Security risks of java applets and possible solutions for remote laboratories

P. Špiláková, R. Jašek, and F. Schauer

***Abstract*—**The contribution deals with the Java language security, predominantly the holes, which can be misused by the hackers´ attacks. Emphasis is on Java applets, used for ISES (Internet School Experimental system) control software of remote experiments spread across the Internet. Due to the security constraints, imposed by the corporation Oracle, these applets need to be replaced by other available alternatives for web communication. For the purpose the programming languages Dart, ActiveX and JavaScript are compared from the point of view of security and the transport rate and we choose as the best one JavaScript.

***Keywords*—**ActiveX, Dart, Java applet, Java security, JavaScript, ISES remote laboratory.

I. Introduction

In this article we deal with the safety issues of Java applets and alternatives to replacing them. This issue directly affects us because in our Internet School Experimental system (ISES) remote laboratories, where Java applets have been used for remote control of ISES experiments. In 2002, when the first remote experiment "Water level control", was designed, and made available on the page http://kdt-14.karlov.mff.cuni.cz/en/mereni.html, the most appropriate method of the transformation from laboratory experiments to real remote experiments, Java applets were used. It was then the best solution for the authors of the ISES remote experiments (RE) [1 - 4] because Java applets recorded in these years its biggest boom with the brightest prospects for the future.

Remote laboratories are often used in education. Therefore, care must be taken to their safety and functionality [5], [6].

Unfortunately due to the security threats that are currently occurring on the Internet, it is not possible to continue along Java lines. The situation requires Java applets to be replaced by other approaches. Therefore, we started to analyze various replacement alternatives how to renew controlling programs for the ISES experiments in the most appropriate manner so as to preserve its functionality and to avoid security hindrances.

The days when the Local Area Network (LAN) was the most widely used are behind us. Now, most Local Networks have been connected directly to the Internet, which means the security issue is more important than ever before. We need to try to protect our data with advanced security mechanisms such as firewalls, secure shells, virtual private networks and a lot of other means, discussed later in this article. Connecting computers together in a network allows computer users to share devices, files, data, programs, and each others' computational resources. In most cases pages are not only purely static but contain a lot of dynamic content that extend their functionality and design. On the creation of dynamic content and Internet communication many tools are used, among others, JavaScript, PHP, CSS styles, and of course Java. Thanks to the development of information and communication technology (ICT) and many others the number of users connected to the Internet is growing at an increasing rate every year as can be seen in Fig. 1, which shows the growth pattern of users connected to the Internet since 1995 till today. Along with these abilities and large numbers of connected computers it comes to the question about computer security.

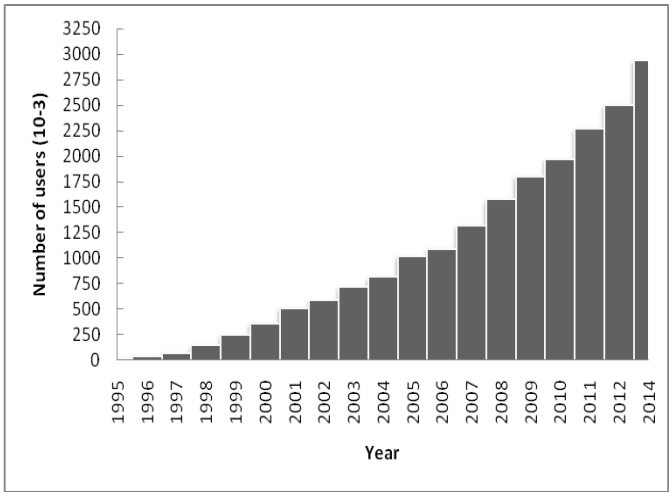

Fig. 1 History and Growth of the Internet from 1995 till Today [7]

II. Java Language

This chapter introduces the programming language Java with the main attention paid to its security policy.

There has been a lasting a problem of writing cross-platform applications. As a result was created the programming language Java which is independent on the platform. The first public version of Java was introduced at the SunWorld conference in 1995. It was created by a group of three men, James Gosling, Patrick Naughton, and Mike Sheridan

members of Sun Microsystems which was later in 2010 bought by Oracle Corporation. Due to its characteristics which are described below, the Java was rapidly spread through the internet [8].

The Oracle Corporation provides on its website the following describe of the Java programming language: Java is a high-level language that can be characterized by these characteristics below:

• Simple - It means that can be programmed without extensive programmer training while being attuned to current software practices.

• Object oriented - The programs are composed of one or more classes. Classes are collections of data objects and the methods that manipulate these data objects. Each class is one kind of object. Classes are arranged in a hierarchy such that a subclass inherits behaviour and structure from its superclass.

• Distributed - The power lies in that any agent in your system can directly interact with an object that "lives" on a remote host.

• Multithreaded - Java programs can execute more than one task at the same time.

• Dynamic - While the Java Compiler is strict in its compile-time static checking, the language and run-time system are dynamic in their linking stages.

• Architecture neutral - The Java Compiler product generates bytecodes-an architecture neutral intermediate format designed to transport code efficiently to multiple hardware and software platforms.

• Portable - The architecture-neutral and portable language platform of Java technology is known as the Java virtual machine.

• High performance - Performance is always a consideration. The Java platform achieves superior performance by adopting a scheme by which the interpreter can run at full speed without needing to check the run-time environment.

• Robust - The Java programming language is designed for creating highly reliable software.

• Secure - With security features designed into the language and run-time system, Java technology lets you construct applications that can't be invaded from outside. In the network environment, applications written in the Java programming language are secure from intrusion by unauthorized code attempting to get behind the scenes and create viruses or invade file systems [9].

Java can be run on a computer that has a Java Virtual Machine (JVM) installed. Because Java byte code runs on the Java Virtual Machine, it is possible to run Java code on any platform [8].

Java can be used to create two different kinds of programs, which are applications and applets, the main differences are that the applet needs to run under the control of a browser, whereas the application runs stand-alone, with the support of the virtual machine and the applet is subjected to more stringent security restrictions whereas application is not.

Java works that the most code is automatically downloaded from the network and runs on your computer. So computer security issues are very important [8]. Next, let us discuss the Java applets from the point of security view because remote experiments were created by applets.

III. JAVA APPLETS

In this chapter we will discuss the issue of Java applets and their advantages and disadvantages, especially their security problems, which nowadays cause great inconvenience to users.

Let us have first a definition of the Java applet, as it is defined on Oracle website [5]. “The Java applet is a special kind of Java program that a browser enabled with Java technology can download from the internet and run. The applet is typically embedded inside a web page and runs in the context of a browser. The applet must be a subclass of the java.applet.Applet class. The Applet class provides the standard interface between the applet and the browser environment [9]. “

A. What Applets Can and Cannot Do

There are two main types of applets either sandbox applets (unsigned) or privileged applets (signed). Sandbox, applets which are considered untrusted are run in a security sandbox. Privileged applets can either run in the sandbox, or can request permission to run outside the security sandbox and have extensive capabilities to access the client and don't apply to the following restrictions. Applets loaded over the network are usually considered to be untrusted code whereas local applets are considered to be more trustworthy.

Unsigned applets can perform the following operations:

• They can make network connections to the host they came from.

• They can display HTML documents.

• They can invoke public methods of other applets on the same page.

• They can read secure system properties.

When launched by using The Java Network Launch Protocol (JNLP), applets can perform:

• They can open, read, and save files on the client.

• They can access the shared system-wide clipboard.

• They can access printing functions.

• They can store data on the client, decide how applets should be downloaded and cached, and much more [9].

The JNLP is a Java protocol to allow applications to be downloaded and run on a client machine by using resources that are hosted on a remote web server.

Unsigned applets cannot perform the following operations:

• They cannot read from or write to the local filesystem. This includes operations such as:

  o Read files.
  o Write files.
  o Delete files.
  o List directories.
  o Rename files.

- Create directories.
- Check for the existence of files.
- Obtain the size or modification date of files.
- Obtain the read and write permissions of a file.
- Test whether a filename is a file or directory.

• They cannot perform networking operations, except in connections to the host they came from.

• They cannot make use of certain system facilities. For instance as are:

- Exit the Java interpreter by calling System.exit() or Runtime.exit().
- Spawn new processes by calling any of the Runtime.exec() methods.
- Dynamically load native code libraries with the load() method of Runtime or System.

• They cannot make use of certain the Abstract Window Toolkit (AWT) facilities and the following:

- Initiate a print job.
- Access the system clipboard.
- Access the system event queue.

The Abstract Window Toolkit (AWT) is Java's original platform-independent windowing, graphics, and user-interface widget toolkit.

• They cannot create or access threads or thread groups outside of the thread group in which the untrusted code is running.

• They cannot use reflection methods to obtain information about non-public members of a class, unless the class was loaded from the same host as the untrusted code.

Implementing the security restrictions is the responsibility of the java.lang.SecurityManager class about which will be discussed below [8], [9].

B. Parts of the Sandbox Model

The default sandbox consists of three interrelated parts: *Verifier*, *Class Loader*, and *Security Manager*. These parts must work perfectly because if any of the three parts crashes, the whole security system crashes, leaving the door wide open for attack.

The Security Manager depends on Class Loaders to correctly label code as trusted or untrusted. Class Loaders also shield the Security Manager from spoofing attacks by protecting local trusted classes making up the Java API. On the other hand, the class loader system is protected by the Security Manager, which ensures that an applet cannot create and use its own Class Loader. The Verifier protects both the Class Loaders and the Security Manager against language-based attacks meant to break the VM. All in all, the three parts intertwine to create a default sandbox.

The Java security model was subsequently extended to include more parts in new versions of the Java. First came the *java.security* package later came *Access Controller*, *Code Source* etc. [8].

The Verifier

As we said from the beginning the Java is cross-platform language through the use of bytecode. Java bytecode is verified before it can run. The Verifier provides a first part of defense against malicious codes that could be dangerous to users. This verification scheme is meant to ensure that the byte code, which may or may not have been created by a Java compiler, plays by the rules.

Thus, class files which contain bytecode are verified, Java automatically examines untrusted code before it is allowed to run. If the Verifier finds any problem with a class file, it throws an exception and the class file never executes.

The process of the verifying is divided into two major steps: internal checks that check everything that can be checked by looking only at the class file itself and runtime checks that confirm the existence and compatibility of symbolically referenced classes, fields, and methods.

The validated bytecode satisfies conditions that the class file has the correct format and proper length; stacks will not be overflowed or underflowed; bytecode instructions all have parameters of the correct type; no illegal data conversions occur; private, public, protected, and default accesses are legal; and that all register accesses and stores are valid [8].

As it is known, the length of time which it takes for a launch of the Java applet is no too short. Many people are thinking falsely that it is caused by downloading the applet from the internet to user's computer. In this time a connection to the Internet is reasonably fast so the part that takes the longest time is verifying code.

The Class Loader

The Class Loader is used to load the Java code from the network. Class loaders perform two functions. First, class loader finds the bytecode which VM needs to load for a particular class of the applet and it undergoes to the bytecode control by the Verifier. Second, class loader defines the namespaces seen by different classes and how those namespaces relate to each other. Problems with namespace management have led to a number of serious security holes.

There are two basic varieties of class loaders: *Primordial Class Loaders* and *Class Loader objects*. The Primordial Class Loader is only one and it is essential part of the default JVM. It cannot be overridden. Classes loaded by the Primordial Class Loader are not subjected to the Verifier prior to execution. Class Loader objects load classes that are not needed to bootstrap the VM into a running Java environment. Classes loaded through Class Loader objects are considered as untrusted by default. The Applet Class Loader is the most important part of the Java security model [8].

Another component of Java security is the way Java classes are loaded over the network. The java.lang.ClassLoader class defines how this is done. Applet viewers and Web browsers create subclasses of this class that implement security policies and define how class files are loaded via various protocols.

The Security Manager

The third part of the base Java security model is the Security Manager. The job of the Security Manager is to keep track of who is allowed to do which dangerous operations. A standard Security Manager will disallow most operations when they are

requested by untrusted code, and will allow trusted code to do whatever it wants.

The Security Manager is a single Java object that performs runtime checks on dangerous methods. The Security Manager can veto the operation by generating a SecurityException. It makes the final decision as to whether a particular operation is permitted or rejected. When a dangerous call is made to the Java library, the library queries the Security Manager. These queries use a set of methods that check access. The Security Manager is installed in each JVM only once.

Responsibilities of the Security Manager are that it prevents installation of new class loaders; it protects threads and thread groups from each other; it controls the execution of other application programs; it controls the ability to shut down the VM; it controls access to other application processes; it controls access to system resources such as print queues, clipboards, event queues, system properties, and windows; it controls file system operations such as read, write, and delete; it controls network socket operations such as connect and accept; it controls access to Java packages, including access to security enforcement classes [8].

The CodeSource

The CodeSource encapsulates the code's origin, which is specified as an URL, and the set of digital certificates containing public keys corresponding to the set of private keys used to sign the code.

The AccessController

The java.security.AccessController class is used for deciding whether access to a critical system resource should be allowed or denied, based on the security policy currently in effect; for marking code as privileged, thus affecting subsequent access determinations; for obtaining a snapshot of the current calling context, so access-control decisions from a different context can be made with respect to the saved context.

C. Types of Security Attacks

Java applets can be subjected to potential hacker attacks. Now follows a brief description of four of many possible attacks, namely - attacks that deny legitimate use of the machine by hogging resources, - attacks that modify the system, - attacks that invade a user's privacy, - attacks that antagonize a user and attacks called Zero-day attacks [11].

Denial of Service Attacks

A Denial of Service Attack abbreviated DoS attack, is an attempt to make interrupt or suspend services of a host connected to the Internet.

The DoS attack belongs to less seriously attacks because this attack consumes only system resources and can slow your computer or your network connection considerably. For example, attacks may involve completely filling a file system, hogging all possible screen space, allocating all of a system's memory, creating many high-priority threads, or using all available file pointers [10].

The Java sandbox does not protect against DoS attacks. Therefore, we should not allow applets which come from suspects and unverified sources [8].

System Modification

This type of attack is the most severe class of attacks. It is able to intrusion into the system itself and consequences of these attacks can be critical and dangerous. Applets that implement such attacks are attack applets.

System modification attacks may modify the contents of memory, create, modify, or delete files, directories, database entries, kill processes or threads and install malware as is Trojan horse, worm, back door and many more to the user computer.

Invasion of Privacy

This type of the attack includes disclosing information about a user or host machine that should not be published. The forging mail can also be perceived as an attack to privacy. Especially if we value our privacy or we have some confidential files on the computer, this attack may be a very annoying issue.

Antagonism

The most commonly we encounter with this type of attack. There are attacks that merely antagonize or annoy a user. For instance, antagonism applets are able to play unwanted sound files, open a large number of new windows simultaneously, etc.

Zero-day attacks

Zero-day vulnerabilities in computer science marking attack or threat that tries to exploit computer vulnerabilities refer to a hole in the software that is unknown to the vendor. This security hole is then exploited by hackers. Zero-day here does not indicate the number or the number of days, but the fact that the user is at risk, until a patch is still in the works to fix bugs (i.e., the zero day).

There were mentioned few of many potential security attacks. All these attacks could be realised by using Java applets and this is seriously threat for users. The Java security model was designed to thwart those threats perceived to be the greatest dangers. The Oracle continuously improves the Java security because new attacks and security holes are discovering. This is the way to protect against hostile applets.

Let's now look at specific examples of security attacks. Attacks are listed chronologically by date of publication.

Targeted attacks to chemical companies, named *The Nitro Attacks*, started in **July 2011** and continued into September 2011. The 29 companies in the chemical sector and another 19 in various other sectors, primarily the defense sector were affected by this attack. The attackers first researched desired targets then sent two types of mail. If the mail was targeted to a specific recipient it contained invitation to meeting. Or if the mail was being sent to a broad set of recipients the email claims to be an update for some piece of commonly installed software. In both cases the mail contained attachment with a malicious Trojan called PoisonIvy which was included in a zipped file with the password. The PoisonIvy allowed access to other computers in the company workgroup. They could copy the content, and upload the information to servers external to the compromised organization. The purpose of the

attacks was likely industrial espionage, and the attackers appear to have been seeking intellectual property, including design documents, formulas, and manufacturing processes, for competitive advantage. The source of the attack was identified as a computer system that was a virtual private server (VPS) located in the United States. The system was owned by an individual named Covert Grove from China [12].

An indisputable advantage of the Java language is a cross-platform but this can easily be exploited to an attack as a good example is following Java applet malware. In **April, 2012** the infection volume is reported at over 600,000 computers with Mac and Windows operating systems. This threat proceeds in three phases. First, Java Applet malware is loaded. Second, if the threat is running on a Mac operating system, it downloads a dropper type malware written in Python. Or if the threat is running on a Windows operating system, it downloads a standard Windows executable file dropper. This Trojan only checks whether it is the required operating system. Finally, one of two back door Trojans depending in OS is dropped on to the computer. These Trojans can download files, open a remote shell, upload files, send information about CPU details, disk details, memory usage, etc. [13].

July, 2013 Symantec published a campaign is targeting government agencies by sending phishing emails with a malicious attachment in the form of the Java remote access Trojan (RAT). Cyber criminals use recent hot media topics to entice users. In this case they used the news coverage surrounding the NSA surveillance program PRISM. The phishing email contains two legitimate non-malicious PDF documents and one Java file named US National Security State. The most of targets were located in the United States [14].

July, 2013: The target of the attack could also be the SIM card (Subscriber Identification Module) which is present in all mobile phones. Every smart card is responsible for the unique identification number known as the IMSI (International Mobile Subscriber Identity) and also for handling the encryption when communicating with the telephone network. If an attacker sends a cleverly crafted silent binary SMS update message over-the-air (OTA) to the mobile phone although the device will refuse the unsigned message but it will answer with an error code signed with the 56-bit DES private key. After the private key is deciphered, an attacker can sign malicious software updates and send them through OTA updates to the mobile phone. For instance, malicious applets can send premium text messages, reveal the geo-location of the device or misuse of payment systems [15].

November, 2013 Symantec has discovered a new back door worm-type threat which targets servers running Apache Tomcat. The Apache Tomcat is an open source web server and servlet container developed by the Apache Software Foundation (ASF) and based on Java language. Servers are quite valuable targets, since they are usually high-performance computers. Back door type Trojan horses and worms enable the attacker to control a computer remotely. Infected computers could be used to attack other Tomcat servers by sending the malware to them and this would lead to DoS attacks [16].

In early **January 2014**, on a website of a major Japanese book publisher and distributor, of books, magazines, comics, movies, and games was encountered a malicious iframe leading to another website hosting an exploit kit. This malware is intended for the purpose of stealing information and located in Japan. The malware monitors open windows for two online banking sites, three online shopping sites, three Web mail sites, three gaming/video websites and fourteen credit card sites. Most providers of these sites are aware of the security risks and have implemented additional layers of protection and verification for their online customers. It is not surprising that contested pages belonged to regional firms that had not done sufficient safety measures [17].

February, 2014 a new spam campaign appeared in the form of the Java remote access Trojan (RAT) known as JRAT. The spam email's sender claims that they have attached a payment certificate to the message and asks the user to confirm that they have received it. An attachment with the file name Paymentcert.jar was actually a malicious Trojan virus. The Windows, Linux, Mac OS X and Solaris computers are threatened by RAT. This attack affected the most individuals in the United Arab Emirates and the United Kingdom [18].

D. A Chronology of Security Holes

In this chapter will be described chronologically security holes in the Java and ways to address them. As mentioned Java programs run in the sandbox which is responsible for a security. Since Java is so widespread is not surprising that there are constantly incidence of security breaches and are used to hacker attacks. Therefore, the Oracle is continuously working on improving its product. The Critical Patch Updates (CPU) are published on their web every 3 months starting from January. Exceptions are updates that needed to be done immediately, given the severity of the security risk. In addition the CPU are published on the website the Common Vulnerabilities and Exposures (CVE). Common Vulnerabilities and Exposures is a dictionary of common names for publicly known information security vulnerabilities. The syntax is CVE-YYYY-NNNN, where Y is a year and N is any number. CVE includes just one vulnerability and makes it easier to share data across separate network security databases and tools, but where CPU can contain multiple CVE.

In the fig. 2 can be seen the bar graph of the number of vulnerabilities (Security Alert, Critical Patch Update Advisory or Security Alert) by year, based on data mentioned by Oracle. The graph includes data to the April 18, 2014 [19], [20].

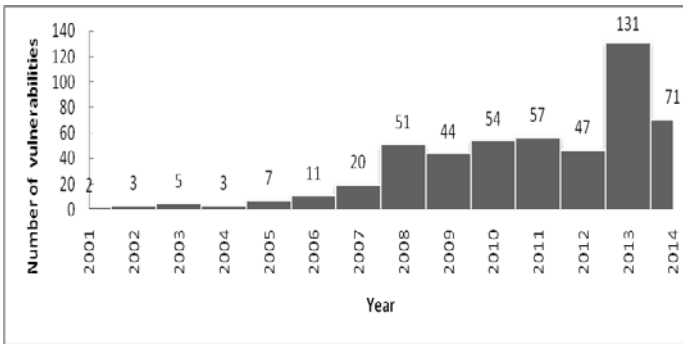

Fig. 2 Graph of the number of vulnerabilities by year [19]

It can be seen that with increasing numbers of users and extension of the Java is also an increasing number of security vulnerabilities discovered because hacker attacks occur more frequently.

The following table lists Critical Patch Updates with description of some vulnerabilities. Data are displayed in descending order by date of the publication.

Table I. List of Critical Patch Updates

Publish Date	Number of security fixes	Description
2013-June-18	40	34 vulnerabilities affect client deployments. 4 vulnerabilities can affect client and server deployments. 1 vulnerability affects the Java installer and can only be exploited locally. 1 vulnerability affects the Javadoc tool and the documents it creates.
2013-April-16	42	39 vulnerabilities may be remotely exploitable without authentication 2 fixes are applicable to server deployments of Java
2013-February-19	5	Special Update which contains additional fixes. All of these vulnerabilities may be remotely exploitable without authentication;
2013-February-01	50	The original CPU was scheduled on February 19th, but Oracle decided to accelerate the release of this CPU. 44 vulnerabilities only affect client deployment of Java (e.g., Java in Internet browsers); 1 vulnerability affects the installation process of client deployment of Java; 3 vulnerabilities apply to client and server deployment of Java; 2 vulnerabilities apply to server deployment of the Java Secure Socket Extension (JSSE);
2012-October-16	30	29 of these vulnerabilities may be remotely exploitable without authentication
2012-June-12	14	12 vulnerabilities may be remotely exploitable without authentication.
2012-February-14	14	All of these vulnerabilities may be remotely exploitable without authentication; Out of the 14 vulnerabilities, 6 affect server deployments of Java SE, including the vulnerability in the Lightweight HTTP server.
2011-October-18	20	19 vulnerabilities may be remotely exploitable without authentication. 6 vulnerabilities are applicable to JRockit. 1 vulnerability is for the “BEAST” exploit. “BEAST” (Browser Exploit Against SSL/TLS) can potentially provide a malicious hacker the ability to bypass SSL/TLS encryption and ultimately decrypt potentially sensitive web traffic.
2011-June-07	17	5 vulnerabilities apply to client and server deployments of Java SE. These vulnerabilities can be remotely exploited by supplying malicious data to APIs in the affected component of the server. 11 vulnerabilities apply to client deployments of Java SE only. These vulnerabilities can only be exploited through untrusted Java Web Start applications and untrusted Java applets. 1 vulnerability applies to server deployments of Java SE only. This vulnerability can

		only be exploited by supplying malicious input to APIs in the specified Component. All of these vulnerabilities may be remotely exploitable without authentication
2011-February-15	21	21 new security fixes across Java SE and Java for Business products. 13 vulnerabilities affect Java client deployments. 12 of these 13 vulnerabilities can be exploited through Untrusted Java Web Start applications and Untrusted Java Applets. One of these 13 vulnerabilities can be exploited by running a standalone application. 3 vulnerabilities affect client and server deployments. 3 vulnerabilities affect Java server deployments only. These vulnerabilities can be exploited by supplying malicious data to APIs in the specified Java components. 1 vulnerability is specific to Java DB, a component in the Java JDK, but not included in the Java Runtime Environment (JRE). 19 vulnerabilities may be remotely exploitable without authentication
2010-October-12	29	29 new security fixes across Java SE and Java for Business products. 28 vulnerabilities may be remotely exploitable without authentication
2010-Mar-30	27	27 security fixes for Java SE and Java for Business releases. All of these vulnerabilities may be remotely exploitable without authentication

These vulnerabilities may be remotely exploitable without authentication. This means that successful attack of these vulnerabilities can result in unauthorized update, insert or delete access to some Java Runtime Environment accessible data as well as read access to a subset of Java Runtime Environment accessible data and ability to cause a partial denial of service (partial DOS) of Java Runtime Environment, can result in unauthorized Operating System hang or frequently repeatable crash (complete DOS) or can result in unauthorized Operating System takeover including arbitrary code execution [19].

IV. Alternatives for Security Improvement

Next, let us analyze alternatives of programming environments with which it is possible to replace control Java applets of experiments.

A. JavaScript Language

JavaScript was developed by Brendan Eich while working for Netscape Communications Corporation. It was first introduced in 1995 under the name of LiveScript. However, due to marketing purposes it was renamed JavaScript. JavaScript's official name is ECMAScript, which is developed and maintained by the ECMA (European Computer Manufacturer's Association) International organization. Today, JavaScript is a trademark of Oracle Corporation.

JavaScript (JS) is scripting dynamic and object-oriented language designed primarily for adding interactivity to Web pages and creating Web applications.

A common misconception is that JavaScript is similar or closely related to Java. There are few similarities between Java and JavaScript. For instance, both have a C-like syntax. They are both object-oriented and typically sandboxed. Also, JavaScript was designed with Java's syntax and standard library in mind. JavaScript's standard library follows Java's naming conventions, and JavaScript's Math and Date objects are based on classes from Java 1.0, but similarities end there.

Key differences between Java and JavaScript:

- Java is an OOP programming language while JavaScript is an OOP scripting language.
- Java creates applications that run in a virtual machine or browser while JavaScript code is run on a browser only.
- Java code needs to be compiled while JavaScript code is all in text.
- Java is loaded from compiled bytecode; JavaScript is loaded as human-readable source code.
- They require different plug-ins.
- Java has an implicit this scope for non-static methods, and implicit class scope; JavaScript has implicit global scope.
- Java has static typing; JavaScript's typing is dynamic. In dynamic typing, a variable can hold an object of any type and cannot be restricted [21].

JavaScript *can* perform the following operations:

- It has access to the computer's clock and can pull the appropriate data.
- It can collect information about the browser.
- It can be used to validate forms data or input.
- It can create cookies.
- It can manipulate the Document Object Model (DOM) where the HTML DOM is the official W3C standard for accessing HTML elements.
- It can create various dynamic HTML elements, such as it can interactive change all the HTML elements, all the HTML attributes, CSS styles, can remove, add new HTML elements and attributes, can react to all existing

HTML events and can create new HTML events in the page.

JavaScript *cannot* perform the following operations:

- It cannot write to files on the server without the help of a server side script. JavaScript can send a request which can read a file but it cannot write to a file unless the file called on the server actually runs as a script to do the file write for you.
- It cannot access databases unless you use Ajax and have a server side script perform the database accesses for you.
- It cannot read from or write to files in the client. The only exception to this are cookies.
- It cannot close a window if it didn't open it.
- It cannot access web pages hosted on another domain.
- JavaScript cannot protect your page source [21].

JavaScript is among the top ten most popular and most widely used programming languages in the world. According to the TIOBE index JavaScript is placed in the ninth place in front of it are placed languages in the following order C, Java, Objective-C, C++, C#, (Visual) Basic, PHP, Python.

The TIOBE Programming Community index is an indicator of the popularity of programming languages. The ratings are based on the number of skilled engineers world-wide, courses and third party vendors. Popular search engines such as Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube and Baidu are used to calculate the ratings [22].

The unquestionable advantages of JavaScript are that JavaScript code can be run on the web, on computers, servers, laptops, tablets, smart phones, and more, JavaScript is a relatively easy language and simple to learn, JavaScript is very fast because any code functions can be run immediately instead of having to contact the server and wait for an answer. JavaScript runs on an Internet browser therefore the users do not need special software or downloads to view it.

Disadvantages include security issues as with all programming languages. JavaScript, providing a high degree of functionality at the expense of security. JavaScript can easily be used to carry out denial of service and invasion of privacy attacks. For instance JS can track a surfer's history, secretly keeping tabs on all sites visited by a user and reporting back to a collection site, read directory listings, learning about a Web surfer's file system and reporting back to a collection site, steal files, mailing the stolen goods back to an attacker, etc. [8].

B. Dart Language

Dart is an open-source Web programming language which was developed by Lars Bak and Kasper Lund who work in the Google. It was first introduced in October 2011 at the GOTO conference in Aarhus. Dart is a class-based, single inheritance, object-oriented language with C-style syntax.

Dart was developed to eventually replace JavaScript. Until then, in order to run in mainstream browsers, Dart code is compiled into JavaScript using the dart2js compiler. This resulting code is compatible with all major browsers with no specific browser adoption of Dart being required. Code written in Dart can, in some cases, run faster than equivalent code hand-written using JavaScript. Dart code can be run without compilation to JavaScript if we use the Chromium web browser that includes the Dart VM, called Dartium. Dartium is intended as a development tool for Dart applications and it should not be used as a primary browser.

In order to completely replace the JavaScript it is necessary that Dart virtual machine placed in browsers. But now creators of the Dart encounter with criticism and negative feedback. In any case, it is only a matter of time before a Dart finds its place among programming languages. As is known Google is among one of the largest corporations which its products constantly improves and develops. Thus Dart has great potential for the future [23].

A. ActiveX Environment

An ActiveX is not a programming language it is a set of technologies and tools for Internet Explorer developed by Microsoft in 1996. It is based on two earlier Microsoft technologies called Component Object Model (COM) and Object Linking and Embedding (OLE).

An ActiveX control is a small program similar to a Java applet. It can be created in any programming language that recognizes Microsoft's Component Object Model such as C, C++, Visual Basic, Java and more. ActiveX control is a control using Microsoft ActiveX technologies which are commonly used in Windows operating system.

The ActiveX approach relies on digital signatures, ActiveX controls can be digitally signed by a developer, distributor, certifier or someone who vouches for the code. ActiveX controls have almost unlimited access to the operating system. With this function comes a certain risk that it may damage software or data on your computer.

Because there was a large amount of harmful ActiveX controls, Internet Explorer version 7 and above displays a warning every time a site attempts to use an ActiveX control. It is up to the user to decide whether or not the request comes from a trustworthy source. Either you trust the code completely and allow it to run unhampered on your computer, or you do not.

Compared with Java applets, ActiveX control has two main disadvantages. First, Java has the ability to run untrusted code fairly safely due the sandbox. Second, Java applets can be run on all platforms, whereas ActiveX controls are a priority limited to Windows environments. Web browsers like Google Chrome, Safari and Firefox do not support ActiveX controls by default due to security concerns. Users of these browsers can download extensions through which ActiveX will be run. But there are not extensions for all environments [24], [25].

V. Discussion and Conclusions

In this paper, we deal with the security issues, namely security of Java applets and options for their replacement by other tools. We may conclude that each of the aforementioned languages have their advantages and weaknesses. Some are

more appropriate for our purposes than others. Due to the security measures of Java applets, which the Oracle Corporation develops, the formation of the control web page of remote experiments is very limiting for us. The user is also constantly harassed by the authorization to run of applets and by the reinstalling the JVM on his/her computer. With each new release of Java, a number of new serious security holes have been discovered. Every new feature introduces new security holes. Consecutively, we decided to abandon the concept of making experiments using the Java, and we were looking for new acceptable alternatives. We reviewed some programming languages and we take into consideration their functionality, limitations, simplicity, and main assumptions of the usability today and in future too. Since remote experiments are made available through the website to users, we focused on languages that can create elements which are executable on web browsers. The user is not exposed to a duty to install any hardware and code is running under different operating systems.

As was mentioned earlier Dart has great potential but is yet at an early stage when it slowly gets into the consciousness of programmers and the general public. Dart programming language, we have not opted for our purposes solely because it is relatively new language, and if there were any problems it would not be easy to solve because there are still scarce professional resources. And also we did not want to risk the possibility that it completely disappeared.

ActiveX is not suitable for our purpose because it is dependent on the platform used and we want to avoid platform dependency. Our aim is to create website for remote controlled experiments that will be user friendly, accessible, and making no demands on the software. Thereby ActiveX can be excluded.

The optimal solution seems to be the JavaScript. It needs less programming skills and it enables easy implementation to the web page of the experiment. JavaScript widgets are the most often offered on the Net as completely functioning items which we can simply grabbed from the web and be used for your purposes. It has been in use already a couple of decades and it looks like it will be for a long time. Use of JS is advantageous over Java applets in that it is much easier and more robust language, it can also be run on mobile phones, tablets, etc. JavaScript code runs faster than Java code because it does not need much time for compilation and security checking. JavaScript code is also smaller than the bytecode file.

ACKNOWLEDGMENT

The paper was published thanks to the Grant of the Internal Agency of UTB No IGA/FAI/2014/034. One of us acknowledges the partial support of the Slovak Research and Development Agency, project no. APVV-0096-11, the Scientific Grant Agency VEGA, project no. 2/0157/12, and the KEGA Agency projects No 011TTU-4/2012 and 020TTU-4/2013.

REFERENCES

[1] F. Schauer, F. Lustig, J. Dvořák, M. Ožvoldová, " Easy to build remote laboratory with data transfer using ISES—Internet School Experimental System," in *European Journal of Physics*, vol. 29, 2008, pp. 753--765, ISBN 978-0-9741252-9-9.

[2] F. Lustig, "Jak si jednoduše postavit vzdálenou laboratoř na internet (How to simply build a remote laboratory on the Internet)," in *Veletrh nápadů učitelů fyziky (Exchange idea shop of Physics teachers)*, 2004, Brno, pp. 9 -- 19.

[3] F. Schauer, M. Ožvoldová, F. Lustig, "Integrated e-learning - new strategy of the cognition of real world in teaching," in *World Innovations in Engineering Education and Research iNEER*, USA, 2009, pp. 119—135.

[4] M. Krbeček, F. Schauer, K. Vlček, "Communication Requirements of Laboratory Management System," in *LATEST TRENDS on SYSTEMS* - VOLUME II: Proceedings of the 18th International Conference on Systems (part of CSCC '14), Santorini, Greece, 2014, pp. 686-691. ISBN 978-1-61804-244-6, ISSN 1790-5117.

[5] R. Hamid, M. SyakirahAfiza, "Remote Access Laboratory System for Material Technology Laboratory Work," in *International Conference on Engineering Education and International Conference on Education and Educational Technologies*, Proceedings, 311–16.

[6] A. S. Drigas, J. Vrettaros, L. G. Koukianakis, J. G. Glentzes, "A Virtual Lab and E-Learning System for Renewable Energy Sources," WSEAS Transactions on Computers 5 (2): 337–41, 2006.

[7] Internet world stats. Available: http://www.internetworldstats.com/emarketing.htm

[8] G. McGraw, E. Felten, Securing JAVA – Getting down to business with mobile code. John Wiley & Sons, Inc. 1999.

[9] Oracle Corporation. Available: http://docs.oracle.com/

[10] D. Flanagan, Java in a Nutshell. 2nd Edition, pp. 628, 1997.

[11] M. Gerža, F. Schauer, R. Jašek, "Security of ISES Measureserver® Module for Remote Experiments against Malign Attacks," in *International Journal of Online Engineering (iJOE)*. Vol 10, No 3. 2014, pp. 4-10.

[12] E. Chien, G. O'Gorman, The Nitro Attacks - Stealing Secrets from the Chemical Industry. Technical report, 2011.

[13] T. Katsuki, Both Mac and Windows are Targeted at Once. Symantec Corporation, Official Blog, 2012.

[14] A. Lelli, Rise of the Java Remote Access Tools. Symantec Corporation, Official Blog, 2014.

[15] C. Wueest, Hijacking SIM Cards through Over-the-Air Updates. Symantec Corporation, Official Blog, 2013.

[16] T. Katsuki, All Your Tomcat Are Belong to Bad Guys? Symantec Corporation, Official Blog, 2013.

[17] Symantec Security Response: Popular Japanese Publisher's Website led to Gongda Exploit Kit. Symantec Corporation, Official Blog, 2014.

[18] L. Payet, JRAT Targets UK and UAE in Payment Certificates Spam Campaign. Symantec Corporation, Official Blog, 2014.

[19] Oracle Corporation, Official website. Available: http://www.oracle.com/technetwork/topics/security/whatsnew/index.html

[20] Oracle Corporation, Official Java website. Available: https://www.java.com

[21] S. Chapman, What Javascript Can Not Do. Available: http://javascript.about.com/

[22] TIOBE Company: TIOBE Index for June 2014. Available: http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

[23] Dart. Available: https://www.dartlang.org

[24] D. Roos, How ActiveX for Animation Works. Available: http://entertainment.howstuffworks.com/activex-for-animation1.htm

[25] Microsoft, Protect yourself when you use ActiveX controls. Official webpage. Available: http://www.microsoft.com/security/default.aspx