SOCIAL ENGINEERING IN CYBER SECURITY EFFECT MECHANISMS, HUMAN VULNERABILITIES AND ATTACK METHODS

1.DR. GHOUSE BASHA, 2. THAKUR SAI MAHITHA,3. T. VISHNU PRIYA, 4. SEEDHIKA PATNAIK

1.PROFESSOR, 2,3&4 UG SCHOLAR

DEPARTMENT OF ECE, MALLA REDDY ENGINEERING COLLEGE FOR WOMEN, HYDERABAD

ABSTRACT

Social engineering attacks have posed a serious security threat to cyberspace. However, there is much we have yet to know regarding what and how lead to the success of social engineering attacks. This paper proposes a conceptual model which provides an integrative and structural perspective to describe how social engineering attacks work. Three core entities (effect mechanism, human vulnerability and attack method) are identified to help the understanding of how social engineering attacks take effect. Then, beyond the familiar scope, we analyze and discuss the effect mechanisms involving 6 aspects (persuasion, social influence, cognition & attitude& behavior, trust and deception, language &thought &decision, emotion and decision-making) and the human vulnerabilities involving 6 aspects (cognition and knowledge, behavior and habit, emotions and feelings, human nature, personality traits, individual characters), respectively. Finally, 16 social engineering attack scenarios (including 13 attack methods) are presented to illustrate how these mechanisms, vulnerabilities and attack methods are used to explain the success of social engineering attacks. Besides, this paper offers lots of materials for security awareness training and future empirical research, and the model is also helpful to develop a domain ontology of social engineering in cybersecurity.

INDEXTERMS—Machine learning, cross-site request forgery, web security.

INTRODUCTION

In the context of computer and cyber security, social engineering describes a type of attack in which the attacker exploit human vulnerabilities by means such as influence, persuasion, deception, manipulation and inducing, so as to get classified information, hack computer system and network, obtain unauthorized access to restricted areas, or breach the security goals (such as confidentiality, integrity, availability, controllability and auditability) of cyberspace elements (such as infrastructure, data, resource, user and operation). Succinctly, social engineering is a type of attack wherein the attacker exploit human vulnerability through social interaction to breach cyberspace security [1]. In hacker community, social engineering is a quite popular attack since 1970s. Compared to classical computer attacks such as password cracking by brute-force and software vulnerabilities exploit, social engineering attacks focus the exploitation of human vulnerabilities, to bypass or break through security barriers, without having to combat with firewall or antivirus software by deep coding. In addition, there is not a computer system doesn’t rely on humans or involves human factors on earth, and these human factors are obviously vulnerable or can be largely turned into security vulnerabilities by skilled attackers. These inevitable and vulnerable human factors makes social engineering to be a universal cybersecurity threat. For some situations, social engineering attacks may be as simple as making a phone call and impersonating an insider to elicit the classified information. Moreover, with the development of new technology and the formation of new cyber-environment, social engineering threat is increasingly serious. Social Network Sites (SNSs), mobile communication, Industrial Internet and Internet of Things (IoT) generate not only large amounts of sensitive information about people and devices but also more attack channels and a bigger attack surface. Unrestricted office environment (bring your own device, remote office, etc.) leads to the weakening of area-isolation of different security levels and creates more attack opportunities. The easy availability of open source intelligence simplifies the information gathering. Specific targets can be carefully selected to craft more creditable and targeted social engineering attacks. A large group of victims can be reached at the same time and some open source tools can be used to launch semi-automated attacks. Technologies such as machine learning and artificial intelligence is likely to make social engineering attacks more efficient and aggressive. Targeted, large-scale, robotic, automated and advanced social engineering attack is becoming possible [1]. Social engineering is evolving to be a serious, universal and persistent security threat. To protect against social engineering attack, an important work is to understand how it works and takes effect. This paper makes the following contributions. - An integrative and structural model to describe how social engineering attacks work and take effect. - Three core entities to get an insight into social engineering attacks. - 30+ effect mechanisms involving 6 aspects. - 40+ human vulnerabilities involving 6 aspects. - Case study of 16 social engineering attack scenarios (including 13 type attack methods).

LITERATURE SURVEY

1) Surviving The Web: A Journey Into Web Session Security

AUTHORS:Stefano Calzavara, Riccardo Focardi, Marco Squarcina, and Mauro Tempesta The Web is the primary access point to on-line data and applications. It is extremely complex and variegate, as it integrates a multitude of dynamic contents by different parties to deliver the greatest possible user experience. This heterogeneity makes it very hard to effectively enforce security, since putting in place novel security mechanisms typically prevents existing websites from working correctly or negatively affects the user experience, which is generally regarded as unacceptable, given the massive user base of the Web However, this continuous quest for usability and backward compatibility had a subtle effect on web security research: designers of new defensive mechanisms have been extremely cautious and the large majority of their proposals consists of very local patches against very specific attacks. This piecemeal evolution hindered a deep understanding of many subtle vulnerabilities and problems, as testified by the proliferation of different threat models against which different proposals have been evaluated, occasionally with quite diverse underlying assumptions. It is easy to get lost among the multitude of proposed solutions and almost impossible to understand the relative benefits and drawbacks of each single proposal without a full picture of the existing literature. In this work, we take the delicate task of performing a systematic overview of a large class of common attacks targeting the current Web and the corresponding security solutions proposed so far. We focus on attacks against web sessions, i.e., attacks which target honest web browser users establishing an authenticated session with a trusted web application. This kind of attacks exploits the intrinsic complexity of the Web by tampering, e.g., with dynamic contents, client-side storage or cross-domain links, so as to corrupt the browser activity and/or network communication. Our choice is motivated by the fact that attacks against web sessions cover a very relevant subset of serious web security incidents and many different defenses, operating at different levels, have been proposed to prevent these attacks.We consider typical attacks against web sessions and we systematise them based on: (i) their attacker model and (ii) the security properties they break. This first classification is useful to understand precisely which intended security properties of a web session can be violated by a certain attack and how. We then survey existing security solutions and mechanisms that prevent or mitigate the different attacks and we evaluate each proposal with respect to the security guarantees it provides. When security is guaranteed only under certain assumptions, we make these assumptions explicit. For each security solution, we also evaluate its impact on both compatibility and usability, as well as its ease of deployment. These are important criteria to judge the practicality of a certain solution and they are useful to understand to which extent each solution, in its current state, may be amenable for a large-scale adoption on the Web. Moreover, since there are several proposals in the literature which aim at providing robust safeguards against multiple attacks, we also provide an overview of them. For each of these proposals, we discuss which attacks it prevents with respect to the attacker model considered in its original design and we assess its adequacy according to the criteria described above.

2) Large-Scale Analysis & Detection Of Authentication Cross-Site Request Forgeries
AUTHORS: Avinash Sudhodanan, Roberto Carbone, Luca Compagna, Nicolas Dolgin,

Alessandro Armando, and Umberto Morelli Cross-Site Request Forgery (CSRF) attacks are one of the critical threats to web applications. In this paper, we focus on CSRF attacks targeting web sites' authentication and identity management functionalities. We will refer to them collectively as Authentication CSRF (Auth-CSRF in short). We started by collecting several Auth-CSRF attacks reported in the literature, then analyzed their underlying strategies and identified 7 security testing strategies that can help a manual tester uncover vulnerabilities enabling Auth-CSRF. In order to check the effectiveness of our testing strategies and to estimate the incidence of Auth-CSRF, we conducted an experimental analysis considering 300 web sites belonging to 3 different rank ranges of the Alexa global top 1500. The results of our experiments are alarming: out of the 300 web sites we considered, 133 qualified for conducting our experiments and 90 of these suffered from at least one vulnerability enabling Auth-CSRF (i.e. 68%). We further generalized our testing strategies, enhanced them with the knowledge we acquired during our experiments and implemented them as an extension (namely CSRF-checker) to the open-source penetration testing tool OWASP ZAP. With the help of CSRFchecker, we tested 132 additional web sites (again from the Alexa global top 1500) and identified 95 vulnerable ones (i.e. 72%). Our findings include serious vulnerabilities among the web sites of Microsoft, Google, eBay etc. Finally, we responsibly disclosed our findings to the affected vendors.

3) State Of The Art: Automated Black-Box Web Application Vulnerability Testing

AUTHORS: Jason Bau, Elie Bursztein, Divij Gupta, and John C. Mitchell

Black-box web application vulnerability scanners are automated tools that probe web applications for security vulnerabilities. In order to assess the current state of the art, we obtained access to eight leading tools and carried out a study of: (i) the class of vulnerabilities tested by these scanners, (ii) their effectiveness against target vulnerabilities, and (iii) the relevance of the target vulnerabilities to vulnerabilities found in the wild. To conduct our study we used a custom web application vulnerable to known and projected vulnerabilities, and previous versions of widely used web applications containing known vulnerabilities. Our results show the promise and effectiveness of automated tools, as a group, and also some limitations. In particular, "stored" forms of Cross Site Scripting (XSS) and SQL Injection (SQLI) vulnerabilities are not currently found by many tools. Because our goal is to assess the potential of future research, not to evaluate specific vendors, we do not report comparative data or make any recommendations about purchase of specific tools.

4) Why johnny can’t pentest: An analysis of black-box web vulnerability scanners
AUTHORS :Adam Doup´e, Marco Cova, and Giovanni Vigna

Black-box web vulnerability scanners are a class of tools that can be used to identify security issues in web applications. These tools are often marketed as “point-and-click pentesting” tools that automatically evaluate the security of web applications with little or no human support. These tools access a web application in the same way users do, and, therefore, have the advantage of being independent of the particular technology used to implement the web application. However, these tools need to be able to access and test the application’s various components, which are often hidden behind forms, JavaScript-generated links, and Flash applications. This paper presents an evaluation of eleven black-box web vulnerability scanners, both commercial and open-source. The evaluation composes different types of vulnerabilities with different challenges to the crawling capabilities of the tools. These tests are integrated in a realistic web application. The results of the evaluation show that crawling is a task that is as critical and challenging to the overall ability to detect vulnerabilities as the vulnerability detection techniques themselves, and that many classes of vulnerabilities are completely overlooked by these tools, and thus research is required to improve the automated detection of these flaws.

5)Mitch: A Machine Learning Approach To The Blackbox Detection Of Csrf VulnerabilitiesAUTHORS:Stefano Calzavara, Mauro Conti, Riccardo Focardi, Alvise Rabitti, andGabriele Tolomei

Cross-Site Request Forgery (CSRF) is one of the oldest and simplest attacks on the Web, yet it is still effective on many websites and it can lead to severe consequences, such as economic losses and account takeovers. Unfortunately, tools and techniques proposed so far to identify CSRF vulnerabilities either need manual reviewing by human experts or assume the availability of the source code of the web application. In this paper we present Mitch, the first machine learning solution for the black-box detection of CSRF vulnerabilities. At the core of Mitch there is an automated detector of sensitive HTTP requests, i.e., requests which require protection against CSRF for security reasons. We trained the detector using supervised learning techniques on a dataset of 5,828 HTTP requests collected on popular websites, which we make available to other security researchers. Our solution outperforms existing detection heuristics proposed in the literature, allowing us to identify 35 new CSRF vulnerabilities on 20 major websites and 3 previously undetected CSRF vulnerabilities on production software already analyzed using a state-of-the-art tool.

EXISTING SYSTEM:

Social engineering is an interdisciplinary field which involves computer science, cybersecurity, psychology, social psychology, cognitive science, psycholinguistics, neuroscience, brain science, etc. In work [1], human vulnerabilities such as credulity, greed, ignorance, curiosity, carelessness, helpfulness have been mentioned. Yet only the human vulnerabilities are not sufficient to describe how social engineering attacks take effect.

For effect mechanism, some works discussed or involved it in different context. Many scholars, e.g. [26], [78], [82]_[84], employ Cialdini's [5], [85] six principles of influence and persuasion (reciprocation, commitment and consistency, social proof, liking, authority, scarcity) to explain the success of social engineering attacks. Literature [86], [87] also discussed some psychological principles that exhibit some kind of power to influence or persuade people and take effect during a social engineering attack (strong affect, overloading, reciprocation, deceptive relationships, diffusion of responsibility and moral duty, authority, integrity and consistency).

Mitnick and Simon [17] describes social engineering based on various kinds of deception. Stajano andWilson [88] discussed seven principles of scam for system security (distraction, social compliance, herd, dishonesty, kindness, need and greed, time). Ferreira *et al.* [89] analyzed the relation (equal, include, overlap) among the above principles and presented a merged list of social engineering persuasion principles, i) authority, ii) social proof, iii) liking, similarity & deception, iv) commitment, reciprocation & consistency, v) distraction. However, the human vulnerabilities were not carefully concerned in theseworks, and other aspects of effect mechanisms are not involved.

PROPOSED SYSTEM:

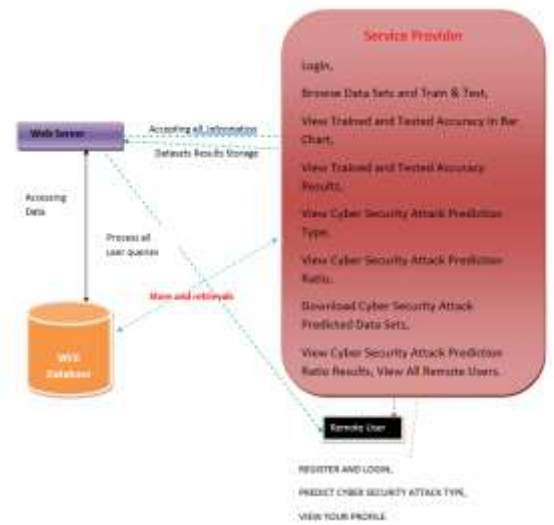
There are three basic perspectives to understand how social engineering attacks take effect in the proposed system

_ From the attacker perspective, the attack method is the way, manner or means of carrying an attack out; it is also the driving force that directly causes a social engineering attack and significantly affects whether the attack can succeed. After all, the advanced and ingenious attack methods usually possess a greater success rate to obtain the attack goals.

_ From the victim perspective, the exploited human vulnerabilities are the root reason why the victim brings about the attack consequences. As one of the confrontational focuses between social engineering attack and defense, human vulnerability is what attackers want to exploit and what victims want to eliminate or mitigate. Other types of vulnerability (e.g. software vulnerabilities) can be exploited together with human vulnerability, yet they are not necessary in social engineering

From the perspective of principle and explaining, effect mechanisms explain how attack methods make the human vulnerabilities take effect. Effect mechanisms describe [R1] how attack methods exploit human vulnerabilities, and explain [R2] why the human vulnerabilities leads to the attack consequences as well as (corresponding to) [R3] how the attack methods achieve the attack goals. In other words, effect mechanisms can be defined as the structural relation that what, why or how specific attack effects consequences) correspond to specific human vulnerabilities, in specific attack scenarios.

SYSTEM ARCHITECTURE


Service Provider
Login,
Browse Data Sets and Train & Test,
View Trained and Tested Accuracy in Bar Chart,
View Trained and Tested Accuracy Results,
View Cyber Security Attack Prediction Type,
View Cyber Security Attack Prediction Ratio,
Download Cyber Security Attack Predicted Data Sets,
View Cyber Security Attack Prediction Ratio Results, View All Remote Users.
Web Server
Accessing all Information
Datasets Results Storage
Accessing Data
Process all user queries
Store and retrievals
WEB Database
Remote User
REGISTER AND LOGIN,
PREDICT CYBER SECURITY ATTACK TYPE,
VIEW YOUR PROFILE.

Service Provider

In this module, the Service Provider has to login by using valid user name and password. After login successful he can do some operations such as
Login, Browse Data Sets and Train & Test, View Trained and Tested Accuracy in Bar Chart, View

Trained and Tested Accuracy Results,

View Cyber Security Attack Prediction Type, View Cyber Security Attack Prediction Ratio, Download Cyber Security Attack Predicted Data Sets, View Cyber Security Attack Prediction Ratio Results,, View All Remote Users.

View and Authorize Users

In this module, the admin can view the list of users who all registered. In this, the admin can view the user's details such as, user name, email, address and admin authorizes the users.

Remote User

In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user will do some operations like REGISTER AND LOGIN, PREDICT CYBER SECURITY ATTACK TYPE, VIEW YOUR PROFILE.

CONCLUSION

Web applications are particularly challenging to analyse, due to their diversity and the widespread adoption of custom programming practices. ML is thus very helpful in the web setting, because it can take advantage of manually labeled data to expose the human understanding of the web application semantics to automated analysis tools. We validated this claim by designing Mitch, the first ML solution for the blackbox detection of CSRF vulnerabilities, and by experimentally assessing its effectiveness. We hope other researchers might take advantage of our methodology for the detection of other classes of web application vulnerabilities.

REFERENCES

[1] Stefano Calzavara, Riccardo Focardi, Marco Squarcina, and Mauro Tempesta. Surviving the web: A journey into web session security. ACM Comput. Surv., 50(1):13:1–13:34, 2017.

[2] Avinash Sudhodanan, Roberto Carbone, Luca Compagna, Nicolas Dolgin, Alessandro Armando, and Umberto

Morelli. Large-scale analysis & detection of authentication cross-site request forgeries. In 2017 IEEE European Symposium on Security and Privacy, EuroS&P 2017, Paris, France, April 26-28, 2017, pages 350–365, 2017.

[3] Stefano Calzavara, Alvise Rabitti, Alessio Ragazzo, and Michele Bugliesi. Testing for integrity flaws in web sessions. In Computer Security - 24rd European Symposium on Research in Computer Security, ESORICS 2019, Luxembourg, Luxembourg, September 23-27, 2019, pages 606–624, 2019.

[4] OWASP. OWASP Testing Guide. https://www.owasp.org/index.php/ OWASP Testing Guide v4 Table of Contents, 2016.

[5] Jason Bau, Elie Bursztein, Divij Gupta, and John C. Mitchell. State of the art: Automated black-box web application vulnerability testing. In 31st IEEE Symposium on Security and Privacy, S&P 2010, 16-19 May 2010, Berkeley/Oakland, California, USA, pages 332–345, 2010.

[6] Adam Doup´e, Marco Cova, and Giovanni Vigna. Why johnny can't pentest: An analysis of black-box web vulnerability scanners. In Detection of Intrusions and Malware, and Vulnerability Assessment, 7th International Conference, DIMVA 2010, Bonn, Germany, July 8-9, 2010. Proceedings, pages 111–131, 2010.

[7] Adam Barth, Collin Jackson, and John C. Mitchell. Robust defenses for cross-site request forgery. In Proceedings of the 2008 ACM Conference on Computer and Communications Security, CCS 2008, Alexandria, Virginia, USA, October 27-31, 2008, pages 75–88, 2008.

[8] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. Foundations of Machine Learning. The MIT Press, 2012.

[9] Michael W. Kattan, Dennis A. Adams, and Michael S. Parks. A comparison of machine learning with human judgment. Journal of Management Information Systems, 9(4):37–57, March 1993.

[10] D. A. Ferrucci. Introduction to “This is Watson”. IBM Journal of Research and Development, 56(3):235–249, May 2012.

[11] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham,

Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587):484–489, Jan 2016.

[12] Michele Bugliesi, Stefano Calzavara, Riccardo Focardi, and Wilayat Khan. Cookiext: Patching the browser against session hijacking attacks. Journal of Computer Security, 23(4):509–537, 2015.

[13] Stefano Calzavara, Gabriele Tolomei, Andrea Casini, Michele Bugliesi, and Salvatore Orlando. A supervised learning approach to protect client authentication on the web. TWEB, 9(3):15:1–15:30, 2015.

[14] Stefano Calzavara, Mauro Conti, Riccardo Focardi, Alvise Rabitti, and Gabriele Tolomei. Mitch: A machine learning approach to the blackbox detection of CSRF vulnerabilities. In IEEE European Symposium on Security and Privacy, EuroS&P 2019, Stockholm, Sweden, June 17-19, 2019, pages 528–543, 2019.

[15] Giancarlo Pellegrino, Martin Johns, Simon Koch, Michael Backes, and Christian Rossow. Deemon: Detecting CSRF with dynamic analysis and property graphs. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017, pages 1757–1771, 2017

.