Security analysis based on the test environment

Aragats Amirkhanyan
Hasso Plattner Institute
aragats.amirkhanyan@hpi.de

Research in the field of Information Security is always faced with the problem of lack of data or a suitable environment. This problem is not limited to research in the field of information security, but also for other areas. For research in the field of information security - is a big problem. Researchers have to find solutions for generate data, constructing a suitable environment, etc. A lot of affords to start the research. In the report, we consider examples of how we in the security group of the chair "Internet Technologies and Systems" solve similar problems. We consider the typical ways of solving them and consider a prototype project that we could help to simplify the implementation of research.

1. Introduction

Many research projects come from the industrial companies. It is seems that if the company is interested in success result of the research, the should provide all needed information, data, their environment. But they do not it. And obviously, the reason has a secure aspect. Most of researchers have to spend a big part of their time for preparing or generate data, deploy research environments and so on. The lack of the data is a big problem academic world. The preparation stage has quite big overhead of the research work, which could be overcome in some cases. It is not a secret that many researchers do the almost same researches which require the same data, the same environment. And every time they have to invent new bicycle.

In this report we mostly speak about the problems of the security researches. We mention certain problems which we impact during the security research and the way of solving the problems. The report include the overview the prototype application which the main goal is to simplify the preparation stage of the research and make possible to overcome the research overhead. Despite the fact that the report contains information about security research all of that could be used for solving problems in the research project of other areas.

2. Deploy test environment

Many security network and software analysis research start with preparing the test environment. The test environment is deployed in the most cases as the virtual network with configured hosts, routers, networks and all others resources including users account. The test environment could be deployed on the local computer by using the software for virtualization like the VirtualBox, VmWare Workstation and others or could be deployed on the remote server with installed hypervisor software like VMware vSphere Hypervisor (ESXi).

The common way to create the developing environment is to do it manually. You have to download image of operation system, install it on the Virtual Machine, configure it, install necessary softwares. The developing environment is usually more complex just one virtual machine, so you have to do the same step several. In some cases you can just clone virtual machine, but you still have to do a lot of manual work.

There are some software which can simplify the process of creating the developing/test environment. One of the them is Vagrant. Vagrant allows to create and configure lightweight, reproducible, and portable development environments. The configured environment could be reused. Vagrant project has the relative project which is called VagrantCloud. The project is hosted on the https://vagrantcloud.com. It is some kind of the catalog of prepared environments. The prepared developing environment is called box. People could share with community their boxes. Everyone could find the appropriate environment and reuse it instead of configuring new one. Vagrant allows to deploy the environment into the local VirtualBox, VMware Workstation, Aamazon Web Service (AWS). It means researchers are quite free in using the platform. The features of the Vagrant are not bound only by running the environment. The application a lot of give capabilities including synchronizing between the host and guest machine, integration with Chef[*], Puppet[*] and other. Learn more on the official website http://www.vagrantup.com/

The process of creating the development/test environment can by simplify by cloud providers. Many cloud providers provide capability easy to install and run virtual machine with any operation systems, configure the network and many other features. The most popular is Amazon Web Service (AWS). AWS is Infrastructure-as-a-Service. AWS provides huge amount of service which could solve any problem. You could combine any infrastructure service to create the certain development environment. I mentioned just AWS, but the market of cloud providers grows extremely so it is possible find any other. As Vagrant AWS also provide capability to save configuration of the development environment, distribute it and reuse.

But we are talking about the researches with security aspects. It means that results and the process of the research in the most cases must secret. In this case it is not possible to use public cloud providers. Here opensource communities come to help us. There si not sense to tell that opensource communities grow extremely. A lot of big IT companies invest into the opensource projects. Many of these opensource projects are widely spread. They are use everywhere. So it does not go past the cloud technologies. There several opensource cloud platform which could be used for creating own IaaS, PaaS and other. Some of them: OpenStack, OpenNebula, OpenShift Origin and so on. So clouds become private. For us it means that we could create our test/development environments by using flexibility and capability of cloud services. But still there are some problems and overhead of creating the test/development environments. It could be solved by using the PaaS, but PaaS's are usually designed for specific task mostly for running the infrastructure for web applications in Java, PHP, Ruby and so on. No one does provide flexible platform as a service based on all functionality of infrastructure as a service. We would like to call it Platform as a Infrastructure (PaaI). Jeslastic[*] uses this definition for describing its service, but they include other meaning into this definition.

3. Simulation of users behaviors

There is task to analyses the behavior of users to predict attacks or abnormal behavior. There are some algorithms for Attack Prediction based on Machine Learning [http://www.csjournals.com/IJCSC/PDF1-2/51..pdf]. The algorithms could be used for research purposes, in this task the big problem is lack of data. The customer do not want and can not provide Active Directory log files which contains information about user activities. We have to start research since writing the script for simulation user behaviors.

The Scenario in our case is the description of the network infrastructure, information about users, and scenario of their behavior. To analysis user behaviors by predict algorithms we must have normal scenario and abnormal scenario.

3.1. Environment description

Description of a network:

- 4 computers.
- Domain controller.
- Wiki Server.
- DB server.

Users:

- Petrov
- Ivanov
- Smirnov
- Admin

3.2. Normal scenario

Figure 1. Usually ordinal users login to their computers. They do it several times per day. All users and admin have an access to Wiki, but Ivanov usually do not use it. Admin usually login to Domain Controller. Other users do not do it, because their do not have an access. Ivanov does not login to wiki, but he has an access. Others users do it. Ivanov sometimes login to DB Server. Others users do not do it, but they have an access.

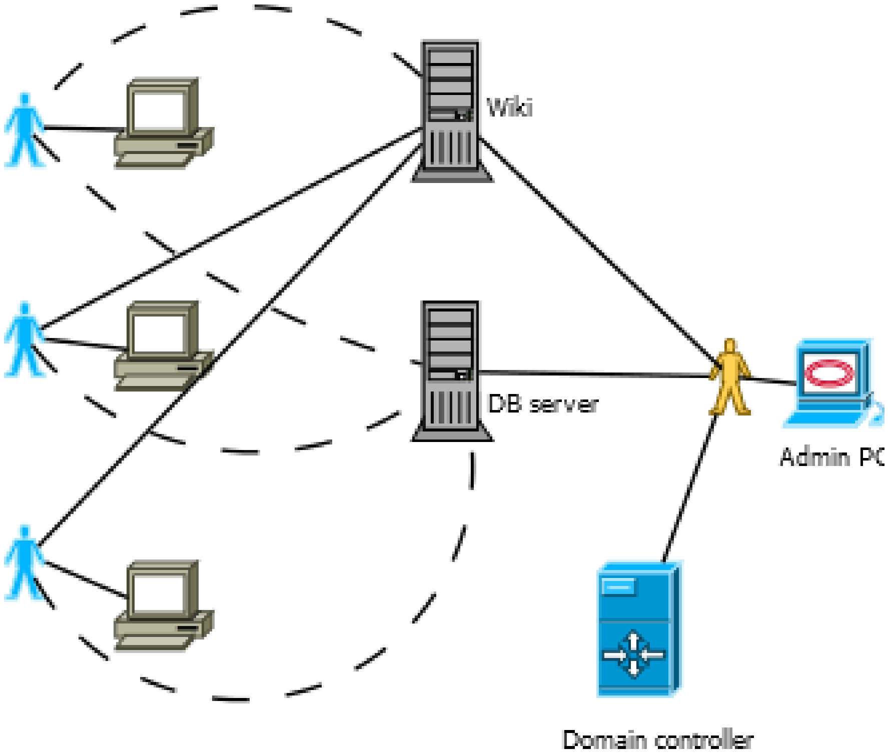

Figure 1: *Normal scenario*

3.3. Abnormal scenario

Figure 2. Ivanov logged to wiki. He usually does not do it, but he has an access and other users do it every day. Petrov logged to DB server. He has an access, but according to the normal behavior only Ivanov uses the DB server. So it could be abnormal behavior.

3.4. Implementation

The network is deployed and configured manually on the FutureSOC server with WMware ESXi. The script for the simulation users activities is written in Python with using additional libraries for connecting to Virtual Machines by VNC and

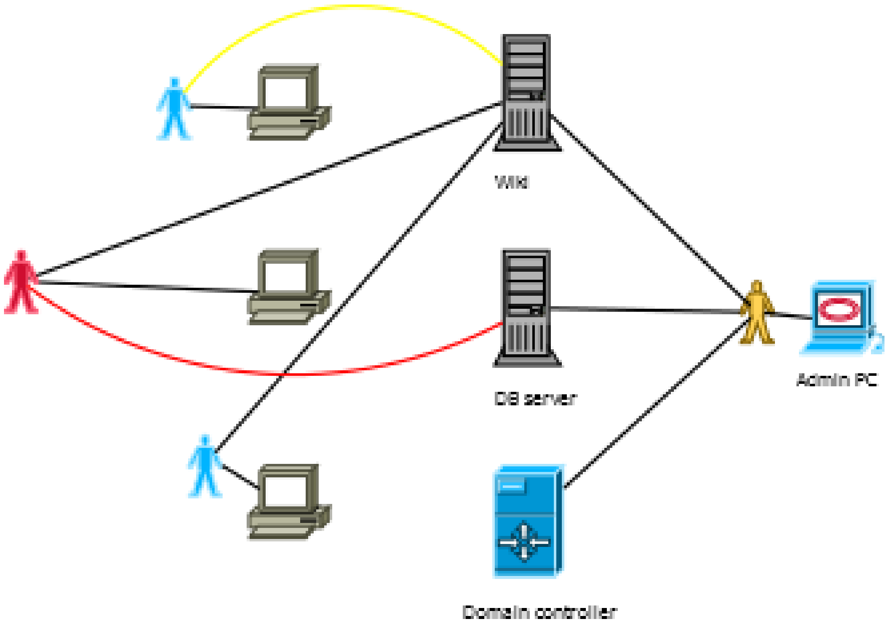

Figure 2: *Abnormal scenario*

making screenshots. There are 3 csv files for describing the whole scenario. The first of them (computers.csv) describe the the all computers which participate in the scenario. It contains the information about computer identifier, IP, port, VNC password and type of operation system(OS). The second (scenarios.csv) describes the main user activity. The main user activity means the connection to the local computer. The files contains computer identifier to connect, username, password, session time, count of sessions and identifier of inner scenario. The third (inner-scenarios.csv) describes the user activity after login to the local computer. For example, it could describe the connection to the wiki, the DB server or Domain Controller or even connection to another user computer. There are two sets of csv files. The first set is used to perform normal scenario and the second is used to perform abnormal scenario accordingly. Normal scenario takes about 3,5 hours and abnormal scenario takes about 5,5 hours.

4. Automatization of research

4.1. Motivation

As we can see from previous sections there is quite big overhead of research in preparing the research environment and performing the basic scenarios. Many researchers usually do common things, scenarios, but for different purpose. Everytime they have to invent new bicycle and in most cases it is just a waste of time. It would be nice to have some tool for overcoming this overhead, to automatize the common research processes and make research easier. We suggest the concept of the project consists of four independent projects. Each project could be used independent and will be useful for researches, users, administrators, developers. Project could be used together or in any combination to solve different research problems. We call the project Security Lab Generator (SLG). It contains four projects (systems):

- Network Designer Service (ND)
- Provision Language Translator (PLT)
- Provision System (PS)
- Network Security Analyzer (NSA)

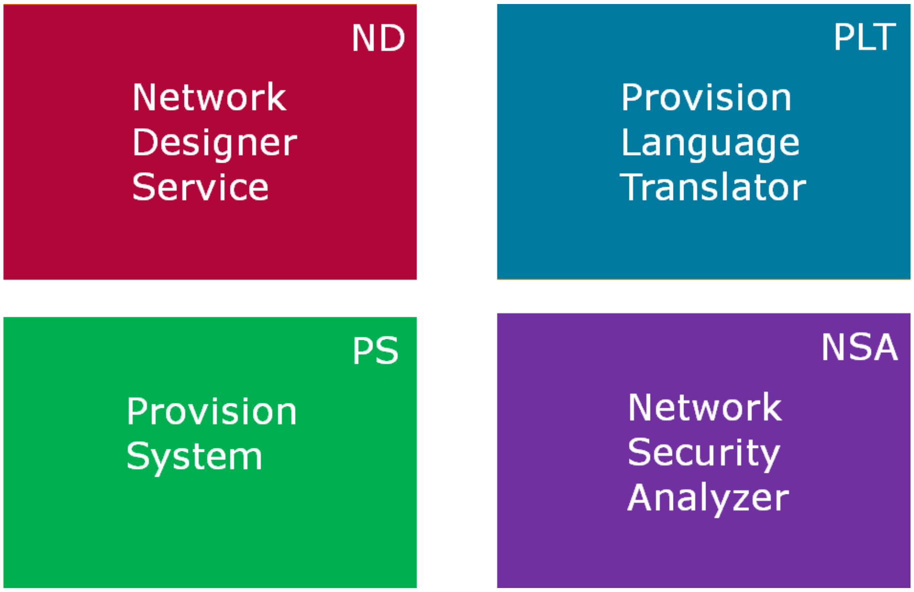

Figure 3: *SLG's systems*

4.2. Network Designer Service

The purpose of the Network Designer Service is to provide easy way to design the network including specifying networks, hosts, connections, softwares, user accounts. The system should provide the flexibility providing by IaaS and simplicity providing by PaaS. It the first part of the big ecosystem. The system should provide the capability to export designed network as structured data. For example XML, JSON or other. It could be used independently by system administrator to design the network and collaboration. And also it could be used in combination with others system to perform the whole life cycle flow. As we mentioned in the previous subsection we want to create project for automatization of research. So it is the first step to reach goal.

Research areas:

4.3. Provision Language Translator

4.4. Provision System

4.5. Network Security Analyzer

5. Methods

Maecenas sed ultricies felis. Sed imperdiet dictum arcu a egestas.

- Donec dolor arcu, rutrum id molestie in, viverra sed diam
- Curabitur feugiat
- turpis sed auctor facilisis
- arcu eros accumsan lorem, at posuere mi diam sit amet tortor
- Fusce fermentum, mi sit amet euismod rutrum
- sem lorem molestie diam, iaculis aliquet sapien tortor non nisi
- Pellentesque bibendum pretium aliquet

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

6. Results

Table 1: *Example table*

Name		
First name	Last Name	Grade
John	Doe	7.5
Richard	Miles	2

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

$$e = mc^2 \tag{1}$$

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

7. Discussion

7.1. Subsection One

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

7.2. Subsection Two

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

References

[1] I. Mizuuchi, R. Tajima, T. Yoshikai, D. Sato, K. Nagashima, M. Inaba, Y. Kuniyoshi, and H. Inoue, “The design and control of the flexible spine of a fully tendon-driven humanoid “Kenta",” in *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, Lausanne, Switzerland, 2002, pp. 2527–2532.

[2] I. Mizuuchi, H. Waita, Y. Nakanishi, T. Yoshikai, M. Inaha, and H. Inoue, "A musculo-skeletal robot leg capable of adding or rearranging the muscles," in *21th Annual Conference of the Robotics Society of Japan*. Robotics Society of Japan, Tokyo, Japan, 2003, presentation number: 1C29.

[3] I. Mizuuchi, T. Yoshiaki, Y. Nakanishi, and M. Inaba, "A reinforceable-muscle flexible-spine humanoid "Kenji"," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2005, pp. 692–697.

[4] I. Mizuuchi, Y. Nakanishi, Y. Sodeyama, Y. Namiki, T. Nishino, N. Muramatsu, J. Urata, K. Hongo, T. Yoshikai, and M. Inaba, "An Advanced Musculoskeletal Humanoid Kojiro," in *Proceedings of the 2007 IEEE-RAS International Conference on Humanoid Robotics*, 2007, pp. 101–106.

[5] Y. Nakanishi, Y. Namiki, K. Hongo, J. Urata, I. Mizuuchi, and M. Inaba, "Design of the musculoskeletal trunk and realization of powerful motions using spines," in *Proceedings of the 2007 IEEE-RAS International Conference on Humanoid Robotics*, 2007, http://planning.cs.cmu.edu/humanoids07/p/85.pdf.

[6] C. Ott, O. Eiberger, W. Friedl, B. Bauml, U. Hillenbrand, C. Borst, A. Albu-Schaffer, B. Brunner, H. Hischmuller, S. Kielhofer, R. Konietschke, M. Suppa, T. Wimbock, F. Zacharias, and G. Hirzinger, "A humanoid two-arm system for dexterous manipulation," in *Proceedings of the 2006 IEEE-RAS International Conference on Humanoid Robotics*, 2006, pp. 276–283.

[7] J. Or, "A control system for a flexible spine belly dancing humanoid," *Artificial Life*, vol. 12, no. 1, pp. 63–87, 2006.

[8] J. Or and A. Takanishi, "From lamprey to humanoid: The design and control of a flexible spine belly dancing humanoid robot with inspiration from biology," *International Journal of Humanoid Robotics*, pp. 81–104, 2005.

[9] ——, "The effect of an emotional belly dancing robot on human perceptions." *International Journal of Humanoid Robotics*, vol. 4, no. 1, pp. 21–48, 2007.

[10] J. Or, "The development of emotional flexible spine humanoid robots," in *Affective Computing, Emotion Expression, Synthesis and Recognition*, J. Or, Ed. Advanced Robotics Systems, 2008.