Security Issues Related To E-Learning Education

Andrei Marius GABOR, Marius Constantin POPESCU, Antoanela NAAJI,

"Vasile Goldis" Western University of Arad
Arad, Romania, Computers Science

Summary
E-learning is the answer to the new challenges of online education, involving the interaction between teachers or trainers and learners. An e-learning platform is a web-based software application, depending on the utilized technology and Internet environment. Due to the close connection between e-learning platforms and the web, platforms are threatened and vulnerable to various cyber-attacks. As a measure to block unauthorized access to e-learning systems, choosing a strong password is a first protective obstacle. The paper is focused on security issues in e-learning platforms and proposes an algorithm that generates strong passwords to secure personal data, while offering the possibility to check the strength of the generated password.

Key words:
brute force attack, E-leaning, security, Moodle, vulnerabilities.

1. Introduction

E-learning is an alternative to traditional courses, where the interaction between teachers and learners is direct. It should be mentioned that e-learning does not replace traditional education, as the two are often found in collaboration. E-learning encompasses all educational activities of an individual or groups of individuals working online or offline, synchronously or asynchronously, communicating through various devices connected to the internet [1].

The LMS (Learning Management System) is an online learning platform involving students, teachers and administrators. The facilities of such a system include delivering educational content to students, online evaluations and class books, communications tools such as forum and blog, multimedia tools such as podcasts, audio-video conferences, MOOC video courses, access accounts, work groups, etc.

Implementation and use of such systems involves certain security problems, issues related to personal data protection, as well as content-related vulnerabilities. Moreover, e-learning systems may be vulnerable to phishing and spyware attacks [2], by which the attacker can take hold of users' login data (credentials). After obtaining credentials, attackers can easily access servers and, implicitly, access to content. One of the most important countermeasures against cyber-attacks is the security of the authentication process. The stronger the password is, the more protected the e-learning platform is against malicious software or hackers. Passwords should be hard to guess or hack, for all accounts created on the platform [3]. In case of educational platforms, administrators may require the use of strong passwords, since in recent years cyber-attacks have increased in number. Thus, in August 2014 1.2 billion passwords were hacked, from a number of 420,000 websites [4], a phenomenon that has caused an average worldwide loss of 3 million euros/company in 2014. In 2015, the number of those who fell prey to identity thieves grew by one third in the UK, whereas in the USA hackers have managed to obtain the personal data of over 100,000 citizens, directly from the IRS website. Personal data can be used to open new accounts, commit crimes under an assumed name, obtain security data from bank cards etc.

According to a report compiled by Cifas [5], 80% of identity thefts occur online, causing damages worth billions of sterling pounds annually. Criminal groups have become so well organized that they have forums where they sell data stolen from tens of thousands of victims. Creating new credit cards in the name of another person, the most common fraud method (41%), is followed by opening new bank accounts (27%). In the USA, 104,000 taxpayers' identities have been used by hackers to obtain tax returns in the name of other people, through a "Get Transcript" application found on the website, and used to fill out tax return forms in the name of other people. With the expansion of the BYOD (Bring Your Own Device) trend, by which more and more employees bring their mobile device to work, as well as access to educational platforms from mobile devices, finding secure connection solutions without impacting productivity is becoming even more important for societies. Losses calculated in the USA for a company are around $201 per client, for each cyber-attack, which translates into an overall average loss of $6 million for each company [5]. One study underlined the general indifference of employees towards the employer's data, showing that one employee out of seven was willing, if the occasion presented itself, to sell their passwords for at least €130 [4]. The purpose of the paper is to describe an efficient solution to enhance security in e-learning systems by introducing new additional policies related to setting a password. This concerns privacy and personal

Manuscript received January 5, 2017
Manuscript revised January 20, 2017

data security policies and describes how personal information must be treated when educational platforms are used.

2. Security and vulnerabilities in e-learning

During the development of educational systems, emphasis was placed more on developing courses and on how they are delivered, and less on user security and privacy. Most of the vulnerabilities of e-learning systems are the same as those encountered by web-based applications. Compromising user privacy can have negative consequences on the social and financial situation, as described in the previous paragraph, when the attacker has access to personal identification data. User privacy is vulnerable, especially due to failure to apply security policies in e-learning systems. Another aspect concerns content integrity [6], with the result of intellectual property rights infringement, through publication of dissemination of the entirety or parts of educational course content, even if they are copyright-protected. E-learning systems are web-based applications, so that they inherit all vulnerabilities of web-based applications. A better understanding of security issues, as well as means of protection, will help users avoid certain security threats. In case of vulnerabilities in educational platforms, attackers can consider various types of attacks described in Table 1.

TABLE 1: Types of Attacks in E-learning and Solutions

Vulnerabilities	Solutions
- Brute force attack - Dictionary attack - Hybrid attack - SQL Injection - XSS Injection - Cross Site Request Forgery (CSRF) - Session Hijacking - Stack-smashing attacks	-Improving login methods, authorization, privacy - Long passwords -Periodically changing passwords -Randomly generated passwords, as long and as complex as possible

Any password can be decrypted, but the time it would take to do this depends on the content of the password. Thus, if a sophisticated password is chosen, the time of success is in the range of years. A strong password must not be communicated to other people [7] and contains at least eight characters [8], without containing a whole word, the user name, or the company name. Also, it should be significantly different from previous passwords, and contain both upper case and lower case letters, numbers and keyboard symbols (all characters on the keyboard not defined as letters or numbers) or spaces. The password should not contain the user's name, birthday, address, and it should not be stored on the computer, saved in the browser, sent by e-mail, Skype, messenger, etc. It is possible for a password to check all the above-mentioned criteria and be weak. The following steps are recommended to memorize the password:

- creating an acronym based on easy-to-remember information, such as choosing a meaningful phrase, such as children's birthdays;
- replacing letters or words in an easy-to-remember phrase with numbers, symbols and misspellings;
- associating the password with a hobby or favorite sport.

Currently, the online security standard requires using a different password for each website where the user has an account, as well as strong passwords of at least 8 characters, containing letters and numbers (at least one capital letter), for added security, but also special characters. To secure all data there must be as many and as well thought-out passwords as possible. Likewise, certain rules should be considered when storing passwords on websites [9]:

- not storing passwords in text files – passwords are saved in the database as they are and can be read by anyone;
- not using only encryption – when passwords are encrypted and several people use the same password it can be found out;
- not using only *hashing*, which is a one-way function converting the password into a random string of characters, although there are "Rainbow Tables" uncovering the password hidden behind that hash.

In this sense, hashing and salting should be used, meaning the password should go through that one-way function, but also combined with a random set of user-specific characters (salting). Thus, if the password and the salt are sufficiently complicated, no one will be able to uncover the password. After creating a secure password, the user will be able to check through specialized programs or websites [10, 11], providing additional information on the time it takes a hacker to break into accounts using a dedicated application. Researchers from Microsoft Research launched the "Telepathword" application that works using a database comprising the most common password and habits of setting. The application detects the vulnerability of a password, guessing the next symbol while typing [12]. Other programs that calculate password strength [13] determine the number of years it would take a hacker to break a password depending on the length of the password and characters used, taking into account the "brute force attack" method [14], which is only used in worse case scenarios, when all other methods fail, as it is the slowest method, but covers all possibilities. The "brute force" technique uses 92 characters consisting of the 26 letters of

a keyboard, lower case and upper case (52 in total), numbers from 0 to 9 and all symbols and punctuation marks (32). When the hacker knows the length of the password, they can guess it out of a limited number of attempts, approximately 3x1015, but if they do not know the length of the password, the number of attempts, although limited, is extremely large. Nevertheless, most passwords are found out through "social engineering" rather than multiple attempts. The simple password hacking method is using software that records key press in password fields. Thus, a motivated hacker can find users and passwords on an unprotected torrent site, with a minimum amount of effort. The mathematical relation used to hack passwords is the one used to calculate permutations [15]:

$$n!/(n-k)! \quad (1)$$

where n is the number of characters included in the password, and k is the length of the password. The highest the value of uncertainty of password H, expressed in bits,

$$H=\log2(nk), \quad (2)$$

the stronger the password is.
Consequently, the longer the password (at least 8 characters), the harder it will be to determine, the odds of "breaking" it being 10-16. However, in the case of a supercomputer, for example, the types used in medicine to decode DNA, which reaches 36.01 trillion operations per second, breaking a password using the brute-force attack method becomes a matter of seconds, minutes, or hours. Computing power is well known in bitcoin mining, being estimated on the market to be somewhere around 64exaFLOPS (1exaFLOPS=1018 operations per second), thus the problem of breaking an easy-to-remember password becomes non-existent. Distributed computing uses the Internet to link personal computers in order to obtain more FLOPS. Another technique to learn a password involves network knowledge and consists in tracking traffic in the internet network, so as to determine the actual content of information in the computer of the defrauded person/company.

3. Security in Moodle

Moodle – Modular Object-Oriented Dynamic Learning is an open source platform, created and developed in Australia by programmer Martin Dougiamas, in 1999. It features several advantages compared to other similar educational platforms, as it can be easily installed and benefits from a large community of users and developers [16].
Also, it has the best communications tools, information is accessible, the source code is written in PHP, and the databases it supports are MySQL and PostgreeSQL. It offers the possibility to create lessons, courses, tasks, and to post tests for checking. It brings modules to be used by learners, such as chat, forum, or polls. One possible type of attack in the Moodle educational platform [17] can be a brute force attack. To prevent this type of attack, Moodle has added a password setting policy system – Site administration → Security → Site policies (Fig.1).

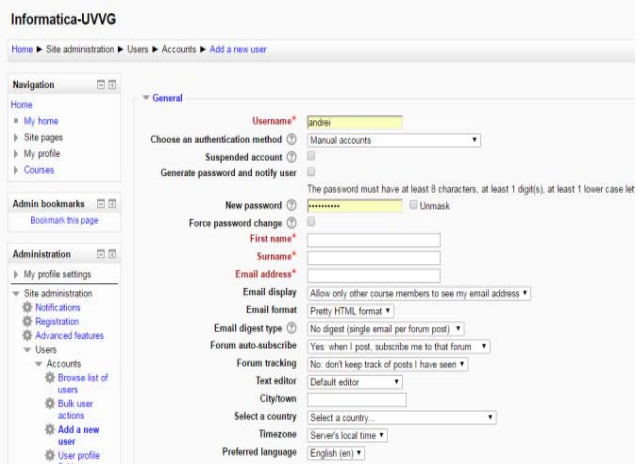

Figure 1. Setting a password for Moodle users

The algorithm proposed in this paper can be useful in preventing brute force attacks, and can be integrated in the user ID creation page, where the administrator can set the password security policy directly in the page.

4. Algorithm Implementation Steps

4.1 Generating Password

Generating password is the starting part of the algorithm and represents a randomize process. In most situations, users create their own passwords, but there are cases when the password must be generated automatically. The most relevant example is when the user forgot their password and must reset the old password and temporarily replace it with a *random* one. One of the benefits of a randomly generated password lies in it being far more difficult to guess than a user-defined one. The character set of a randomly generated password contains both lower and uppercase letters, and numbers and symbols. Known character sets are used to generate the password, which are introduced into a string through the implode function. This

method makes it easier to work with and configure those strings.

```
$lower='abcdefghijklomnopqrstwvxyz';
//lower case characters
$upper='ABCDEFGHIJKLMNOPQRSTWVXYZ';
//upper case characters
$numbers='0123456789''//numbers
$symbols='!@#$%^&*-?''//symbols
$lower=implode(range('a','z'));
$chars=$lower.$upper.$numbers.$symbols;//
concatenate all 4 strings
```

The random selection of a character from the string of characters is done with one of the following functions: `rand()`,`rand_mt()`or `random_int()`.The `rand()` function has two arguments, the minimum value and the maximum value, and returns an integer between the two values, including those minimum or maximum values. The `rand_mt()` function uses a "Mersenne Twister" generator, and replaces the `rand()` function, being 4 times faster and more `random` than `rand()`.If the intention is to generate several secure passwords at the same time, they will not be perfectly secure, since the server time is used in the program calculating random, so that it is possible for the hacker to manage to find a pattern, after which it is fairly easy to hack the password. The `random_int()` function is used in such situations. The only problem with this function is the fact that it requires the PHP 7 variant, or newer:

```
$i=mt_rand(0,strlen($string)-1);

//$i=random between the value 1 and the
length of the string of characters formed by
concatenating the 4 strings
```

The `random` creation of an array of characters with a random length is given by the:

```
random_string($length,$char)
```

function, where `$length` is the length of the string, and `$char` is the selected string of characters:

```
function random_string($length,$char_set)
{ $output='';
  for($i=0;$i<$length;$i++)
  {$output.=random_char($char_set);}
return $output;
```

The password is generated depending on the desired length and the desired string of characters, which will include the password:

```
function generate_password($length) with
return random_string($length,$chars);
```

If we want the password to not contain any upper case letters then:

```
$use_upper=false
```

The same is true for the 3 other strings of characters.

4.2 Password fluency

Simple positive words are used to create the dictionary and to remember a password as easily and possible. The dictionary thus created is based on such words. Passwords are not secure because hackers use such dictionaries for the very purpose of reducing the workload of the "brute force attack" method, by employing words from different concatenated dictionaries with numbers and symbols. Words are selected from the dictionary by introducing positive words into a .txt file, each word on a row. There will be two files, one with random positive words, and another with words representing a brand. The file is read in PHP with the *file* function, as it reads the entire file, it is then divided into an *array*, and each line in the file becomes an element of that *array*. Thus, it is far easier to choose an element from that *array*:

```
function read_dictionary($filename="") {// can
use full path or relative path
$dictionary_file="dictionaries/{$filename}";
  return
file($dictionary_file,FILE_IGNORE_NEW_LINES|FILE
_SKIP_EMPTY_LINES);}
```

The `random` *strings* composing the friendly password are created through a series of functions. The functions return *random* symbols, numbers and words depending on the length of the required password:

```
function pick_random_symbol()
{  $symbols='$*?!-';
  $i=mt_rand(0,strlen($symbols)-1);
  return $symbols[$i];}
function pick_random_number($digits=1)
{ $min=pow(10,($digits-1));//e.g.1000
  $max=pow(10,$digits)-1;//e.g.9999
  return strval(mt_rand($min,$max));}
```

Words in the file are filtered depending on the required length:

```
function
filter_words_by_length($array,$length)
{ $select_words=array();
 foreach($array as $word)
   { if(strlen($word)==$length)
    { $select_words[]=$word;}
    } return $select_words;}
```

Depending on the length of the requested password, a password is returned using `friendly` words from utilized dictionaries, separated by symbols and numbers (Fig.2).

```
   $password.=pick_random_word($words,$next_wlen
gth
```

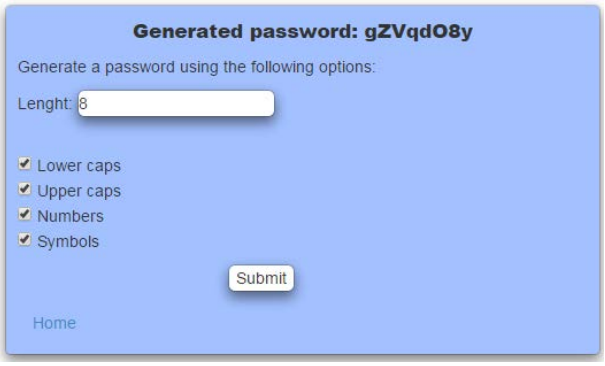

Figure 2. Interface for generating the password.

4.3 Determining password strength

A strong password is given by its construction, so that, if it is not possible to create a password of length 4, only from numbers from 0 to 9, then there will be 10 possibilities for each separate position [18]. If the password is composed of only lower case letters, there will be 26 possibilities for each separate position, and if upper case letters are added there will be 72 possibilities to set the password. Consequently, if there are only numbers, the number of possibilities for the entire password will be 10^4, and if there are also lower or upper case letters, the number will be 26^4. The `rate` function measures the password strength *rating* according to the following score: 1 if 1 number is used and 2 if 2 numbers are used, 1 if a symbol is used and 2 if 2 symbols are used, 2 if password length is equal to or higher than 8, 0.5 for each added character of length between 8 and 16, and 0 when the length of 16 characters is exceeded. The `rate` function for password length detects if the string of characters contains upper case letters, lower case letters, numbers or symbols.

```
   function detect_any_uppercase($string)
   { //true if lowercasing changes string
     return strtolower($string)!=$string;}
     function count_numbers($string)
   { return
    preg_match_all('/[0-9]/',$string);}
```

Depending on the result received, the calculation is made based on the score and the password *rating* is obtained (Fig.3).

```
function password_strength($password)
{ $strength=0;
  $possible_points=12;
  $length=strlen($password);
if(detect_any_uppercase($password))
{ $strength+=1;}
  if(detect_any_lowercase($password))
  { $strength+=1;}
$strength+=min(count_numbers($password),2);
$strength+=min(count_symbols($password),2);
  if($length>=8)
   { $strength+=2;
     $strength+=min(($length-8)*0.5,4);}
$strength_percent=$strength/(float)$possible_poi
nts;
  $rating=floor($strength_percent*10);
  return $rating;}
  $password=$_POST['rate'];
  $rating=password_strength($password);
```

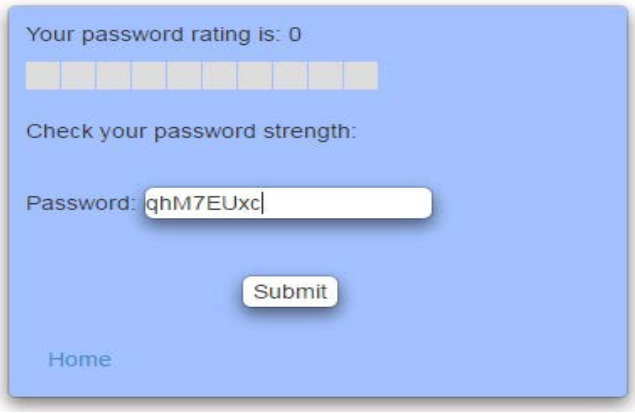

Figure 3. Password input interface.

The color is displayed depending on password strength in an .html file, in separate divs, different colors depending on the rating (Fig.4). Switching between the functions performed by interfaces is performed by clicking on the Home button

Figure 4. Password rating check interface

4.4 Number of attempts for breaking the password

The tool is based on the factorial calculation algorithm of arrangements and is structured into several entropy detection loops (Fig.5).

```
   if(detect_any_lowercase($password)&&!detect_a
ny_uppercase($password))
   {//If an uppercase character is detected
execute
   $f1=1.0;
   for($i=1;$i<=26;$i++){$f1=$f1*$i;$f2=1.0;}
   for($i=1; $i<=26-$length; $i++){//26 is the
number of lower case ASCII characters
   $f2=$f2*$i; $aranj=$f1/$f2;}
   $_POST['aranj']=$aranj; //approximate number
of iteration required to break the password
```

After selecting the entropy loop matching the entered password, the number of arrangements required for breaking the password is calculated. In the case of the above code, arrangements of 26, i.e. the number of lower case letters, are taken depending on the `$length` variable. The result is displayed with the help of the POST method. If the number in the *result* contains the letter "E", then the password is secure against a *brute-force attack* carried out from an average computer. When the error appears in the calculation of arrangements, this means that the number of arrangements that can be stored in the variable is exceeded, but at the same time this shows that an average computer would take more than 60 years to break the password, given its speed of 10,000 password/ second.

Figure 5. Interface checking the number of attempts required to break it.

Since in Moodle platform only the administrator can create and set passwords for users, a model simulating the algorithm proposed and descried in our paper, as well as in the generated password strength testing part, was posted at the following link: http://informatica-uvvg.esy.es/Password-Generator/.

5. Conclusion

In order to operate efficiently in the digital environment, most companies and universities must have a robust cyber protection system within their e-learning platforms, to inspire confidence and protect information. Although the time required breaking a password increases exponentially with its length and the number of characters utilized, a "brute force attack" cannot be prevented. To make digital operation more efficient, that is, to make things harder for hackers and cause them to continually seek and develop new methods to break passwords – which will lead to a more rapid development of information and communication technology – an algorithm was implemented. It helps to create a stronger, more efficient, much more difficult to break password, which would take a long time to uncover using current technology. Since any online program can be attached a database which accumulate all password attempts or even passwords, which are checked by users, thus creating online dictionaries of alternative passwords that could be much more efficient that the 'brute force attack" method. At the same time, the algorithm implemented in the paper emphasizes how a dictionary is created and the reasons why it is much more efficient against a "brute force attack" in educational platforms.

References

[1] Alexander J. Romiszowski, How's the E-learning Baby? Factors Leading to Success or Failure of an Educational Technology Innovation, Educational Technology, Volume 44, Number 1, pp. 5-27, January-February 2004J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73

[2] Price, Sean M., Protecting Privacy Credentials From Phishing and Spyware Attacks. In: Information Assurance and Security Workshop, 2007. IAW'07. IEEE SMC. IEEE, p. 167-174, 2007

[3] International Workshop on Soft Computing Applications, Published in Advances in Intelligent and Soft Computing, Springer, Vol.195, pp.357-365, 2013.

[4] A. Al Abdulwahid, N. Clarke, S. Furnell, I. Stengel, The Current Use of Authentication Technologies: An Investigative Review, Published in Cloud Computing (ICCC), 2015 International Conference on, pp.1-8, Riyadh, April 2015.

[5] Ioana Manea, De ce companiile trebuie să aloce bugete mai mari pentru securitatea informatică, Cisco Romanian Blog http://gblogs.cisco.com/ro/de-ce-companiile-trebuie-sa-aloce-bugete-mai-mari-pentru-securitatea-informatica/.

[6] Z. Sajjadi, A. A. Khodami, and N. Modiri, "Learning Contents integrity verification on E-Learning Systems Using Digital Watermarking Technique." pp. 1-3, 2008.

[7] S. Kawano, Y. Ohmori, T. Akai, H. Mori, Computer system, on-screen keyboard generation method, power-on-

password checking method and memory, US Patent 6832354, 2004.
[8] S. Moran, Security for mobile ATE applications, Anaheim IEEE, Published in AUTOTESTCON, pp.204-208, sept. 2012.
[9] B. Ross, C. Jackson, N. Miyake, D. Boneh, JC Mitchell, Stronger Password Authentication Using Browser Extensions, Published in the Proceedings of the 14thUsenix Security Symposium, Baltimore, August 2005.
[10] How secure is your password, https://howsecureismypassword.net/.
[11] R. Eftimie, Ghid practic - cum sa-ti creezi parole puternice, Journal Hit, Internet - Securitate, decembrie 2011.
[12] Preventing weak passwords by reading your mind https://telepathwords.research.microsoft.com/.
[13] Password Recovery Solution, http://lastbit.com/pswcalc.asp.
[14] J-S. Cho, S-S. Yeo, S.K. Kim, Securing against brute-force attack: A hash-based RFID mutual authentication protocol using a secret value, Computer Communications, Vol.34, Issue 3, pp.391–397, March 2011.
[15] The Importance of Choosing Strong Passwords, http://its.virginia.edu/accounts/passwords.html/.
[16] Mihai Jalobeanu, Antoanela Naaji, Roza Dumbraveanu, Cosmin Herman, Using Moodle platform in distance education, Proceedings of the 7th International Scientific Conference e-learning and Software for Education, pp 402-409, 2011.
[17] A Naaji, C Herman, Implementation of an e-learning system. Optimization and security-related aspects, Recent Research in Computer Science, Proceedings of the 15th Wseas International Conference on Computers, pp 412-417, 2011.
[18] A. Kovacs, Protecția informațiilor confidențiale prin criptare, Journal Info KAM Technologies, mai 2013.