Institute of Advanced Engineering and Science

International Journal of Information & Network Security (IJINS)
Vol.2, No.2, April 2013, pp. 124~128
ISSN: 2089-3299

Creating enhanced Wikileaks

Amr Galal*, Hesham Hassan*
* Department of Computer Science, Faculty of Computers and information, Cairo University

Article Info

Article history:

Received Nov 5th, 2012
Accepted Dec 10th, 2012

Keyword:

Wikileaks
SOA
Security
Software engineering
Web services

ABSTRACT

In this paper we discussed how to create an enhanced version of Wikileaks using the web services and SOA concepts. We showed why we believe that Wikileaks using its announced methods and features was not too hard to be prevented; moreover we believe it is doable to spread any document on the internet as if it were a Wikileaks document. We showed how it was technically possible for the governments to prevent it if they want to do so, how Wikileaks was supposed to be happen using web services and how it could optimally be if public UDDIs exist.The paper suggesting how to make wikileaks alike harder to be prevented and at the same time by showing its weak points and how to overcome those points it could be considered as a step forward to prevent future wikileaks.

Corresponding Author:

Department of Computer Science,
Faculty of Computers and information, Cairo University.
Email: amrgalal@aucegypt.edu

1. INTRODUCTION

We believe governments were capable to prevent Wikileaks and we think applying some of the SOA concepts may enhance the Wikileaks and make it much harder to be prevented. Regardless the debate about the benefit or risks of the Wikileaks we concentrated only on its technicalities and how it was supposed to be. we discussed In this paper from technical point of view only what are the weak points of the Wikileaks that could be used to prevent it, what suspicious techniques used that make us believe it is not real, how it was easy to replace those techniques and finally we suggested two different methods to create "hard to prevent Wikileaks". The motivation of the paper is to discuss how the SOA concepts could be applied to reach a better solution for one of the hottest topics. The objective of the paper is neither to show how to prevent Wikileaks itself nor how to enhance it and make it "harder to be prevented", meanwhile it could be considered a step forward in both ways.

2. BACKGROUND

For so many reasons we believe Wikileaks has been permitted, we will neglect all non-technical reasons and list in this section two reasons that support our opinion.

2.1. governments had different methods to prevent it and did not use any of it:

Governments had at least three methods to prevent it. Optimally all methods can be combined together to accomplish a tighter solution.
The first method depend on the fact that the governments control the main DNS servers or at least able to force it to achieve its requirement so it can remove the DNS entry of the Wikileaks site from the DNS servers or redirect it to a dead page, that could be accomplished by forcing following the cyber-low or via DNS poisoning. DNS poisoning attack associate the victim domain name with a false IP address [1] and considered a trivial electronic warfare mission.

Another method to prevent it is through filtering its keywords from the search engines or through adding noise entries to over crowd the search results or even increase the weight of trivial pages over the important ones by creating a crafted traffic to the trivial ones.

It is also possible for the government to prevent it by controlling the core routers traffic to the wikileaks site either by either explicitly drop the traffic distained to it or by narrowing its bandwidth to unpractical limit. Another technique also which depend on controlling the routers is to use URL rewriting to redirect the traffic distained to the Wikileaks to other pages specially designed for that reason.

2.2. Practically it is illogical:

One of the very astonishing things in Wikileaks is the absence of files non-repudiation and file signing techniques accompanied with using the torrent files which is a format to identify a specific file using its features like the name of the file, extension and its size to enable exchange it between the user of the torrent viewer and agent. This technique furnish for announcing any file with any content at anytime just by creating the file with the specified features and put it on some machines to spread it all over the users of the torrent agent especially the wikileaks permitted the users to download the torrent files without any descriptions or even descriptive names. This combination permits any torrent user with enough knowledge to spread a crafted file with any information he wants as if it were wikileaks files.

Another point is the segregation of duties which is one of the very fundamental concepts in security and in the wikileaks case the size of the information forces this rule by nature since no one user or even one department need or should have or has the capacity to use such huge amount of information. Some other authors mentioned this huge size and how it is difficult to process [2].

Another point stems from the huge size of the information and its scatter over so many fields is the need to transfer, classify and filter it while time span for this is not enough. As per some authors 54,834,989 classified material exist [5] which practically impossible to be handled that easy.

3. RELATED WORK

Although the extraordinary fame of the wikileaks it seems that little academic researchers got interested in it, for example until the authoring of this paper Google scholar search engine brought only 437 entries in all scientific branches has the keyword "Wikileaks" in the domain of .edu and a vast majority of it is non technical. Only 273 entries remain if we removed the domain of mit.edu. Only 108 of it has the words abstract and conclusion. Finally only 3 of it mentioned the word SOA and only one of it truly only one of it mentioned the word intentionally [6] and it is a master degree in "mass communications and media arts".

On the other hand, so many researcher discussed web services and SOA related topics but according to our survey none of them merged it with wikileaks topic although some of them highlighted the fact that the web service may not honestly describe its real function [7] but no one managed to think about using this fact to build a time bomb model or to build a reveal secrete information in a certain time although the simplicity of applying the idea.

4. PROPOSED APPROACH

It is doable to make a real Wikileaks, in the following part we will show two techniques to do it. Moreover we also suggest how to control the speed of its spread and even how to selectively spread the information depending on the domain or geographic distribution of the audiiance.

4.1. Using public UDDIs:

Microsoft and many other large companies announced the closure of its public UDDI long time ago [3] because of security and some other reasons and did not mentioned what is meant by those security reasons or described "clearly" the details of those reasons, one obvious reason is that any developer may publish a web service that is not honestly do its documented task [7] but we believe there are so many other reasons one of it may be the ability to publish content on the net in a way that resists blocking. That could be done by creating a set of web services each of them return one of the site contents, which mean we can change the site contents by changing the called services or changing the values of services calling parameters. Since both names of the services and parameters values could be any non related things like large numbers or combination of characters it could be announced when the secrete information needed to be revealed by spreading the name of the service and the specific parameter value even without affecting any other application that currently use this web service to handle its normal previously announced task. A lot of methods could be used to spread the secrete revealing service and its specific parameter, those method including for example emails, blogs and surprisingly web services also. If public UDDIs were available and commonly used that would permit registering of the wikileaks contents in so many UDDIs without even those UDDIs to know it is part of the wikileaks plus the difficulty of removing those services after being components of some other applications. Moreover the concept of the cloaking or phantom page technology could be applied to create pages with components that only visible when a certain set of conditions come true [4], in such case all what is needed is to select a set of conditions that permit enough time to widely spread those components to large enough number of sites then the secrete information revealed on all those sites that include this component. Figure 1 shows a simple graphical representation of the above idea. Let us consider the enhanced wikileaks-alike model composed by the following main parties i) client (c), the entity that need to integrate remote service. ii) public Service Discovery (UDDI), a registry of services. iii) Service provider (sp), the entity implementing remote services accessed by c and need to publish secrete information. iv) Triggers host (th) the entity that hold information that enable starting of publishing secrete information.

A service provider sp stores the service interface of a service S in a public UDDI. When a client c send a search request to the public UDDI receive the service descriptor of the service S that could be used to call the service S from the service provider sp. when the service provider sp receives a call it check the triggers conditions in the trigger host th which send the triggers status to the service provider sp. depending on the triggers status the service provider sp provide the service S to the client with or without the secrete information.

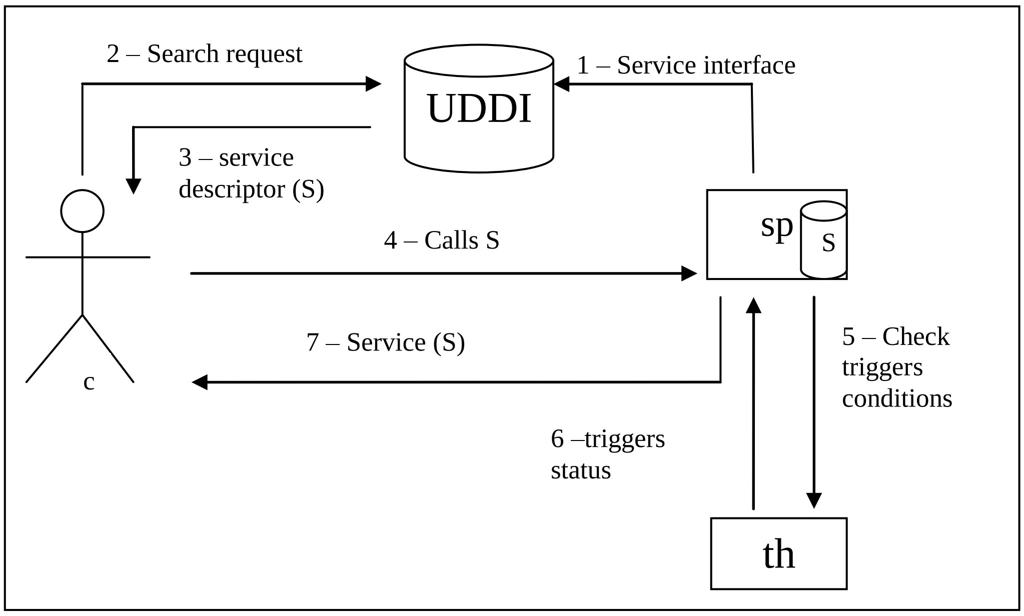

Figure 1: enhanced wikileaks alike model using public UDDIs

4.2. Using web services:

It is also possible to create a real wikileaks without public UDDI using the concepts of reusability. The main idea is to create large number of website frameworks, each of it need XML file to direct it to locations of files that could be included to assemble the final website; those locations could be on public file sharing sites or free hosting sites, and may be encrypted also, The XML could include also parameters to reform the website. If those website frameworks are commonly used adding alternate data streams, Steganography and phantom page technology may produce a perfect Wikileaks recipe. Of course the needed XML files could be published later via e-mail, blogs or any other mean. Figure 2 shows how to accomplish enhanced wikileaks alike model using web services and without public UDDIs.

The enhanced wikileaks-alike model composed by the following main parties. i) initiator of the SOA i, the entity that initiate the SOA and want to publish secrete information. ii) user of the application u, the entity that will surf the application created by c. iii) client c, the entity that will search the ps and download SOA and use it to create a. iv) public share ps, the entity that share SOA for public. v) application a, the entity that created by c using the downloaded SOA and surfed by u. vi) trigger host th, the entity that publicly share the status of the trigger.

An initiator i upload SOA to a public share ps. when a client c search the public share ps for an SOA with a specific feature and found it, he downloads it. Then he use it to create an application a using it. Frequently the application a check the trigger th host and receive trigger status from it. When a user u surf the application a the site contents showed depending on the trigger status.

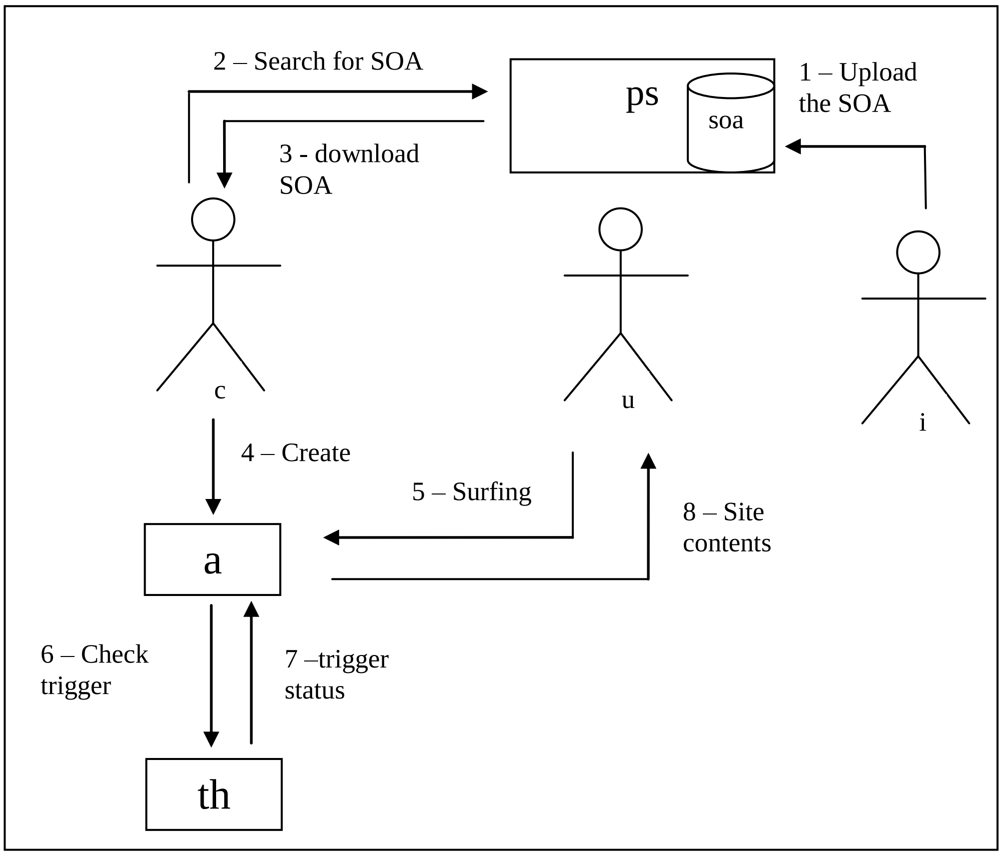

Figure 2: enhanced wikileaks alike model using web services.

Both suggested method can force the spread level of the secrete information that wanted to be revealed by the selection of the domain and commonality of the originally announced web service or framework function. Let us consider for example a web service that provides the local time or temperature of the weather or even a more commonly needed web service example could be a user login authentication service. A pretty good and easy to implement idea is to let the web service do its originally announced function beside doing the secrete function, for example a function that provide a local time may continue produce the local time string text but after concatenating with it another text that contain today's wanted to reveal secrete and optimally the local area code and the time could be used as selection criteria for that secrete.

5. CONCLUSION AND FUTURE WORK

Depending on all of the above we conclude 3 things:

- Governments permitted the wikileaks
- Wikileaks could be done using web services
- Wikileaks could be done in optimal way if we have public UDDIs
- Very rare academic researches discussed wikileaks

Future directions of this work will be:

- Injecting a handcrafted document to the torrent as a Wikileaks document with content that proof fake.
- To apply one of the proposed techniques to produce a Wikileaks alike model with enhanced feature like:
    - being harder to stop
    - Immunized against fake document injection.

REFERENCES

[1] D.Dagon, M.Antonakakis, K.day, X.Luo, CPLee, W.Lee: "Recursive DNS Architectures and Vulnerability Implications". In Proceeding of the 16th Annual Network and Distributed System Security Symposium (NDSS 2009), San Diego, CA, February 2009

[2] Roberts, Alasdair S.: "WikiLeaks: The Illusion of Transparency". International Review of Administrative Sciences, Suffolk University Law School Research Paper No. 11-19. , 2012

[3] Martin Treiber, Schahram Dustdar: "Active Web Service Registries". IEEE Internet Computing 11(5): 66-71 (2007)

[4] David Y. Wang, Stefan Savage, and Geoffrey M. Voelker: "Cloak and Dagger: Dynamics of Web Search Cloaking". Proceedings of the 18th ACM conference on Computer and communications security, New York, NY, USA 2011

[5] Maurer, Tim: “WikiLeaks 2010: A Glimpse of the Future?”. Discussion Paper 2011-10, Cambridge, Mass., Belfer Center for Science and International Affairs, Harvard Kennedy School, August 2011.

[6] Kravchuk, Olesya: "Change in Journalistic Practices in the Age of Global Networked Technologies". Master degree, department of Mass Communications and Media Arts In the Graduate School, Southern Illinois University Carbondale, May 2011, http://opensiuc.lib.siu.edu/gs_rp/83

[7] M. Paolucci, T. Kawmura, T. Payne, and K. Sycara.: "Semantic Matching of Web Services Capabilities". In The First International Semantic Web Conference, 2002.

BIBLIOGRAPHY OF AUTHORS

Amr Galal is an Egyptian researcher, Postgraduate diploma in computer science, from ISSR, Cairo University, Egypt in 1997. M.Sc. in computer science, from ISSR, Cairo university, Egypt, in 2005. PhD in computer science from Faculty of Computers and Information, Cairo University (until now). Amr has been published many research papers in scientific journals and Conferences. Amr interests are Information and network security, software engineering, reusability and Service Oriented Architecture (SOA).

Prof. Hesham A. Hassan is an Egyptian researcher born in Cairo in 1953. Hesham’s educational background is as follows: B.Sc in Agriculture, Cairo University, Egypt in 1975. Postgraduate diploma in computer science, from ISSR, Cairo University, Egypt in 1984. M.Sc in computer science, from ISSR, Cairo university, Egypt, in 1989. PhD in computer science from ISSR, Cairo University (dual supervision Sweden/Egypt) in 1995. He is now a PROFESSOR and HEAD of computer science department at the faculty of computers and Information, Cairo university. He is also IT Consultant at Central Laboratory of Agricultural Expert System, National Agricultural Research Center. He has published over than 51 research papers in international journals, and conference proceedings.
He has served member of steering committees and program committees of several national conferences. Hesham has supervised over 27 PhD and M. Sc theses. Prof. Hesham interests are Knowledge modeling, sharing and reuse, Intelligent information retrieval, Intelligent Tutoring systems, Software Engineering. Cloud Computing and Service Oriented Architecture (SOA)