# Security of ISES Measureserver® Module for Remote Experiments against Malign Attacks

http://dx.doi.org/10.3991/ijoe.v10i3.3132

M. Gerža[1], F. Schauer[1,2] and R. Jašek[1]
[1] Tomas Bata University in Zlín, Zlín, Czech Republic
[2] University of Trnava, Trnava, Slovak Republic

***Abstract*—This paper focuses on the security of the remote laboratories formed by remote experiments against malign attacks. Surprisingly, in spite of the fact the remote laboratories have been existing for at least three decades but virtually no attention has been devoted to this vital subject. Various malign attacks are analyzed in detail related to the Measureserver® module, which is the basic unit of the ISES (Internet School Experimental System) remote experiments accessed by clients via web pages. Such the laboratories are sometimes called e-laboratories. In the introduction a state of the art, basic features and principles are described. The following chapter analyses general security risks of remote laboratories for their software, hardware and specific risks. Next, possible malicious attacks are determined which could affect the Measureserver module. Finally, the suggestions for adequate security software and hardware solutions are outlined to prevent such attacks.**

***Index Terms*—ISES, Measureserver, remote laboratory, remote experiment, malign attack, security.**

## I. STATE OF THE ART OF REMOTE LABORATORIES

The traditional approach of teaching and dissemination of scientific educational subjects, oriented on both students at secondary schools and universities, are quite outmoded and not so popular. The contemporary students demand a higher level of the learning methods, which help more efficiently to perceive phenomena of the real world. An accessibility of experimental tools is also a factor that is important for students who often prefer studying scientific subjects via the Internet. This development reveals several new important ICT (Information and Communication Technologies) tools. One of them is a remote laboratory (RL) with remote experiments (RE) so called the e-laboratories accessible globally in the time regime 24/7, accessible in majority cases free of cost (the present state of remote laboratories may be found in recent monographs [1][2]). Their physical SW&HW, as well as the informatics SW&HW differ widely [3].

RLs are built on the ISES that has been entirely developed for educational purposes. The ISES is a complex tool for the real time acquisition and remote data acquisition, data processing, experiments control and other processes. It is an open system consisting of a basic ISES hardware with the ISES WIN software intended for local computer oriented laboratory and ISES WEB Control Kit for its remote counterpart [4]. Recently, the friendly environment for an intuitive compiling of the control programs of REs was written called EASY REMOTE - ISES (ER - ISES) as the superstructure based on the ISES equipment. The ER - ISES has been invented and developed in cooperation with TBU in Zlín and Charles University in Prague [5][8]. Let us briefly describe the functioning principle of ISES RL.

The control system of ISES RE is depicted in Figure 1 and its functioning is described in the client-server communication scheme in Figure 2.

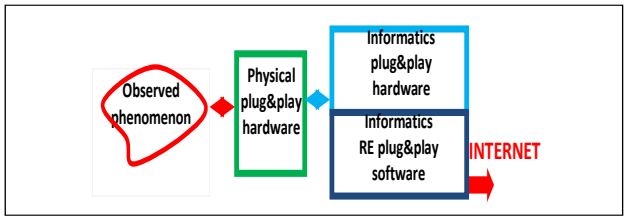


Figure 1. The HW&SW components of ISES remote experiment [9]

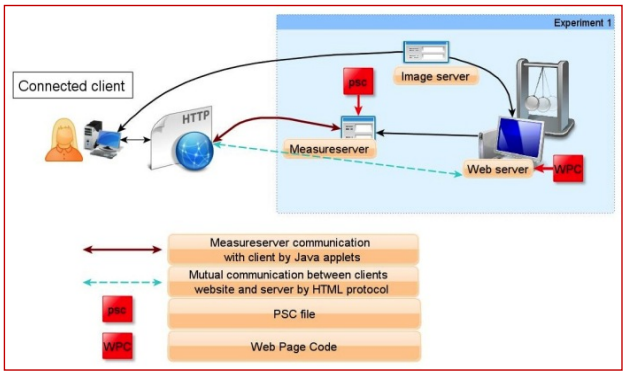


Figure 2. The arrangement of ISES remote experiment [11]

The first component of ISES REs is the apparatus (physical HW), which is the most important hardware component including all the equipment in order to examine and measure the phenomenon. The apparatus consists of ISES probes, sensors, meters, (volt-meter, ampere-meter, ohm-meter, flow-meter), thermocouple, etc. It is wired to the ISES panel transferring electrical signals incoming from those measurement accessories to the PCI AD-DA (Analog-to-Digital - Digital-to-Analog) converter installed on a control computer to gather and process all the data from the apparatus.

The second component is the Measureserver module, the most important software component of ISES REs functioning as the finite state machine (FSM) [12]. It directly communicates with the apparatus, processes the

measured data and control commands. The main feature of Measureserver is setting the ISES panel, sensors with meters for a data collection and processing control commands. The Measureserver dynamically responds to signals incoming from the apparatus, as well as all the commands transmitted from the client's web page. Measureserver has the following three components:

- Measureserver's core,
- Hardware plug-in,
- PSC script file.

The Measureserver core is responsible for the data and commands transfer, clients serving and for the execution of control commands, which is controlled by the PSC script file imported to Measureserver. The PSC script is a unique programming language specially designed for ISES REs. This language is a non-compliable that defines a behavior of the Measureserver's core. The hardware plug-in provides needed functionality to control the ISES panel transferring signals from/to the apparatus [6].

The third component is called HttpRelayServer. This component cooperates with the Nginx web server and takes care about available ports needed for the LAN (Local Area Network) communication. It is a common habit by system administrators to block all ports for security reasons except ports defined as 80 and 443. HttpRelayServer dynamically monitors and changes the communication to the available port.

The fourth component is ImageServer in order to serve for life view of ISES REs. The module periodically saves an actual view of the realized RE from the USB web camera to store the image data file [7].

The fifth component is the web server intended for the communication between RE and the client's side. At present, Nginx is used due to its sufficient stability to transfer the RE web page to the client's computer.

The sixth and the last component is the RE web page providing clients all the available features of running ISES RE via standard web browsers.

Clients have a non-stop accessibility to enter ISES RE via their web browsers connected to the Internet from anywhere. The most significant advantage is real-time experimenting with the apparatus installed in a remote laboratory. The apparatus sends measured data through particular subsystems to all entered clients. After the accomplishment, the active client is allowed to sort, filter and visualize all the received data in graphical charts and well-arranged tables to find corresponding answers regarding the observed phenomena.

Surprisingly, in spite of the fact the RLs have been existing for at least three decades [10], virtually no attention has been devoted to the vital subject of their security that is the topic of the paper. The paper analyses the security of RLs, especially the Measureserver module. The following chapter describes general security risks of RLs due to their software, hardware and specific risks. Next, the potential and existing attacks are analyzed that threaten Measureserver and the most vulnerable ISES RE components. It concerns the unauthorized access into the Measureserver module, faking queries performed via communication protocols and failure of hardware components in the apparatus. Finally, the suggestions are outlined for adequate security software and hardware solutions related to the Measureserver module.

## II. GENERAL SECURITY RISKS OF REMOTE EXPERIMENTS

Let us enumerate already occurring general security risks of RLs. The security aspect is very important for real-time experimenting, especially when newcomers enter the system to try RE in order to test its capabilities. Unreliable and malfunctioning REs may cause a bad reputation that is perceived negatively by some potential clients. Such affection can also seriously degrade usefulness of this new educational tool.

All REs share some common security risks. The RE consists of the physical & informatics HW and informatics SW. Let us now describe the common security risks for individual components of the RE as presented in Figure 1.

*Physical and Informatics HW security aspect*: As the name suggests, this category includes deliberate damage of physical HW. This aspect is given in the most of cases by situating the Res in a building and locked room. Adequate and regular inspection including maintenance of the REs equipment is a stringent and often underestimated condition for running of the e-laboratory. The hardware of REs can be damaged as well through the fault or mishandling by connected clients. All these circumstances should have been attended to the controlling program of REs [10].

*Informatics SW security aspect:* This kind of attacks comes from the Internet in the most cases. Since the apparatus is completely controlled and monitored by a computer (server) connected to LAN, across which the access to the Internet is realized, all the applications residing on that computer are vulnerable to unauthorized people or hackers. These attacks lead to the harming or overloading of involved computer or damage front end applications like is Measureserver and HttpRelayServer. Intentional unauthorized intrusions to Measureserver and its possible damage are analyzed later in the paper. This sort of attacks also relates to the web applications at the client's side. All the potential security risks of web application are discussed later due to their importance.

*Security aspect due to the environment:* This security aspect deals with all the failures of the power supply like power outage or voltage spikes. Usually, the surge protection and circuit breakers are adequate security for elementary experiments. However, the experiments using expensive measuring and controlling devices require comprehensive security measures.

### *A. Security risks due to web applications*

If unknown users can access the web application, they are potentially malicious users who can try to gain unauthorized access. Servers connected to the public on the Internet are constantly probed for vulnerabilities. Therefore, it is recommended to take precautions and build security into all involved web applications.

Implementing adequate security mechanism is only part of the solution. Another important part of the efficient measures is vigilance. Even if the system has many security safeguards, it is needed to watch it closely in following precautionary ways:

- Monitor the operating system's event logs. Watch for repeated attempts to log into the system or for excessive requests being made against the web server.

- Continually keep the application server up to date with the latest security updates, as well as any updates for other data sources that the application might use.

The important part of developing more secure web application is to understand well the threats to it. For example, Microsoft has developed a way to categorize threats. The subchapter below describes these threats and how they apply to the web applications [13].

*Spoofing* - to spoof is to impersonate a user or process in an unauthorized way. At its simplest, spoofing can mean typing in a different user's credentials. Malicious users might also change the contents of a cookie to pretend that he/she is a different user or that the cookie comes from a different server. In general, user can help prevent spoofing by using stringent authentication. Any time someone requests access to non-public information, be sure they are who they say they are. User can help defend as well against spoofing by keeping credential information safe. Such as, not to keep a password or other sensitive information in a cookie, where a malicious user can find or modify it.

*Tampering* - means changing or deleting a resource without authorization. One example is defacing a web page, where the malicious user gets into your site and changes files. An indirect way to tamper is by using a script exploit. A malicious user manages to get script to execute by masking it as user input from a page or as a link. A primary defense against tampering is to actively use e.g. Windows security to lock down files, directories and other resources. The application should also run with minimum privileges. User helps guard against script exploits by not trusting any information that comes from a different user or even from a database. Whenever user gets information from an untrusted source, he takes steps to be sure it does not contain any executable code.

*Repudiation* - this kind of threat involves carrying out a transaction in such a way that there is no proof after the fact of the principals involved in the transaction. In a web application, this can mean impersonating an innocent user's credentials. User can help guard against repudiation by using stringent authentication. In addition, use e.g. the logging features of Windows to keep an audit trail of any activity on the server.

*Information Disclosure* - simply means stealing or revealing information that is supposed to be private. A typical example is the stealing passwords but information disclosure can involve access to any file or resource on the server. The best defense against information disclosure is to have no information to disclose. For example, if user avoids storing passwords, malicious users cannot steal them. An alternative is to store only a hash of password. When different user presents credentials, user can hash the user's password and compare only the hashes of the two. If user does store sensitive information, he can use e.g. Microsoft Windows security to help secure it. As always, user should simply use up authentication to help ensure that only authorized users can access restricted information. If user has to expose sensitive information, it is recommended that a user encrypts the information when stored and use SSL (Layer Secure Sockets) to encrypt the information when sent to and from the browser.

*Denial of Service* - this attack deliberately causes an application to be less available than it should be. A typical example is to overload a web application so that it cannot serve ordinary users. Alternatively, malicious users might try to simply crash your server. Internet Information Services (IIS) enables user to throttle applications, which means that it limits the number of requests it will serve. User might be able to deny access to other users or IP addresses known to be malicious. Keeping the applications online is a matter of running robust code. User should test the application thoroughly and respond to error conditions wherever possible.

*Elevation of Privilege* - this attack uses up malicious means to get more permissions than normally assigned. For example, in a successful elevation-of-privilege attack, a malicious user manages to get administrative privileges to the web server, giving himself or herself access to any data on the server as well as control over server capabilities. To help protect against the elevation of privilege, authorized user should run the application in a least-privilege context if practical. It is recommended for a certainty that such user does not run applications based on ASP.NET (Active Server Pages developed by .NET language) as the system (administrative) user [13].

Clients should also use general security equipment when using the Internet. The most frequent used equipment is the antivirus software to prevent, detect and remove malware like computer viruses, malicious BHOs (Browser Helper Objects), hijackers, keyloggers, rootkits, trojan horses, worms, malicious LSPs (Layered Service Providers), adware and spyware.

The second used security equipment is the firewall, SW or HW based network security system that checks the incoming and outgoing network traffic by analyzing the data packets on a rule set and determining whether they should be allowed through or not. The firewall builds a bridge between the internal computer network it protects and the client after checking if the connected network is secure and trusted enough as shown in Figure 3 [14].

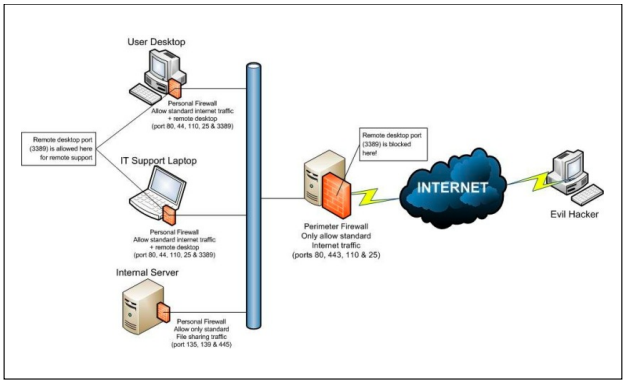


Figure 3: The scheme of firewall functioning [16]

## III. SECURITY RISKS OF MEASURESERVER

The Measureserver module substantially contributes to the security risks. Since the module is the most important software component of REs, let us deal with these risks in more detail. Next, the analysis is performed about the likeliest attacks on that module by unauthorised clients or hackers in order to eliminate it or slow it down.

### *A. Denial of service attack*

This kind of attack has been shortly mentioned in the previous chapter. Since the attack significantly affects the

Measureserver and web server, it is indispensable to analyze this attack. Both components reside on the same computer (server) connected indirectly through LAN or directly to the Internet by means of the static IP address. As mentioned, Measureserver acts as the communication server between the ISES panel attached to an apparatus and the client. The web server provides client a web page of the RE. The denial of service attack (DoS attack) as illustrated on Figure 4 or the distributed denial of service attack (DDoS attack) may attempt to make the server computer (or network resource) unavailable to all the authorized clients using RE. Although the means to carry out, motives for, and targets of a DoS attack may vary. It consists of the efforts to temporarily or indefinitely interrupt or suspend important services of a hosting server which is connected to the Internet.

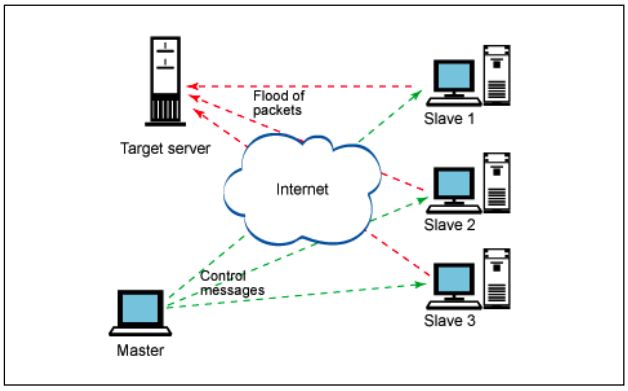


Figure 4: The typical scheme of denial of service attack [17]

One common method of the attack involves saturating server computer with external communications requests so much that it cannot respond to legitimate traffic or responds so slowly as to be rendered essentially unavailable. These attacks usually lead to a server overload. This attack is aimed at either forcing the targeted computer to reset or consuming its resources so that it can no longer provide its intended service or obstructing the communication between legal clients and the victim so that they can no longer appropriately communicate. Such kind of the attack negatively affects Measureserver with respect to the authorized clients.

Symptoms and manifestations have been defined for DoS attack as follows [15]:

- Inability to access any web site.
- Disconnecting from the Internet.
- Unavailability of a particular web site.
- Unusually slow network performance.
- Increase in the number of spam emails received.

A denial of service attack may include execution of malware intended to:

- Take the maximal processor's usage.
- Trigger errors in the microcode of the machine.
- Trigger errors in the sequencing of instructions in order to force the targeted computer into an unstable state or wholly lock-up for all authorized clients.
- Exploit errors in the operating system, causing resource starvation and/or thrashing.
- Crash the operating system itself.

In most cases, the denial of service attacks involve forging of IP sender addresses called IP address spoofing, so that the location of the attacking machines cannot be easily identified and to prevent filtering of the packets based on the source address [15].

### *B. Cross-site scripting attack*

Cross-site scripting (XSS) is a type of the computer security vulnerability typically found in web applications. XSS allows attackers to inject client's side script into web pages browsed by other users. XSS vulnerability may be used by attackers to bypass access controls such as the same origin policy. An effect of the attack may range from a petty nuisance to a significant security risk depending on the sensitivity of the data handled by the vulnerable site and the nature of any security mitigation which is implemented by the site's owner.

The Measureserver module can be seriously affected through the web applications (browsers), which clients connected to the RE have to use. The web browsers intermediate the web page containing data outcomes and visualization. The client's web page is provided by the Nginx web server residing on the same computer as Measureserver. When a client completely loads a web page of the RE then the direct communication is established between that page and Measureserver. Since the web page contains many scripts written in JavaScript language and applets (small Java applications) so the risk of such attack is really high. Every implemented script or applet can be hit in order to modify them. After the malicious intervention, the modified script/applet can behave weirdly or uncontrollably with respect to the Measureserver, e.g. to send undefined values/events for processing or to perform cyclically some operations. Exploited operations can lead to overloading of the Measureserver or seriously damage the RE by the reason of short/long time cycling.

Let us now to analyze the cross-site scripting attack. Security on the web is based on a variety of mechanisms including underlying concept of trust known as the same origin policy. This states that if the content from one site is granted permission to access resources on the system, then any content from that site will share these permissions, while content from another site will have to be granted permissions separately.

Cross-site scripting uses known vulnerabilities in web based applications, their servers or plug-in systems they rely on. Exploiting one of these, they fold malicious content into the content being delivered from the compromised site. When the resulting combined content that arrives at the client's web browser, it has all been delivered from the trusted source and thus operates under the permissions granted to that system. By finding ways of injecting malicious scripts into web pages, an attacker can gain elevated access-privileges to sensitive page content, session cookies or a variety of other information maintained by the browser. Cross-site scripting attacks are therefore a special case of the code injection.

The cross-site scripting expression originally referred to the act of loading the attacked third-party web application from an unrelated attack site, in a manner that executes a fragment of JavaScript prepared by the attacker in the security context of the targeted domain The definition gradually expanded to encompass other

modes of code injection, including persistent and non-JavaScript vectors (including Java, ActiveX, VBScript, Flash or HTML), causing some confusion to newcomers to the field of application security.

Most experts distinguish between at least two primary flavors of XSS: non-persistent and persistent shown in Figure 5. Some sources further divide these two groups into traditional (caused by server-side code flaws) and DOM-based (in client-side code) [18].

*Non-persistent* - The non-persistent XSS vulnerability is by far the most common type. These holes show up when the data provided by a web client, most commonly in activating HTTP communication query parameters or in HTML form submissions, is used immediately by server-side scripts to parse and display a page of results to that user without sanitizing the request.

*Persistent* - The persistent XSS vulnerability is more devastating variant of a XSS flaw: it occurs when the data provided by the attacker is saved by the server and then permanently displayed on normal pages returned to other users in the course of regular browsing without proper HTML escaping. Persistent XSS can be more significant than other types because an attacker's malicious script is rendered automatically without the need to individually target victims or lure them to a third-party website. Particularly in the case of social networking sites, the code would be further designed to self-propagate across accounts, creating a type of a client's side worm.

*Traditional versus DOM-based vulnerabilities* - The traditional vulnerabilities, XSS would occur in server's side code responsible for preparing the HTML response to be served to the client. DOM-based type occurs in the content processing stages performed by the client, typically in client's side JavaScript [18].

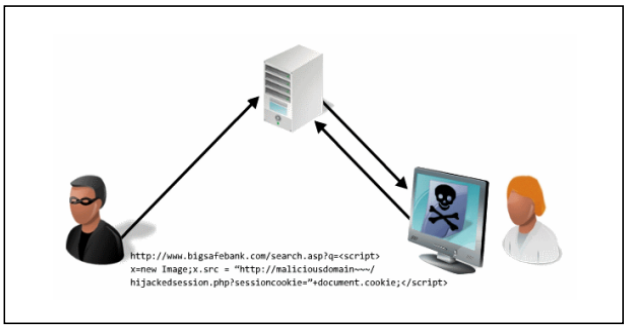


Figure 5: The typical scheme of persistent cross-site scripting [19]

### C. *Brute force attack*

The brute force attack is used by intruder who attempts to gain access to a server by guessing a user password, usually the root administrator, through the SSH server (based on the cryptographic network protocol for secure data communication), mail server or other service running on the server. The attacker will use software that will check every possible combination to find the one that works. Brute force detection software will alert client when multiple failed attempts to gain access are in progress and disable access from the obtrusive IP address.

It is worth mentioning the automated SSH brute force attack shown in Figure 6, which consists of the following phases in order to obtain valid information to log in the exposed system on a server [20]:

- Attacker scans for present SSH daemons which are exposed to the Internet.
- The scans can be done by bots (software applications that run automated tasks over the Internet). It usually scans target the regular SSH TCP 22 port.
- Assuming the TCP 22 port is found open, service identification is attempted.
- SSH banners can expose info about the SSH service and the underlying operating system.
- Based on the gathered information, attacker tries to discover valid username and password combinations through SSH brute force attacks. These are not quite brute force attacks but an attacker rather uses weak passwords dictionaries for this purpose.
- A prime target will be the root account in principle to enter the attacked system.
- If root login over SSH is allowed with a weak password, so the attacker may end up in complete control of the exposed system to make any illegal intervention [20].

The brute force attack could be eventually applied on a central computer where the Measureserver resides and provides services between the authorized clients and ISES RE. This attack is probable because the hosting computer is connected through the static IP address to the Internet. An administrator of the REs has always an option to enter the computer by username and belonging password via the Remote Desktop application which is an integral part of the Window operating system. Such a way of the administrating is important in order to maintain the REs by responsible person. If any hacker succeeds in his malicious intervention to break through the authentication, then all the installed applications reveal him on the computer including the operating system. It means the potential intruder would affect Measureserver, e.g. to shut it down or re-configure it to exploit the equipment of the REs. This inexperienced encroachment would have an unexpected impact on the reliability and performance.

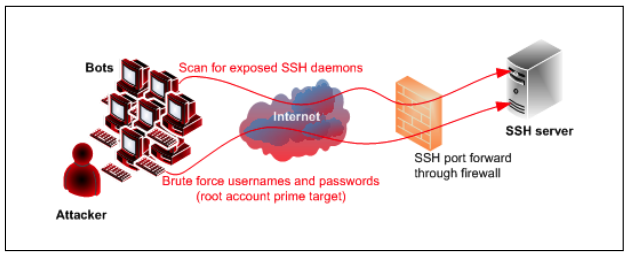


Figure 6: The example of automated SSH brute force attack [20]

## IV. SUGGESTION OF SECURITY SOLUTIONS

Every realized RE consists of many various SW and HW components communicating with each other by packets, protocols and signals. These components are controlled by the central computer that is exposed to clients on the Internet. Hence, it is important to deal with the security policy and countermeasures to prevent malicious attacks leading to the damage of the REs.

### A. *SW countermeasures*

The countermeasures should include several security mechanisms for an efficient defense of the exposed computer covering the Measureserver and other related

applications. The following recommendations should be realized to avoid the mentioned attacks.

*1) Authenticating and account management before connection to the network.*

- All accounts should have strong passwords.
- Administrative or root accounts should have even stronger passwords or pass phrases.
- Administrator or root account should be used only when absolutely necessary.
- Adding service accounts should be avoided to the default administrators group.
- Assignment of a unique administrative account and password should be done for each individual to distinguish activities among administrators.
- Regular review of the access list or log for users should be done in order to look for unexpected rights or changes.

*2) Installing and patching the operating system before connection to the network.*

- Installed software should be current on the computer. The operating system and applications should be vendor supported for security patches.
- When installing applications, it is important to make sure to install applications that are needed only and to install the latest versions of all applications including available security patches.

*3) Using necessary operating system and services.*

- Configuration of installed applications should be done, to disable all unused features and to limit the availability of any enabled features.
- Computer should only provide services needed for its role in an organization.
- Elimination of Telnet and FTP should be done, to use SSH instead.

*4) Installing adequate filters or firewall.*

- Installation and configuration of a packet filtering utility should be done like the TCP wrappers or SW firewall to protect individual services.
- The rules should reflect the acceptable usage and security policies which have been defined for the target computer.
- Operating system filters that deny or permit certain traffic should be used if available.
- Periodical review of the filters should be done for inappropriate or unneeded access.

*5) Setting up and reviewing intended logs.*

- Configuration of all used services should be done so that they log all connections and authentication information. Therewithal, forwarding all logs is recommended to a highly secure computer.
- Someone should be assigned the responsibility to review possible security violations identified in the system logs.

*6) Installing security related software.*

- Installation of anti-virus or other virus filtering applications should be done including daily updating process for the latest virus definitions.
- SSH or other encrypted and secure methods of access should be installed if remote access or remote administration services are needed for a maintenance. SSH improves the security of user accounts by efficient encrypting login sessions.

### *B. HWcountermeasures*

The countermeasures provide administrator of ISES RE the following mechanisms for the defense of the exposed computer residing in a laboratory.

*Locking up the room* - Even before the administrator locks down the computer, in fact, before he even turns it on for the first time, he should ensure that good locks are used on room doors. Of course, the best lock in the world does no good if it is not used, so the administrator also needs policies requiring those doors to be locked any time the room is unoccupied and the policies should set out who has the key or key code.

*Setting up surveillance* - Locking the door to the computer room is a good first step but someone could break in or someone who has the authorized access could misuse that authority. An administrator needs a way to know who goes in, out and when. A log book for signing in and out is the most elemental way to accomplish this but it has a lot of drawbacks. Persons with malicious intent are likely to bypass it. A better solution than the log book is the authentication system incorporated into the locking devices like the smart card, token, or biometric scan to unlock the doors and the record is required to identify persons who enter the laboratory.

*Backing up data* - Backing up important data is an essential element in a disaster recovery but the administrator must not forget that the information on those backup tapes or disks can be stolen and used by someone outside the building. Many administrators keep the backups next to the computer in the room. They should be locked in a drawer or at the similar place. Ideally, a set of backups should be kept off site and the administrator must take care to ensure that they are secured in that offsite location.

*Disabling unnecessary drives* - If the administrator does not want on-site persons copying information from the computer to removable media, he can disable or remove floppy drives, USB ports and other means of connecting external drives on the computer.

## V. CONCLUSIONS

The remote experiments provide students and teachers with a new way of absorbing information in the field of natural sciences and technological sciences. This modern approach of education allows all interested people to clearly understand particular phenomena by the real experimenting. These REs enable remote touching of specific equipment in the e-laboratory. Students are allowed to monitor only or to monitor and control together the RE according to the type of chosen sophistication and complexity. This way gives a chance for students of a distant study or to those who cannot, for different reasons, participate on regular laboratory classes to understand the taught subject matter.

The REs are perceived as the user friendly tool for easy entering the e-laboratory via the Internet by web browsers anywhere. This advantage unfortunately brings security risks for involved SW and HW modules. Some of the modules are more resistant against malicious attacks but some of them are more or less vulnerable. Attackers generally exploit weak points and flaws in particular SW

components to control them partially or wholly in order to damage RE, decelerate it or cause shutdown. Hence, the security aspect becomes more important nowadays to ensure the proper operation for authorized students and teachers. Unstable or malfunction REs exert a bad effect in the eyes of prospective and existing users.

Attackers have to often struggle with various SW and HW countermeasures, which are breached sometimes. Fortunately enough, these attacks are occasionally reduced or fenced off. That is why this issue is analyzed in detail for the purpose of finding adequate measures to minimize or prevent such malicious attacks. This analysis uncovers common and specific security risks, which would seriously affect the most important components of the REs. The paper especially deals with the core component called Measureserver intended for a direct communication among the clients and the RE apparatus. All the possible security aspects are described in detail for the physical & informatics HW and informatics SW.

Another serious danger constitutes the security risks of web applications due to the remote connecting of the interested clients via web browsers. The client's interface allows potential attackers to apply various intrusion methods to exploit possible flaws on the hosting computer where all the important SW components reside. The description is not so deep, it just concerns of threats like the spoofing, tampering, repudiation, information disclosure, denial of service and elevation of privilege.

The major topic of the paper is focused on the security risks of the Measureserver which can be damaged or even crashed by malicious attackers. The most probable threat is the denial of service attack that can significantly decelerate or put aside the hosting computer controlling the RE. The next threat is the cross-site scripting used up for injecting client's side script into web pages browsed by other users who wish for experimenting. The last threat is the brute force attack performed by intruders who attempt to gain access to the hosting computer by guessing a password, usually the root administrator.

Finally, we provide the efficient security solutions. At first, the SW countermeasures are proposed including several common and specific security mechanisms for the practical defense. The description comprises the client's authentication, patching the operating system, review logs usage, security-related applications and others. Furthermore, the HW countermeasures are provided. They consist of the locking up the computer room, setting up surveillance and backing up important data to prevent or mitigate physical attacks.

## REFERENCES

[1] J. Garcia-Zubia and G. Alves, eds., *Using Remote Labs in Education*, Ed. University of Deusto, Miller, F. P., 2009.

[2] K. Azad, M. Auer, J. Harward, *Internet Accessible Remote Laboratories: Scalable E-Learning Tools for Engineering and Science Disciplines*, Ed. IGI Global. USA, 2012. ISBN: 9781613501863. http://www.igi-global.com/book/internet-accessible-remote-laboratories/52730 http://dx.doi.org/10.4018/978-1-61350-186-3

[3] J. Garcia-Zubia, P. Orduna, D. Lopez-de-Ipina, G.. Alves, *Addressing Software Impact in the Design of Remote Laboratories*, IEEE Transactions on: *Industrial Electronics*, Volume 56, Issue 12, pp. 4757-4767, 2009. http://dx.doi.org/10.1109/TIE.2009.2026368

[4] F. Schauer, I. Kuřitka and F. Lustig, *Creative Laboratory Experiments for Basic Physics Using Computer Data Collection and Evaluation Exemplified on the Intelligent School Experimental System (ISES)*, in Innovations 2006, *World Innovations in Engineering Education and Research*, iNEER Special Volume, W. Aung et al. (eds.), pp. 305-312, USA, 2006.

[5] F. Schauer, M. Krbecek and M. Ožvoldová, Controlling Programs for Remote Experiments by Easy Remote - ISES (ER-ISES), *Proceedings of the International Conference on Remote Engineering and Virtual Instrumentation - REV 2013,* Sydney University, 6-8 February 2013.

[6] ZEMAN, Petr. Softwarové prostředí pro integraci naměřených dat ze vzdálené laboratoře a simulace. *Studentská Tvůrčí a Odborná Činnost 2012*. Ostrava: VŠB Ostrava, 2012.

[7] ZEMAN, Petr. Softwarové prostředí pro řízení vzdálených experimentů. Ostrava, 2011.

[8] SCHAUER, František, František LUSTIG a Miroslava OŽVOLDOVÁ. *Innovations 2011: World Innovations in Engineering Education and Research: Internet Natural Science Remote e-Laboratory (INTRE-L) for Remote Experiments*. USA: iNEER, 2011, s. 51-68. 1. ISBN 978-0-9818868-2-4.

[9] SCHAUER, František a Miroslava OŽVOLDOVÁ. Plug and play system for hands on and remote laboratories. In: *Proceedings of 8th International Conference on Hands-on Science*. Ljubljana: University of Ljubljana, 2011, s. 17-21. ISBN 978-989-95095-7-3.

[10] KRBEČEK, Michal a František SCHAUER. Security aspects of remote e-laboratories. Zlín: UTB ve Zlíně, Fakulta aplikované informatiky, 2012.

[11] KRBEČEK, Michal. *ISES Remote experiment - present state*. Zlín. UTB ve Zlíně, Fakulta aplikované informatiky, 2013.

[12] Finite state machine. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [read: 2013-07-18]. https://en.wikipedia.org/wiki/Finite-state_machine

[13] Overview of Web Application Security Threats. *MSDN Library* [online]. USA: Microsoft, 2013 [read: 2013-06-18]. http://msdn.microsoft.com/en-us/library/f13d73y6%28v=vs.100%29.aspx

[14] Firewall (computing). In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [read: 2013-06-18]. http://en.wikipedia.org/wiki/Firewall_%28computing%29

[15] Denial-of-service attack. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [read: 2013-06-18]. https://en.wikipedia.org/wiki/Denial-of-service_attack

[16] Security: Firewalls. [online]. Calne, UK: Assyst Solutions Limited [read: 2013-06-18]. http://www.assystsolutions.com/ourservices/security.htm

[17] E-Commerce security: Attacks and preventive strategies: Denial of service attacks. [online]. Toronto, Canada: IBM, 2005 [read: 2013-06-18]. https://www.ibm.com/developerworks/library/co-0504_mckegney

[18] Cross-site scripting. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [read: 2013-06-18]. http://en.wikipedia.org/wiki/Cross-site_scripting

[19] Cross-Site Scripting (XSS) Tutorial: Learn About XSS Vulnerabilities, XSS Injections and How to Prevent Cross Site Scripting Attacks: Persistent XSS. [online]. Burlington, USA: Veracode [read: 2013-06-18]. http://www.veracode.com/security/xss

[20] SSH Brute Force Attacks: Anatomy of automated SSH brute force attacks. [online]. Noblesville, USA: Dreaming Tree Technology [read: 2013-06-18]. http://www.firewalls.com/blog/ssh_brute_force_attack

## AUTHORS

**M. Gerža, F. Schauer** and **R. Jašek** are with the Tomas Bata University in Zlín, Faculty of Applied Informatics, Nad Stráněmi 4511, Zlín, 760 05, Czech Republic. **F. Schauer** is also associated with University of Trnava, Faculty of Education, Priemyselná 4, 917 01, Trnava, Slovak Republic (michal.gerza@email.cz, fschauer@fai.utb.cz, jasek@fai.utb.cz).

Submitted 19 August 2013. Published as re-submitted by the authors 28 April 2014.