Towards a Unified Data Storage and Generic Visualizations in Cyber Ranges

Radek Ošlejšek, Dalibor Toth, Zdenek Eichler, Karolína Burská
Masaryk University, Faculty of Informatics, Brno, Czech Republic
oslejsek@mail.muni.cz
totik@mail.muni.cz
zdenek.eichler@mail.muni.cz
396296@mail.muni.cz

Abstract: Cyber ranges enable security experts to emulate computer networks where they can perform cyber security exercises and experiments. Although their architecture may differ, the following four types of services are usually provided: (a) *Resource management*, i.e. the allocation of network infrastructure with requested topology and running applications. (b) *Interaction of users with hosts*, i.e. allowing users to log into hosts and run applications in the same way they interact in real computer networks. (c) *Monitoring services*, when network activities are monitored on the fly and measured data are stored and mediated to end users on demand. (d) *Learning and understanding* of cyber security processes by providing users with a continuous overview of events and developments by means of analytic tools, interactive visualizations, and other enhanced user interfaces. This paper deals with (c) and (d) service types. The low-level infrastructure of modern cyber ranges is usually generic, enabling to instantiate topologies and hosts of many types and then to support end user with variable objectives and tasks. However, this diversity of objectives put big demands on the design of a monitoring subsystem and corresponding user interfaces providing insight into the measured data and cyber security processes. It is because the monitored data and their manipulation strategies may differ scenario to scenario. This paper discusses a generic approach to data storage using traditional entity-relationship databases. Our approach exploits data analysis patterns enabling us to define scenario-specific security phenomena without modifying rational scheme or other parts of the cyber range architecture. A flexible web-based system of user interfaces adaptable to the variable data is discussed as well. Our approach has been developed as part of a KYPO Cyber Range. Its usability has been evaluated within many diverse successfully performed cyber exercises. This paper presents several types of exercises that differ in required phenomena and interactions and then sufficiently demonstrate generality and usability of our approach.

Keywords: kypo, cyber range, testbed, phishing scenario, monitoring, visualization.

1. Introduction and background

In this paper, we aim to support learning and understanding capabilities of cyber ranges. To be successful in this endeavor, we have to rise to the following major challenges.

Monitoring of complex and aggregated data: Learning capabilities depend on complex and aggregated data. However, cyber ranges do not know in advance what data to be monitored for particular scenario or how to aggregate them. Current cyber ranges, therefore, natively measure only a common phenomena like utilization of links or CPU load and provides these data as a standard service. More complex or aggregated data and events, like availability of remote services, are usually delegated to external monitoring systems that have to be configured and deployed into the cyber range infrastructure by end users. However, end users are never permitted to access a virtualization layer or a low-level technical infrastructure realizing the simulation of a computer network and, therefore, their possibilities are very restricted. Moreover, the preparation is very time-consuming, annoying and requires deep knowledge of the low-level cyber range infrastructure. Therefore, our goal is to integrate monitoring of complex and aggregated data into cyber ranges and offer them out-of-the-box as a standard service for end users.

Adaptable user interfaces: Learning activities cover a wide range of particular user interests, from situation awareness, where users try to keep track on the infrastructure state, to forensic analysis. These interests can be supported by various specialized applications installed into the network by end users and used as necessary. However, this approach brings the same drawbacks we already mentioned in connection with manually installed data monitoring: it is laborious and hardly reusable. On the other hand, availability of a standard unified system for data storage would allow us to design advanced interactive visualizations built upon the data and provide them to end users as a part of a cyber range. However, we still face the problem of

huge diversity of users and their objectives. Even if two users are interested in the same data like network traffic, for instance, one would aim to analyze utilization of links while the other one would train defense tactics against DDoS attacks. Both the interests require completely different view on the same data, different user interfaces, and interaction tactics. Therefore, user interfaces have to be defined at user-level taking into account specific needs of particular users and their interests.

1.1 Related work

There exist a number of cyber security related testbeds world-wide. They differ in their application domains, hardware and software requirements, and facilities. Leblanc et al (2011) provide state of the art in the simulation and modeling of cyber attacks and defensive responses to those. They discuss a number of concrete simulations of cyber warfare including live, virtual, and constructive simulations.

A comprehensive overview of recent cyber ranges and testbeds can be found in Davis et al (2013). The survey mentions more than 30 cyber ranges and testbeds. We focus only on a few selected representatives that support user-defined cyber experiments. We describe those solutions that are intended to provide users with some broader interaction-related abilities. Mainly we consider the ability to measure various types of data during the experiment, the means of their visualization, and the range of user interactions with the data.

One of such testbeds is publicly available *Emulab/Netbed* (Garcia, 2012), which provides users with a flexible environment as it can automatically and dynamically map physical components (e.g. servers, switches) to a virtual topology. According to Siaterlis et al (2013), Emulab offers only limited functionality beyond link-tracing, i.e. packet capture of network traffic, and also it lacks support for measuring individual node metrics, e.g. CPU utilization. The only way, how to integrate a generic user-friendly data monitoring into a cyber experiment is to deploy software probes and data collectors as part of the experiment, as discussed in Jonhson (2008) or Siaterlis (2011), for instance. According to White et al (2002), it also lacks any user-friendly monitoring GUI and uses tools and open source applications for network traffic generating and capturing or for wide-area links emulation. Burtsev et al (2009) add that for various events generation the Emulab uses a simple command-line tool Event System.

Another relevant project is *DETER* and its facility *DETERLab* (Wroclawski et al, 2016), which employs the above-mentioned Emulab infrastructure solution. DETERLab contains a low-level substrate (MAGI) that provides the infrastructure needed to support workflow tools that instrument and control an experiment. MAGI also provides two further base-level tools that allow deploying, configuring, and visualizing the experiment. The bootstrap tool is used to deploy, install, and configure the software libraries and the messaging substrate on experiment nodes. It is typically run once when resources are first allocated to an experiment. A graph tool is used to collect basic measurements from the sensor agents and display the current status on the experiment for real-time and offline analysis.

In a *CyberVAN* testbed (CyberVAN, 2017), the background traffic is based on nfdump traces from the provider's network, TCP, UDP, and ICMP flows. According to Chiang et al (2013), it enables performing the post-mortem analysis for two possible targets - logs recorded throughout the simulations and VM images at the end of the simulations. One of the abilities of this testbed that makes use of hybrid emulation with a simulated network is the ability to dynamically reconfigure the simulated network and the host nodes. In addition to reconfiguration, there is also a need to be able to monitor the elements of the experiment for situational awareness. CyberVAN uses Big data analytics engine and techniques, for example, the tools in the Hadoop family and Map-Reduce paradigm.

SimSpace Cyber Range (Rossey, 2015) offers an ability to customize and extend the existing pre-defined networks or generate an entire network in a virtual environment. It contains an integrated User Emulation system that emulates the behaviour of enterprise users via artificial software agents. Network activities can be monitored at network and scenario level. The cyber range is controlled via a web portal that also provides access to results of analysis and assessment of monitored activities within the cyber range.

National Cyber Range, NCR, is a military facility enabling representation of operational networks and interconnection with military command and control systems with the ability to restore to a known checkpoint

baseline to repeat the test with different variables (Ferguson et al, 2014). The NCR is instrumented with traffic generators and sensors collecting network traffic and data from local and distributed nodes.

2. Use case: Phishing scenario

To demonstrate abilities and purposes of cyber ranges and mainly to explain various aspects of our approach, we formulate a concrete security scenario, which is in the further text referred as *Phishing Scenario*. This scenario was adapted from a realized exercise and was adjusted in order to capture various features of data measuring and interactive visualizations, as discussed in what follows.

The main goal of phishing attacks is to compromise users accounts or an IT infrastructure by fraudulently luring access data or by persuading users to install a harmful software. There exists plenty of tactics and technical ways of executing this kind of attacks but the most popular are those using e-mail communication. To make a phishing scenario believable and useful for of participants, it is necessary to put phishing e-mails into some broader story so that the participants do not know about the phishing test. Our phishing exercise is therefore covered by the attempt to compromise a web server using known vulnerabilities:

1. At the beginning, participants of the exercise know nothing about the network they are dealing with. A supervisor of the exercise informs participants that they should log into the dedicated computer, open an e-mail client and follow instructions available in the inbox.
2. Participants receive new e-mails with new instructions whenever they successfully complete the previous task. The tasks are:
    a. Find devices connected to the network by scanning ports. There should be one database server, one web server, and a few other devices connected to the network.
    b. Use SQL injection vulnerability to compromise and take over the database server.
    c. Run DoS attack from the database server to the web server by executing software already installed on the database server.
3. Harmless e-mails with instructions 2(a) - 2(c) are interspersed with forged phishing e-mails on variable level of plausibility, from naive machine-generated texts to messages masking themselves as instructions from the supervisor but demanding something strange or dangerous like installing software from unusual URL or providing private information by logging into a suspicious website.
4. Winners are those who successfully complete tasks 2(a) - 2(c) without being tricked by phishing e-mails.

3. Unified schema for data storage

To avoid the necessity to adapt a database scheme whenever a scenario with new data requirements is invented, we employ the *Observation* software pattern from the collection of Fowler's analysis patterns (Fowler et al, 1997) which splits database tables into two logical levels, as shown in the entity-relationship diagram in Figure 2. Obtained generic scheme enables us to store arbitrary raw and derived cyber security data, as discussed in what follows.

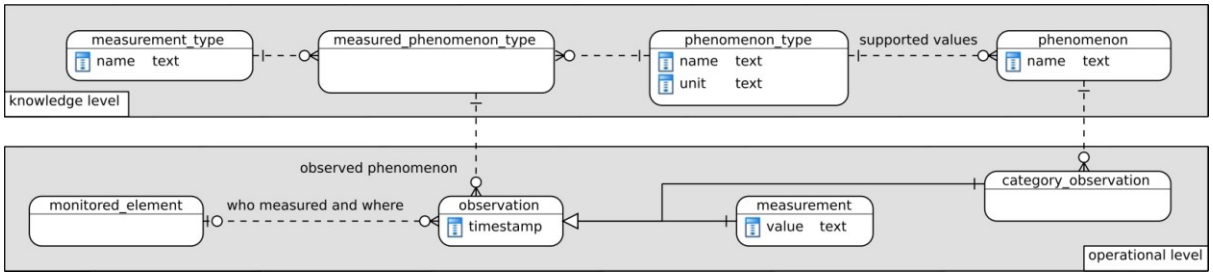

Figure 2: Data scheme for storing cyber security phenomena

3.1 Knowledge level

Tables in the knowledge level cope with the preparation phase of a new cyber exercise. They determine what is to be measured. The *phenomenon_type* table defines basic phenomena that occur in computer networks (and are measurable) like link capacity, link response time, CPU load or network protocol. These phenomena are of two types, (a) with unlimited domain, i.e. numerical values, IP addresses, etc., and (b) with enumerated

domain, e.g. network protocols. Enumerated values are defined in the *phenomenon* table and linked with the corresponding *phenomenon_type*.

Security experts often work with derived characteristics that either aggregate raw low-level data or combine several phenomenon types together. A typical example is average utilization of links in 5 minutes intervals. Such derived characteristics are defined in the *measurement_type* table, which is connected to one or more corresponding phenomenon types via the *measured_phenomenon_type* link table.

Table 1: Knowledge level for the Phishing Scenario

phenomenon	**phenomenon_type**	**measurement_type**
visible	visibility	host property
invisible	visibility	host property
attacker	role	host property
victim	role	host property
	traffic in bits	2 min average traffic
	new incoming e-mail	e-mail
	processed instruction e-mail	e-mail
	ignored phishing e-mail	e-mail
	successful phishing	e-mail

In the *Phishing Scenario*, the knowledge level consists of three types of information, shown as three line groups in Table 1: host properties, line properties, and e-mail delivery.

Host properties - the first goal of participants in step 2(a) is to scan the network in order to find out connected devices and mainly the vulnerable database server. Therefore, it is necessary to somehow distinguish between visible and currently invisible network nodes. To simulate such behavior we define "*host property*" measurement type connected to "*visibility*" phenomenon type with enumerated values "*visible*" and *invisible*". In the same way we can easily extend host properties with other phenomenon types, for example "*logical role*" with enumerated values "*attacker*" and "*victim*", and use them to visually emphasize nodes in the topology according to their roles in the cyber attack and to clarify the overview of the virtual battlefield, as discussed in Section 4.

Link properties – to capture the utilization of links during the DoS attack in step 2(c) we define "*traffic in bits*" phenomenon type and corresponding "*2 min average traffic*" measurement type to store 2-minute average utilization in b/s.

E-mails – in the *Phishing Scenario*, they represent the most difficult data because it is necessary to distinguish between e-mail types (ordinary e-mail with instructions vs. phishing e-mail) and their states (incoming, opened, processed, ignored, etc.). To keep the example simple, we restrict ourselves to only single e-mail type with four states:

- *new incoming e-mail* – applicable to both phishing and instruction e-mails,
- *processed instruction e-mail* – successfully fulfilled instruction,
- *ignored phishing e-mail* – phishing e-mail ignored or deleted by the user,
- *successful phishing* – the user has been tricked.

Our database scheme defines *e-mail* measurement type linked to the four aforementioned phenomenon types (e-mail states). However, this classification is not the only one possible. Our database scheme provides the freedom in the distribution of information among phenomenon and measurement types and then, for example, we could alternatively define *instruction e-mail*, *naive phishing e-mail,* and *advanced phishing e-mail* measurement types with shared *new*, *opened*, and *deleted* phenomenon types.

3.2 Operational level

Tables in operational level are used by data probes during the exercise to store measured values. Measured data are primarily stored in the *observation* table. Every record contains measurement time (the *timestamp* attribute), place of the measurement (the *who measured and where* relation), and what characteristics have

been measured (the *observed phenomenon* relation). The measured value itself is stored either in the *measurement* table in the case of the unlimited domain or in the *category_observation* table by linking the record with one of the predefined enumerated phenomena.

In the *Phishing Scenario*, a new incoming e-mail, for instance, is stored in the *measurement* and *observation* tables (the *value* attribute is set to e-mail ID, the *timestamp* is set to the time of the detection) and linked to the *measured_phenomenon_type* which connects *e-mail* measurement type with *new incoming e-mail* phenomenon type.

4. Adaptable user interfaces

The unified data storage brings the opportunity to support variable security scenarios with data defined at the user-level. However, if the user-defined data are to be mediated to end users then the front-end applications have to be highly configurable and adaptable, supporting the creation of GUI arrangement on demand.

Enterprise web portals, as defined in JSR 168 and JSR 286 Java portlet specification documents, bring a possible solution. They are designed to aggregate and personalize information through application-specific modules, so-called portlets. A portlet is a cross-platform pluggable software component that visually appears as a single window located on a web page. They can provide interactive forms and visualizations, graphs, tables, information banners and other views on data and processes. Once created, a portlet can be reused in many security scenarios.

Portlets are grouped together into so called *Sites Templates*. They handle the layout of web pages, configuration settings of portlets, and also inter-portlet communication and synchronization. We utilize Site Templates to enable end users to interactively create scenario-specific GUIs as self-contained pre-configured web pages composed of mutually cooperating specialized portlets adapted to scenario-specific data.

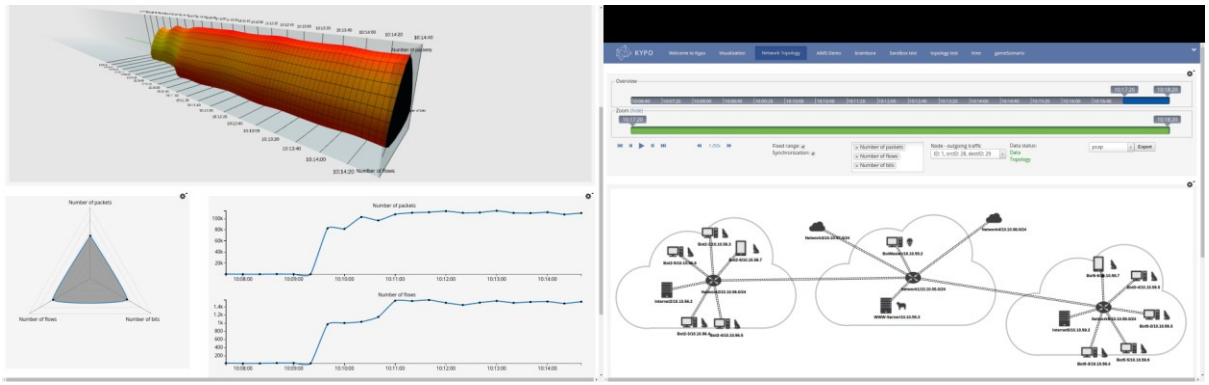

Figure 3: Screen shot of a web interface during the hands-on training (dual display mode)

So far, we have developed many specialized interactive portlets providing variable views on stored data in a unified manner. Their combination supports users in their activities and provides them with an insight into the cyber processes. As an example, we can introduce a portlet enabling to manipulate, filter and keep track on arbitrary dynamic temporal data, or analytics graphs that are useful for the analysis of aggregated phenomena, as discussed in (Eichler et al., 2015). An example of complex layout with interconnected portlets is shown in Figure 3. There is not enough space to describe all of them and then we rather present only one selected portlet that well demonstrates the utilization of the generic storage approach.

4.1 Network topology example

Topological view is a key visualization of our cyber range. Versatility was one of the key requirements on this visualization since the network topology is present in all scenarios but phenomena and other cyber security characteristics may significantly vary among scenarios. We demonstrate the mapping of visual elements to the measured phenomena of the *Phishing Scenario*.

At the beginning (task 1 of the scenario), participants are provided with very basic overview of the network, as shown in Figure 4. The visualization utilizes the "*visibility*" phenomenon type to show only a part of the network. "*Role*" phenomenon types are shown as symbols next to the node icon. The computer is intended to be used by the participant for the attack and then the computer has the logical role set to "*attacker*" and visualized by a skull symbol. E-mail events are displayed above network nodes, providing a direct overview of e-mail types and states (coded into colors). The black circle indicates one unread e-mail.

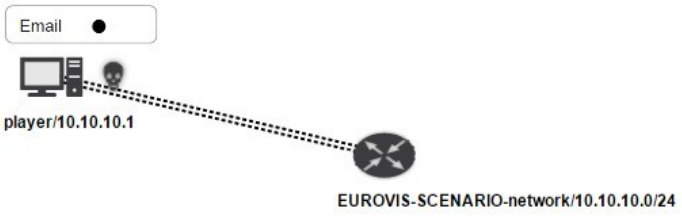

Figure 4: Phishing Scenario – phase 1

Users can connect to the computer via VNC by clicking on the computer icon and selecting access mode from the menu. VNC connection is opened as a new tab in the web browser, displaying the device screen.

When probes in the cyber range infrastructure detect port scanning (task 2(a) of the *Phishing Scenario*) they change the *"visibility"* phenomenon type of invisible nodes and the topology visualization reveals the whole network, as shown in Figure 5. Two servers in the revealed network are marked with the sheep symbol representing victims – targets of the attacker. Devices without logical roles have no special meaning in the scenario and they are there only to make some common traffic in the network. The event bar above the player's computer has more events recorded now – the blue ones refer to fulfilled tasks, the green one stands for a simple phishing attack, which was not successful, and the black one is pending unread e-mail.

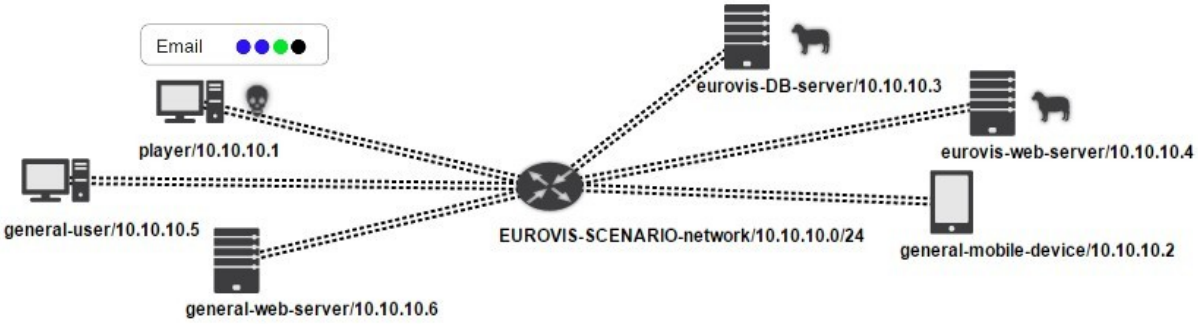

Figure 5: Phishing Scenario – phase 2

Figure 6 captures the last phase of the scenario. The player got control over the DB server (its role changed from sheep to wolf/bot) and ran a DoS attack (task 2(c) of the scenario). A red e-mail indicates that the player had been tricked by a phishing in the meantime. Double-dashed lines between the DB server to the web server are colored and animated to visualize high traffic measured on these links. The color is computed from "*2 min cumulative traffic*" values.

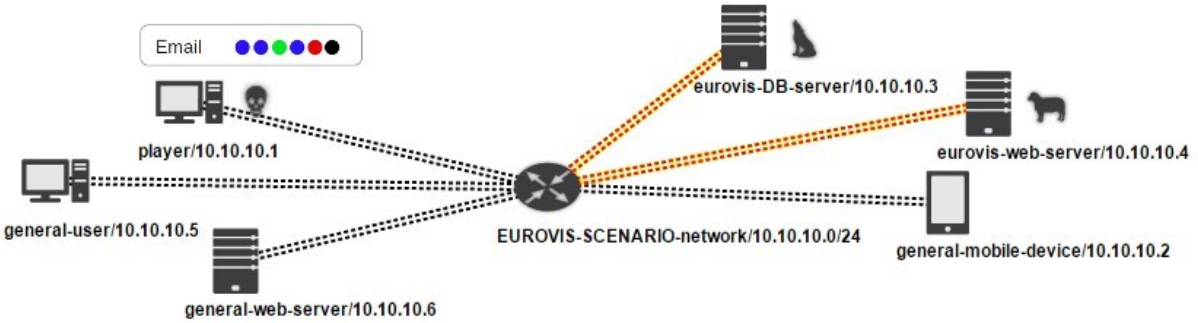

Figure 6: Phishing Scenario – phase 3

5. Evaluation and discussion

Our approach to data storage and visualization was developed and tested within a *KYPO Cyber Ranger* project (Čeleda et al, 2015). Sandboxes are built on top of a cloud managed by OpenNebula (Milojicic et al, 2011). Advantages of using cloud virtualization and other technical details about the implementation of sandboxes can be found in (Kouril et al, 2014). Activities within a sandbox are monitored by software probes (Hofstede et al., 2014; Velan et al., 2012) which gather monitored data on the fly and store them in a PostreSQL database using the scheme as discussed in this paper. Access to sandboxes is mediated via Liferay portal – an open source web portal implementation (Sezov et al, 2012). Portlets are implemented in various technologies, most frequently employing Javascript, D3.js library, SVG, and WebGL.

The main contribution of this paper is the ability to support heterogeneous scenarios as a service, i.e. by providing environment "out-of-the-box" without the necessity of manually modifying or adjusting infrastructure or architecture. To prove that this goal was achieved, we describe concrete practical demonstrations, courses and cyber security drills that were organized in KYPO Cyber Range. These events cover a wide range of different security scenarios. Besides the following large and significant events, we also organized many other smaller exercises and demonstrations so far.

5.1 Attack demonstrations

At the Network Operations and Management Symposium, NOMS 2014, we demonstrated a *Distributed Denial of Service* (DDoS) attack. Data measured in the cyber range included flows and relative and absolute traffic on links (average traffic in 5 mins intervals). The topology visualization was used to present impact of DDoS attack and counter-actions of admins on the network by coloring and animating links according to the current traffic. Moreover, analytics graphs were used for detail exploration of links load.

At the Future Crises Workshop in 2014, we organized a simplified version of the *Phishing Scenario* that was used in this paper. Participants were asked to recognize suspicious e-mails. If they failed, they were able to see the impact of the attack on the network visualization because their computer becomes part of a botnet and participated in a DDoS attack on a company server. Monitored data included flows, traffic on links and e-mail states similar to those discussed in this paper.

5.2 Hacking games

At the Autonomous Infrastructure, Management and Security workshop, AIMS 2014, we organized a so-called *capture-the-flag game*. In this scenario, one group of players was instructed to apply several simple attacks inside the network while the aim of another group was to defend the infrastructure. Similar scenario with different topology, participants, and types of attacks was repeated at the 44th TF-CSIRT meeting workshop in 2015. This workshop was organized for Computer Security Incident Response Teams from various parts of the world and then the participants represented very skilled experts. In these exercises, we monitored the availability of services and host states. Topology visualization was used for remote access to hosts, revealing topology step by step and providing an overview of roles and states of network nodes.

5.3 Cyber Czech exercises

Cyber Czech is the largest regular cyber security drill at Czech republic organized in the cooperation with Czech National Security Authority. The purpose of this technical two-day hands-on exercises is to simulate attacks on big critical infrastructures. Participants, about 40 in total, are divided into several teams:
White and green teams are supporting teams. They consist of referees, organizers, and storytelling members, whose purpose is to simulate real environment (e.g. simulation of communication with involved organizations and agencies).
Four blue teams, each of 4-5 participants, represent a primary target group of the drill. Their aim is to protect a critical infrastructure consisting of about 25 computers connected in complex topology and running various operating systems and services. Every blue team has its own sandbox. They operate simultaneously trying to be more successful that rivals.
One red team consists of attackers. This team shares sandboxes with all blue teams and attacks them simultaneously via vulnerabilities prescribed by the scenario.

Preparation of a new Cyber Czech takes roughly one year of extensive elaboration of a particular scenario. The most difficult part of the preparation is the composition and configuration of user interfaces because the exercise is based on very complex collaboration schemes where red and blue teams shared sandboxes and all teams have their individual views on current state of the KYPO infrastructure. From data storage point of view, every scenario requires its specific scenario-related data. A common data include several network traffic phenomena, Syslog events and critical services running on hosts.

KYPO team has realized two different scenarios so far, every scenario was successfully repeated several times (with different participants). In 2015 blue teams defended a complex power plant control system, while in 2016 their goal was to protect a train transporting a danger material.

6. Conclusion

In this paper, we have presented a cloud-based cyber range aiming to study various cyber threats. This cyber range is primarily focused on education and practical exercises. Discussed unified relational scheme for data storage together with a powerful web-based portal technology presents a flexible and scalable solution enabling us to prepare and realize a wide variety of security scenarios at run-time without modifying or manually adapting the cyber range infrastructure. Usability of our solution was verified by several practical demonstrations including the largest security drill in Czech Republic co-organized with Czech National Security Authority. On the other hand, we also met several obstacles and limitations. Major issues are summarized in lessons learned.

6.1 Lessons learned

Unified monitoring. This paper addresses only the data storage part of monitoring infrastructure. Data gathering by means of probes or monitoring software running in the background inside the cyber range infrastructure was omitted. However, realized exercises show that the preparation of a monitoring infrastructure is very laborious and still far from automation.

NoSQL databases as an alternative to rational databases. Our approach of storing complex cyber security phenomena in rational databases seems to be generic and useful. On the other hand, there exist non-rational databases which do not depend on fixed rational scheme so strictly. Utilization of these technologies in cyber ranges would bring event better scalability and flexibility to scenario-specific data. However, even they do not solve the monitoring issues, neither the adaptive GUIs issues.

Portlets have to be extremely configurable by users. Consider the topology visualization, for instance. This portlet is very complex, supporting many visualization and interaction features, e.g. visualization of roles and states of network nodes, events, utilization of links, etc. However, a concrete scenario often requires only a subset of these features to be available. And even worst, many scenarios require different features to be available to different participants involved in the scenario. Although the Liferay portal technology employed in the KYPO Cyber Range enables developers to implement some kind of run-time configuration options, the provided mechanism is far from being suitable for our purposes. Solving this problem is necessary, otherwise the preparation of user-defined scenarios can not be done entirely at run-time and, on the contrary, requires changes in portlets code.

Acknowledgements

This research was supported by the Security Research Programme of the Czech Republic 2015-2020 (BV III/1 – VS) granted by the Ministry of the Interior of the Czech Republic under No. VI20162019014 – Simulation, detection, and mitigation of cyber threats endangering critical infrastructure, and the Specific University Research provided by MŠMT.

References

Burtsev, A., Radhakrishnan, P., Hibler, M. and Lepreau, J. (2009) “Transparent Checkpoints of Closed Distributed Systems in Emulab”, Proc. of the 4th ACM European Conf. on Computer Systems, Nuremberg, Germany, pp 173-186.

Čeleda, P., Čegan, J., Vykopal, J., Tovarňák, D. (2015) KYPO – A Platform for Cyber Defence Exercises, STO-MP-MSG-133: M&S Support to Operational Tasks Including War Gaming, Logistics, Cyber Defence, Munich, Germany.
Chiang, C.J., Poylisher, A., Gottlieb, Y. and Serban, C. (2013) "Cyber Testing Tools and Methodologies", Presentation at ITEA, November.
CyberVan (2017) Cyber Virtual Ad hoc Network, [online], Applied Communication Sciences, www.appcomsci.com/research/tools/cybervan.
Davis, J. and Magrath, S. (2013) A survey of cyber ranges and testbeds, technical report, DTIC Document, 2013.
Eichler, Z., Ošlejšek, R., Toth D. (2015) KYPO: A Tool for Collaborative Study of Cyberattacks in Safe Cloud Environment, HCI International 2015: Human Aspects of Information Security, Privacy, and Trust, Los Angeles, pp 190-199.
Ferguson, B., Tall, A., Olsen, D. (2014) National cyber range overview, IEEE Military Communications Conference, pp 123-128.
Fowler, M. (1007) Analysis patterns: reusable object models, Addison-Wesley Professional.
Garcia, A.P., Siaterlis, C. and Masera, M. (2012) "Designing repeatable experiments on an emulab testbed", Broadband Communications, Networks, and Systems, Vol 66, pp 28-39.
Hofstede, R. et al. (2014) "Flow Monitoring Explained: From Packet Capture to Data Analysis with NetFlow and IPFIX", *IEEE Communications Surveys Tutorials*, Vol PP, No. 99, pp 1-1.
Johnson, D., Gebhardt, D. and Lepreau, J. (2008) Towards a High Quality Path-Oriented Network Measurement and Storage System, Springer Berlin Heidelberg, Berlin, pp 102-111.
Kouřil, D. et al. (2014) "Cloud-based Testbed for Simulation of Cyber Attacks", Proc. of the Network Operations and Management Symposium, Krakow, Poland.
Leblanc, S.P., Partington, A., Chapman, I. and Bernier, M. (2011) An overview of cyber attack and computer network operations simulation. Proc. of the 2011 Military Modeling & Simulation Symposium, pp 92-100.
Milojicic, D., Llorente, I.M. and Montero, R.S. (2011) OpenNebula A Cloud Management Tool, *IEEE Internet Computing*, Vol 15, No. 2, pp 11-14.
Rossey L. (2015) SimSpace Cyber Range, ACSAC 2015, Panel: Cyber Experimentation of the Future (CEF): Catalyzing a New Generation of Experimental Cyber-security Research.
Sezov, R. Kim, B. (2012) "Liferay in action", Shelter Island: Manning Publications Co.
Siaterlis, C., Garcia, A.P. and Masera, M. (2011) Using an Emulation Testbed for Operational Cyber Security Exercises, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 185-199.
Siaterlis, C., Garcia, A.P. and Masera, M. (2013) "On the use of emulab testbeds for scientifically rigorous experiments", Communications Surveys & Tutorials, IEEE, Vol 15, No. 2, pp 929-942.
Velan, P. and Krejčí, R. (2012) "Flow Information Storage Assessment Using IPFIXcol", Dependable Networks and Services, Vol 7279, pp 155-158.
White, B. et al. (2002) "An Integrated Experimental Environment for Distributed Systems and Networks", *SIGOPS Oper. Syst. Rev.*, Vol 36, December, pp 255-270.
Wroclawski, J. et al. (2016) "DETERLab and the DETER Project", The GENI Book, Springer International Publishing, pp 35-62.