attacker might (1) believe it, (2) suspect it or (3) disbelieve it. The evaluation will be focused on monitoring attacker's reaction of the deception techniques that have been deployed in the environment.

Acknowledgment

We would like to express profound gratitude to ASK-PENTEST SDN.BHD. for their useful suggestion for this project. Work presented in this paper forms part of the research on Effective Deception Techniques for Securing Web Application, which was funded by Universiti Tenaga Nasional Internal Research Grant scheme (UNIIG) and Intelligent Deception Attack Detection Algorithm for Industry Control System funded by Seed Fund provided by Tenaga Nasional Berhad (TNB), the parent company of UNITEN.

References

[1] Symantec, "Internet Security Threat Report 2017," 2017.

[2] OWASP, "OWASP Top 10 Application Security Risks - 2017," *2017.* .

[3] Trustwave, "2017 Trustwave Global Security Report," 2017.

[4] X. Li and Y. Xue, "A survey on server-side approaches to securing web applications," *ACM Comput. Surv.*, 2014.

[5] X. Han, N. Kheir, and D. Balzarotti, "Evaluation of Deception-Based Web Attacks Detection," *Proc. 2017 Work. Mov. Target Def. - MTD '17*, pp. 65–73, 2017.

[6] B. Djamaluddin, A. Alnazeer, and F. Azzedin, "Web Deception Towards Moving Target Defense," *Proc. - Int. Carnahan Conf. Secur. Technol.*, vol. 2018–Octob, pp. 1–5, 2018.

[7] M. M. Moreno-Fernández, F. Blanco, P. Garaizar, and H. Matute, "Fishing for phishers. Improving Internet users' sensitivity to visual deception cues to prevent electronic fraud," *Comput. Human Behav.*, vol. 69, pp. 421–436, 2017.

[8] C. Iuga, J. R. C. Nurse, and A. Erola, "Baiting the hook: factors impacting susceptibility to phishing attacks," *Human-centric Comput. Inf. Sci.*, vol. 6, no. 1, p. 8, 2016.

[9] C. Adams, G. V. Jourdan, J. P. Levac, and F. Prevost, "Lightweight protection against brute force login attacks on web applications," *PST 2010 2010 8th Int. Conf. Privacy, Secur. Trust*, pp. 181–188, 2010.

[10] A. Praseed and P. Santhi Thilagam, "DDoS attacks at the application layer: Challenges and research perspectives for safeguarding web applications," *IEEE Commun. Surv. Tutorials*, vol. 21, no. 1, pp. 661–685, 2019.

[11] Team Acalvio, *Deception 2.0 for dummies*. 2017.

[12] X. Han, N. Kheir, and D. Balzarotti, "Deception Techniques in Computer Security," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–36, 2018.

[13] F. Cohen, "The Use of Deception Techniques: Honeypots and Decoys," *Handb. Inf. Secur. Vol. III, Part1*, vol. 3, no. 1981, p. p646, 2005.

[14] M. T. Qassrawi and Z. Hongli, "Deception methodology in virtual honeypots," *NSWCTC 2010 - 2nd Int. Conf. Networks Secur. Wirel. Commun. Trust. Comput.*, vol. 2, pp. 462–467, 2010.

[15] T. Ishikawa and K. Sakurai, "Parameter manipulation attack prevention and detection by using web application deception proxy," in *Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication - IMCOM '17*, 2017.

[16] B. Gupta and K. Jyoti, "Big Data Analytics with Hadoop to analyze Targeted Attacks on Enterprise Data," *Int. J. Comput. Sci. Inf. Technol.*, 2014.

[17] G. Pan, S. P. Sun, C. Chan, and L. Chu Yeong, *Analytics and Cybersecurity: The shape of things to come*. 2015.

[18] V. E. Urias, W. M. S. Stout, J. Luc-Watson, C. Grim, L. Liebrock, and M. Merza, "Technologies to enable cyber deception," *Proc. - Int. Carnahan Conf. Secur. Technol.*, vol. 2017–Octob, no. 1303051, pp. 1–6, 2017.

[19] Lockheed Martin Corporation, "Cyber Kill Chain® | Lockheed Martin." .

[20] B. Whaley, "Toward a general theory of deception," *J. Strateg. Stud.*, vol. 5, no. 1, pp. 178–192, Mar. 1982.

[21] S. Jajodia, V. S. Subrahmanian, V. Swarup, and C. Wang, "Cyber deception: Building the scientific foundation," *Cyber Decept. Build. Sci. Found.*, pp. 1–312, 2016.

[22] B. M. Bowen, S. Hershkop, A. D. Keromytis, and S. J. Stolfo, "Baiting inside attackers using decoy documents," in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, 2009.

[23] D. Fraunholz *et al.*, "Demystifying Deception Technology:A Survey," 2018.

[24] D. Gavrilis and I. Chatzis, "Detection of Web Denial-of-Service Attacks using decoy hyperlinks," *5th Int. Symp.*, 2006.