Implementation and Comparison of OTP Techniques (TOTP,HOTP,CROTP) to Prevent Replay Attack in RADIUS Protocol

Amna S.M Abukeshipa
Tawfiq S.M Barhoom

تاريخ التسليم: 1 /2014/6 تاريخ القبــول: 2014/10/16

Abstract

RADIUS is one of the most popular protocols used in network communication for user authentication. Unfortunately, there are many vulnerabilities facing the security issue in RADIUS network protocol. One of these vulnerabilities is a replay attack problem which is need to be prevented. The previous protocols have presented number of techniques to reduce the effects of replay attack in RADIUS protocol. One time password (OTP) technique is one of the most important techniques which are used to enhance the security of user authentication in numerous environments and to close the potential gap in network security. With several OTP techniques, three OTP techniques (TOTP), (HOTP),(CROTP) are chosen in this paper. This paper highlights some of the current security approaches related to RADIUS protocol and OTP techniques. The main focus of this paper is the implementation of the three OTP techniques through providing our ELSBOT and the comparison of these techniques through presenting a set of factors such as preventing replay attack, CPU overhead, technique speed, server response time and OTP duration.

الملخـــــص:

بروتوكول RADIUS هو أحد البروتوكولات الأكثر شيوعاً والمستخدمة في شبكة الاتصالات لمصادقة المستخدم (التحقق من المستخدم). للأسف، هناك العديد من نقاط الضعف التي تواجه قضية الأمن في بروتوكول RADIUS. مشكلة هجوم إعادة الإرسال Replay Attacks تعتبر من أحد نقاط الضعف في هذا البروتوكول التي نحتاج إلى منعها. البروتوكولات السابقة قدمت عدد من التقنيات للحد من آثار هذا الهجوم في بروتوكول RADIUS. تقنية كلمة المرور لمرة واحدة One Time Password(OTP) من أهم التقنيات التي استخدمت لتعزيز أمن مصادقة المستخدم في بيئات عديدة ولسد الفجوة المحتملة في أمن الشبكات. مع العديد من تقنيات OTP، مساهمتنا في هذا البحث هو اختيار ثلاث تقنيات: TOTP، HOTP، CROTP. وهذا دفعنا لتقديم ELSBOT لتفادي مشكلة هجوم إعادة الإرسال Replay Attacks في بروتوكول RADIUS. في هذه الورقة سوف نطبق تقنيات OTP الثلاثة في ELSBOT ضمن بيئة RADIUS ومن ثم نقدم مقارنة بين تقنيات OTP. ولفحص وتقييم نظامنا ELSBOT في ظل وجود مجموعة من العوامل مثل: منع هجوم إعادة الإرسال، العبء على المعالج وسرعة المعالج وزمن الاستجابة للسيرفر ومدة صلاحية كلمة المرور. تظهر النتائج بأن تقنيات OTP الثلاثة في ELSBOT تمنع مشكلة هجوم إعادة الإرسال ضمن بيئة RADIUS وكما أن العبء على المعالج في تقنية TOTP أقل من التقنيات الأخرى وكذلك سرعة المعالج في تقنية TOTP هى الأعلى بينما سرعة المعالج في تقنية CROTP أعلى من تقنية HOTP ولكن تقنية HOTP هى الأفضل بالنسبة لزمن إستجابة السيرفر، وبالتالي فإن نظام ELSBOT هو أفضل الأنظمة المتوفرة من منظور أمني.

***Keywords**: RADIUS Protocol , Replay Attacks ,One Time Password.*

.

I Introduction

Internet has become an essential part of our lives, and it is changing every aspect of our life. As Internet is widely being used at most of organizations which need to protect their information from unauthorized access and data during transaction process.

Many security protocols were developed to provide mechanisms for protecting data during transmitting over networks. Remote Authentication Dial In User Service (RADIUS) is one of the most common protocols to perform distributed services of Authentication , Authorization, and Accounting (AAA) for dial-up remote access [2]. Unfortunately, there are several vulnerabilities in RADIUS protocol. These vulnerabilities are caused by the protocol itself or are caused by poor client implementation such as response authenticator based shared secret attack, Denial Of Service (DOS) arising from the prediction of the request authenticator, user-password based password attack and replay attack. A replay attack is a form of network attack which is carried by an adversary who intercepts the data and retransmits it, a valid data transmissions maliciously or fraudulently repeated or delayed [13].

While poor implementation in Pseudo Random Number Generator (PRNG) in RADIUS server can be predicable and more likely to be repeated and violated from replay attack. Shared secret between RADIUS server and a RADIUS client can be violated from attacker through passive eavesdropping. this attacker can build a dictionary to find patterns and break a cipher, then reply to server with valid login information. Previous studies have provided numerous security solutions to reduce the effects of replay attacks [5][6][13]. Some of these solutions used IPSec [9] and others used the strong cryptographically request authenticators [2] to reduce the effects of this attack. As described in [15,17,18], the researchers used the smart cards to prevent this type of attack.

One Time Password (OTP) technique is used at numerous environments to close the potential gap in network security and can be used to provide a secure access to remote users [16,19]. There are different OTP techniques such as Mobil OTP(MOTP), SMS OTP, Linux OTP(LOTP), Time OTP (TOTP), Challenge-Response OTP(CROTP), and Hash OTP (HOTP). This paper presents three OTP techniques (TOTP, HOTP, CROTP) and then implements these techniques in RADIUS protocol in order to prevent replay attacks and to provide a secure environment for our organizations.

The rest of the paper is arranged as follows: Section II includes related work with its review and discussion; Section III presents our ELSBOT which implements three OTP techniques in RADIUS protocol to prevent replay attack; Section IV tests and evaluates our ELSBOT and then presents a comparison of the three OTP techniques. Finally, Section V draws the main conclusion and outlines our future work.

II Related work

In the past years, a number of security approaches have been proposed for securing RADIUS protocols and for preventing replay attacks. Also there are different studies presented OTP techniques in different areas in security.

Md. Hashmathur, et al.[2] have presented different approaches to minimize or resolve problems of the RADIUS protocol using strong shared-secrets, cryptographic-quality values for the request authenticator. In [3] Ang, et al. have discovered that some weaknesses exist in Encrypted Key Exchange(EKE) protocol that are subjected to replay attacks; also they proposed an improved scheme Efficient Password-Proven Key Exchange (EPPKE) protocol that generates the session keys timestamp. In 2012, Rohan Deshmukh [1] has developed a proposal to improve the communication efficiency between NAS and RADIUS server by allowing the RADIUS

server to communicate its state (active/dead) to NAS. Their proposal has effectively helped to improve less CPU utilization in the network.

In 2014, P.Anuja, et al. [23] proposed a user authentication technique of OTP (TOTP,HOTP,SMSOTP) that can be used to prevent from password stealing and password reuse attacks by installing the applications on the Android smartphone. Their system has a recovery procedure when the user loses the cellphone. The researchers of [7] have analyzed the problems of vulnerability of authentication mechanisms by using existing S/key authentication mechanism. They used CROTP and TOTP techniques with public key infrastructure. Their proposed mechanism can prevent spoofing attack in advance by authenticating user with the use of certificate information. There is a new improved unilateral asymmetric authentication protocol that can be used against replay attacks and active attacks as described in [16]. The complexity of this proposed protocol is higher. they used a smart card to protect user authentication. This protocol is useful for applications that are important and critical. While Misbahuddin, et al. [18] have used a password based authentication schemes for authenticating remote users. They proposed an efficient scheme for remote user authentication with smart card.

III. Implementing OTP Techniques (TOTP,HOTP,CROTP)

This section presents the ELSBOT (E-Learning System Based OTP Techniques) for implementing the three OTP techniques (TOTP, CRTOP, HTOP) in the RADIUS environment in order to prevent the replay attack. The ELSBOT is shown as the basic part of our work. ELSBOT is an educational system that offers a set of services for students and staff in a college like add new students and courses, register and withdraw courses, delete students and courses, update students and courses. The aim of ELSBOT is to find a secure authentication mechanism to prevent the replay attack in the RADIUS environment and to protect user information from theft. This section discusses the architecture, the requirements, the databases and the replay attack process for ELSBOT in the following subsections:

A. ELSBOT Architecture

In general, We provide the functional architecture for our ELSBOT. This architecture has five components as shown in Figure 2. These components are namely:

1. **User** is a person who requests access to the system services.

2. **System** is a website that will design to provide services to end users.

3. **RADIUS server** checks the user information (username and password) which is entered in the sign-in form. The RADIUS server has a database to store the user data.

4. **Authentication server (AS)** is responsible of the second phase in our system. After The system user enters the PIN and the generated OTP in the OTP page, the AS will check the secret key, PIN and last OTP .

5. **OTP application** is a software which is downloaded on the user computer at anytime and in anywhere. This application is responsible of the OTP generation process for each of the three techniques.

To represent the work mechanism between the components of ELSBOT architecture, a set of steps are presented in Table 2.

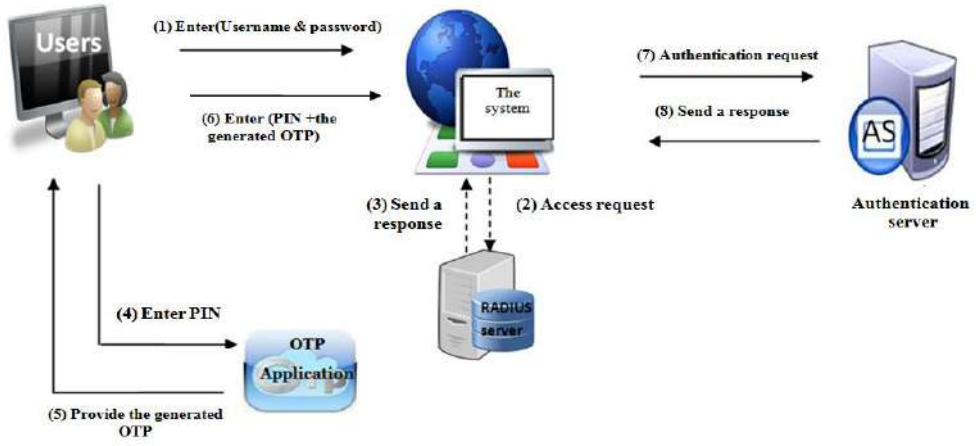


Figure 2. Functional Architecture of ELSBOT

Table 2. Shows the steps of ELSBOT architecture

Step #	Description
1.	The user enters the username and password in the login screen. The password should contain more than six characters.
2.	The system sends user data to the RADIUS server to authenticate the user.
3.	The RADIUS server verifies the username and password and sends a response either accept or reject.
4.	The user opens the OTP application and enters the PIN.
5.	The OTP application provides the generated OTP for the system user. In TOTP , this OTP is expired after 3 minutes.
6.	The user enters the PIN and the generated OTP through the system ,but in CROTP techniques the PIN send from server that send challenge random PIN to user.
7.	The system sends a request to the AS to check the last OTP, the PIN and the secret key of the system user.
8.	The AS verifies user OTP and sends a response to system either accept or reject.

B. ELSBOT Requirements

Table 3. ELSBOT Requirements

Platform	Operating Systems	Software	Additional libraries	Modules
RADIUS server	Linux CentOS 6.4.1	FreeRADIUS 2.1.12	SOAP	LAMP
Authentication Server	Windows 7	Apache	SOAP	Xampp
OTP application	Windows 7 at user computer	Java language with JDK() and Netbeans IDE 7.3	Hash -MD5	-
Simple Web site	Windows 7	PHP	Jquery	Xampp

This subsection presents the guaranteed requirements to be taken when applying the ELSBOT. The basic requirements of ELSBOT are categorized in terms of software, hardware and network. These requirements are shown in Table 3.

C. ELSBOT Databases:

To implement our ELSBOT, we need three databases as shown in Figure 3. The three databases of ELSBOT are described below:

- **College Database** is a data store that contains information about all the users, students, courses, and other information related to our system.
- **RADIUS Database** is checks and verifies from user login process. This database includes two tables: Acct.radius_userstable, and Logs table.
- **Authentication Database** includes two tables, settings table, and logs table.

There are three steps to represent data traffic between these databases in our system. These steps are:

(1) RADIUS server checks that if the user is exist in the radius database which contains all system users.

(2) Authentication server checks the last OTP, PIN and secret key of the system user. This data is saved in the authentication database.

(3) If the user is admitted by the two servers, the user can enter to the system which has a college database.

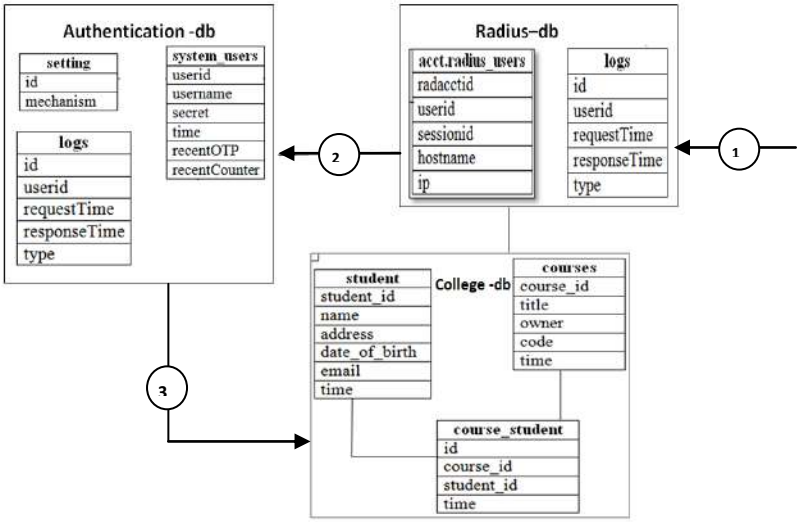


Figure 3. ELSBOT Databases

D. Replay Attack

OTP authentication methods are vulnerable to

certain kinds of attacks such as replay attacks [5], [6]. To represent the replay attack process in the ELSBOT, the chrome extension application is used to create malicious JavaScript to take data from server attack and sent back to attacker .

Chrome extension application can listen to any request processed to user system based on jquery library to implement data replay using command *" jQuery.parseJSON(data)"* . Attacker retains data stolen in console log *"console.log(data);"* where chrome console tool continues to operate connect with the user, then the attacker

captures all the required data (username/static password)), PIN and OTP " *$("#otp").attr("value",data.otp); $("#pin").attr("value",data.pin); $(".data")*" .

Database for attacker includes two tables, the first table called *users* that includes username of all users with their passwords that have been captured through sniffing on ports or implement malicious script. The second table is *OTP* table that includes Id PIN and OTP that stolen from entering the users of the system. Attacker database is shown in Figure 4.

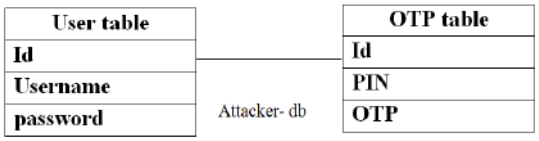


Figure 4. Attacker Database

To apply replay attack process on our ELSBOT, six components are needed. These components are shown in Table 4**.**

Table 4. Replay Attack Components

Step#	Components	Module	library
1.	Authentication server	XAMPP	SOAP
2.	RADIUS server	LAMPP	SOAP
3.	Attacker server	XAMPP	jquery
4.	Attacker computer	XAMPP	jquery
5.	The user (Victim)	XAMPP/LAMPP	
6.	Internet browser	Google *Chrome*	Chrome Consol

In replay attack architecture, the attacker server is able to capture the username and password from user within a malicious script which is recording the private information for users, the attacker server sends the username & password to attacker who is stealing and trying to authenticate username and password in the system. This attacker stills to listen and take any information from user in each traffic over the network. The replay attack architecture is described in Figure 5. And then we describe the replay attack process through a set of steps which are shown in Table 5.

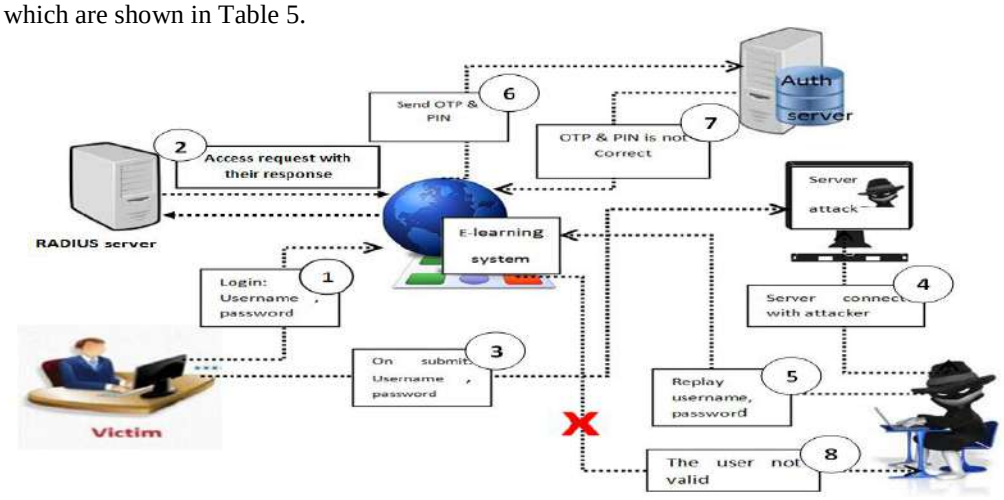


Figure 5. Replay Attack Architecture

Table 5. Steps for replay attack process

Step#	Description
1.	The user enters the username and password in the login screen.
2.	The system sends user data to the RADIUS server to authenticate the user . the RADIUS server verifies the username and password and sends a response either accept or reject.
3.	Attack depends on the reception of incoming requests to the server continuously without interruption.
4.	Attacker server sends malicious software script to user to listen on any request processed to system.
5.	Attacker steals data and last request , then sends these data for ELSBOT.
6.	Attacker returns sending data for ELSBOT through running malicious script.
7.	ELSBOT receives stolen data from attacker and replays for authentication server.
.8	Authentication server implements DB query and verifies of PIN, OTP, secret key, lastOTP and response with " the user is not authenticate".

IV. Test and Evaluation

This section draws three directions in order to test and evaluate our ELSBOT. Firstly, the average response time for authentication server is measured at each OTP technique. The second direction offers the synchronization process at each of the three OTP techniques to measure CPU overhead and technique speed. The other important direction of this research tests our ELSBOT against the replay attack by applying an attacked script which tries to steal user data. Finally, this section draws a comparison between the three OTP techniques. This comparison is presented by considering a set of factors which are mentioned later.

A. Server Response Time :

To measure the server response time, ten different cases are offered for each of the three OTP techniques. These cases show the server response time for each request by the system user. Table 6 shows the user request time and its response time by the server. While Unix time stamp is a way to track time as a running total of seconds. This count starts at the Unix Epoch on January 1st, 1970 at UTC. Therefore, the unix time stamp is merely the number of seconds between a particular date and the Unix Epoch [22]. For example, the user request time for case 1 is equal to 1386261391 as timestamp. This case is equal to 12/05/13,4:36:31pm as current date.

Case #	Request Time	Response Time	Technique Type	Drift Second
1	1386261391	1386261393	CROTP	2
2	1386261499	138626151	CROTP	2
3	1386261925	1386261926	CROTP	1
4	1387639886	1387639887	CROTP	1
5	1387639978	1387639980	CROTP	2
6	1387642561	1387642563	CROTP	2
7	1387705748	1387705750	CROTP	2
8	1387705796	1387705797	CROTP	1
9	1387705710	1387705711	CROTP	1
10	1387705711	1387705711	CROTP	0
11	1387705812	1387705815	HOTP	3
12	1387806696	1387806697	HOTP	1
13	1387807011	1387807014	HOTP	3
14	1387807056	1387807058	HOTP	2
15	1387807100	1387807101	HOTP	1
16	1387809597	1387809599	HOTP	2
17	1387809598	1387809598	HOTP	0
18	1387870770	1387870772	HOTP	2
19	1387870777	1387870778	HOTP	1
20	1387870790	1387870762	HOTP	2
21	1388519419	1388519420	TOTP	1
22	1388519497	1388519499	TOTP	2
23	1388519572	1388519573	TOTP	1
24	1388519580	1388519582	TOTP	1
25	1388519585	1388519586	TOTP	1
26	1388519592	1388519593	TOTP	1
27	1388519635	1388519636	TOTP	1
28	1388519640	1388519642	TOTP	2
29	1388519670	1388519671	TOTP	2
30	1388519679	1388519680	TOTP	1

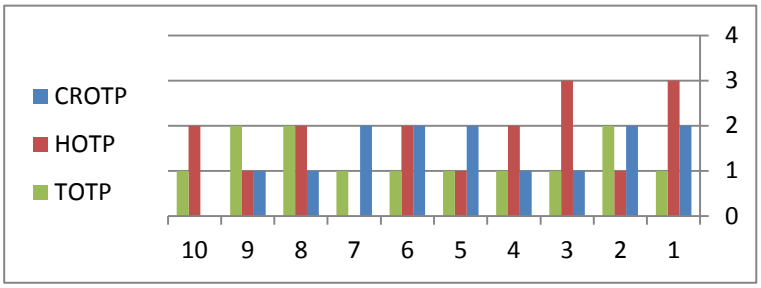
Table 6. Shows the user request time and server response time in different times

After showing these cases, the authors compute the average response time for each technique. In table 7, the results show that the average response time of HOTP is equal to 1.7500 sec., the average response time of TOTP is equal to 1.2500 sec. and the average response time of CROTP is 1.6250 sec. Thus, we find that the average response time of TOTP is lower than CROTP and HOTP. The Figure 6 illustrates the average server response time for the three OTP techniques.

Table 7. Average Response Time Values

OTP Technique	CROTP	HOTP	TOTP
Average response time	1.6250	1.7500	*1.2500*

Figure 6 . Average Response Time at the three OTP techniques



B. Synchronization Process:

After applying ELSBOT, some of the synchronization challenges are found at each OTP technique. One of the most important synchronization challenges is the CPU overhead. The CPU overhead challenge is caused by the hash process which is used in the OTP generation process in the ELSBOT. Moreover, if the system performs more hash processes, the system will busy the CPU. Thus, we measure the CPU overhead for each OTP technique in ELSBOT through representing a different cases. The synchronization process is offered for each OTP technique separately as shown below:

# Case	*User*	*IP address*	*User clock*	*Server clock*	*Hash processes (roll)*
1	***A***	***10.10.10.1***	***12:29:10***	***12:30:00***	***5***
2	***B***	***0.10.10.31***	***12:25:20***	***12:27:00***	***10***
3	***C***	***0.10.10.24***	***12:15:00***	***12:13:20***	***28***
4	***D***	***0.10.10.16***	***11:46:00***	***11:48:00***	***12***
5	***E***	***10.10.10.8***	***12:32:00***	***12:29:10***	***35***
6	***F***	***10.10.10.22***	***10:30:00***	***10:32:30***	***15***

B1. TOTP Synchronization:

The TOTP synchronization process is provided to synchronize between the server time and the user time. In this technique, the time is the basic factor in the hash process which is used for generating our OTP. In

TOTP technique of our ELSBOT, the authors determine 36 rolls (i.e., 36 hash processes) as a worst case to represent the synchronization process for this technique. In the Figure 7, the 36 rolls of TOTP synchronization process are divided into two halves: (1) The first half of the synchronization process is used when the user time is less than the server time; (2) The second half of the synchronization process is used when the user time is greater than the server time. The system works a hash roll every 10 seconds (i.e., one roll is equal to 10 seconds). Thus, the worst case of the TOTP synchronization process is equal to 36 rolls/hash processes. This case needs 360 seconds

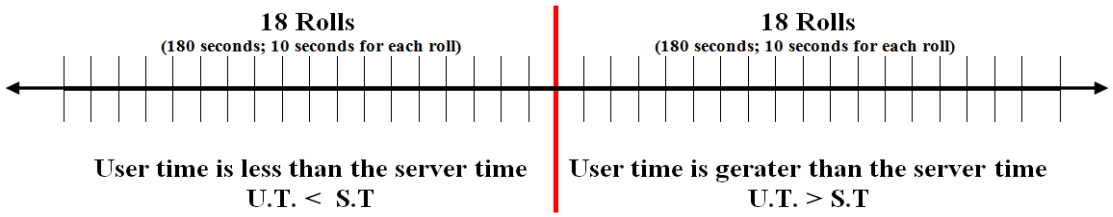


Figure7. Represents the Synchronization process of TOTP

To represent the TOTP synchronization process in ELSBOT, a set of different cases are presented to synchronize the time between the server and the user as shown in the Table 8. If the server clock and the user clock are differ, the system will synchronize the time between the server and the user in order to match between the server clock and the user clock. The synchronization process may require more hash processes by the system. As stated previously, the synchronization process of TOTP technique provides 36 rolls/ hash processes as the worst case. We choose two cases to show the synchronization process of TOTP .

Table 8: Synchronization Cases of TOTP

Case 1 : The server time is greater than the user time

User A requests a response from the server at the time 12:29:10, while the server time is 12:30:00. We find that the server time is greater than the user time by 50 seconds. The researcher find the first half of the figure to compute the number of rolls. Thus, the system needs to 5 rolls /hash processes.

Case 3: The user time is greater than the server time

User C requests a response from the server at the time 12:15:00, while the server time is 12:13:20. We find that the user time is greater than the server time by 100 seconds. In this case, firstly, the system works 18 rolls of the first half of the Figure 7 and works 10 rolls from the second half. Thus, the system needs to 28 rolls /hash processes in this case of synchronization

B2.CROTP/HOTP Synchronization:

The synchronization process is the same in CROTP and HOTP techniques. The two techniques use the counter to synchronize between the server and the user. This counter is the basic factor in the hash process. The server saves the counter value for the last valid login process by the system user (i.e., if the user works five valid login processes, the last counter at server is equal to 5. When the user works invalid login processes as the margin of error, the server will add 1 to its counter in order to match with the user counter. This can lead to the counter value being out of synchronization between the user and server. Thus, the server needs a number of

rolls/hash process for the matching process. For two techniques, the authors determine 50 rolls (i.e., 50 hash processes) as a worst case to represent the synchronization process for these techniques.

To represent the synchronization process at the two techniques, a set of different cases are presented to synchronize between the server counter and the user counter as shown in Table 9. If the server counter and the user counter are differ, the server will synchronize its counter with the user counter. The synchronization process may require more hash processes by the system. As stated previously, the synchronization process of two techniques in the ELSBOT provides 50 rolls/ hash processes as the worst case. The synchronization cases for the two techniques are shown below:

Case 1:

User A works invalid login through pressing the login button. Thus, the counter of this user becomes 50 as the margin of error. While the server counter is equal to 5. The system needs 45 rolls /hash processes in this case of synchronization. Case 1 is the worst case in the synchronization process of CROTP and HOTP.

Case 6:

User F works invalid login through pressing the login button. Thus, the counter of this user becomes 6 as the margin of error. While the server counter is equal to 5. The system needs 1 roll /hash process in this case of synchronization. Figure 8 illustrates the number of rolls/hash processes for the different cases of the two techniques.

Table 9: CROTP/HOTP Synchronization Cases

# Case	*User*	*IP address*	*User*	*Server*	*Hash processes (roll)*
1	*A*	*10.10.10.1*	*50*	*5*	*45*
2	*B*	*0.10.10.31*	*45*	*5*	*40*
3	*C*	*0.10.10.24*	*15*	*2*	*13*
4	*D*	*0.10.10.16*	*22*	*10*	*12*
5	*E*	*10.10.10.8*	*10*	*15*	*5*
6	*F*	*10.10.10.22*	*5*	*6*	*1*

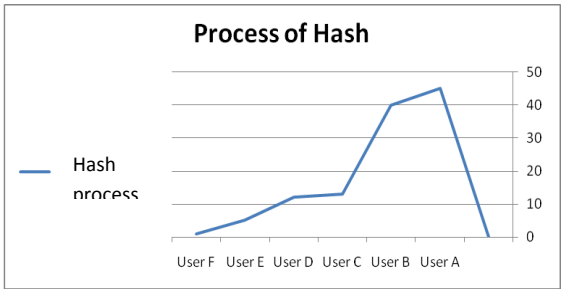


Figure 8: Number of hash processes for CROTP/HOTP synchronization Cases

C. Measuring ELSBOT against replay attacks:

The researchers present two steps to draw the process of data theft by the attacker. In the first step, the user enters his data (username and password) in the login screen of our system. On the other hand, the attacker snoops and steals the user data to login to the system and then takes a successful authentication from RADIUS server. Thus, the attacker can move to the next screen in the system. In the second step, after entering the user name and password from user, the user enters the PIN and OTP generator. The attacker can capture PIN & OTP from user, but the authentication is failed from the authentication server. Therefore, the attacker cannot enter to our system because the authentication server generates a unique number for each connection and checks the secret key, the PIN and the last OTP which is not exist in the attacker side.

D. Results Comparison

The implementation of our work proves that the ELSBOT is more effective, more powerful, and more able to satisfy the security needs in the RADIUS environment. A comprehensive comparison between the three OTP techniques of ELSBOT is presented in this sub section. This comparison considers a set of factors. These factors are: preventing replay attack; CPU overhead, speed, server response time, and duration of generated OTP.

Firstly, the authors measure the server response time at each of the three OTP techniques based in 30 cases (10 cases for each of the three OTP techniques). The result shows that the average server response time of TOTP is equal to 1.2500 seconds, the average server response time of HOTP is equal to 1.7500 seconds, and the average response time of CROTP is equal to 1.6250 seconds. Thus, the TOTP technique in the ELSBOT is the best because this technique gets less average response time by the server. The OTP techniques of ELSBOT are arranged by less server response time in the order TOTP,CROTP, and HOTP.

The CPU overhead factor is measured through six cases for each of the three OTP techniques in the synchronization process. These cases show that the TOTP technique needs 36 hash processes as the worst case, while the HOTP and CROTP techniques need 50 hash processes as the worst case. The result of measuring the CPU overhead finds that the CPU overhead at TOTP technique is less than the CPU overhead at HOTP and CROTP techniques. Also, through the synchronization process, the speed is measured for each OTP technique. Whenever the system works more hash processes, the technique speed is low. The result of measuring the speed factor shows that the speed at TOTP technique is the highest, and the speed at CROTP technique is higher than HOTP technique. In the CROTP technique, the system provides the user with a random PIN and then the system does not check this PIN leading to speed up the processor in the system matching process.

After measuring ELSBOT against replay attacks, attacker can steal username and password in the first phase of login process . In the second phase of login process, after entering the PIN and the generated OTP, the attacker can also steal the user PIN and OTP. But, this attacker can not login to system because he doesn't have the secret key. And then the system matching process at the attacker is failure. We apply this type of attack on the ELSBOT in order to test each of the three OTP techniques against replay attack. According to testing and evaluating these techniques against replay attack, we reach that the three OTP techniques prevent this replay attack in our ELSBOT.

In the ELSBOT, three OTP techniques are implemented for generating the OTP. In TOTP technique, the generated OTP is valid for a short time (i.e., ends after 3 minutes) and this generated OTP cannot be reused. In HOTP and CROTP techniques, the generated OTP is valid for a long time and is used for an unknown amount of time.

Moreover, the TOTP technique is usually used in the applications that need to restrict the time like mobile applications and banking transactions. The HOTP technique is used in the applications that are not concerned time like emails. While the CROTP is used with the complex applications, especially in servers applications. Finally, It is clear from the results obtained in this section that the TOTP is the most secure technique, while the CROTP is a more secure than HOTP because the server challenges us with the random PIN in the CROTP. Table10 summarizes the comparison among the three OTP techniques:

Table 10. OTP Techniques Comparison

Factor	OTP Techniques		
	HOTP	***TOTP***	***CROTP***
Replay Attack	Prevent	Prevent	Prevent
CPU Overhead	High (50 rolls/hash processes)	Low (36 rolls/ hash processes)	High (50 rolls/hash processes)
Technique speed	Low	High	Medium
Server Response Time	High	Low	Medium
OTP Duration	Long time	Short time	Long time

V. Conclusion and Future Work

A. *Conclusion*

this paper, we implemented and analyzed three OTP techniques to prevent replay attacks in RADIUS protocol and provided a comparison between these techniques of our ELSBOT by considering a set of factors such as preventing replay attack, CPU overhead, technique speed, server response time and OTP duration. After testing and evaluating of our work, the researchers found that the three OTP techniques prevent the replay attack in RADIUS environment. The CPU overhead at TOTP technique is less than others and the speed at TOTP technique is the highest while the speed at CROTP technique is higher than HOTP technique. The average server response time at TOTP technique in our ELSBOT is the best in terms of server response time. Finally, according to these results, we reach that the TOTP is the most secure technique because this OTP is valid for a short time, while the CROTP is a more secure than HOTP because the server challenges us with the random PIN in the CROTP. Thus, our ELSBOT is an efficient overall solution from the security perspective and it will be much more difficult for attackers to reach the ELSBOT server.

B. Future Work

In this section, a set of points are drawn to extend our future work. These points are namely below:

1. Send the secret key to user via the SMS service or email.
2. The length of the generated OTP in our work is 6 characters. The authors propose to increase the length of OTP because the long OTP makes a system more secure.
3. In this work, two fields are used in the authentication process (PIN and OTP). Consider other fields to enhance the security for user authentication that requires additional research.

4. Extend our ELSBOT to be become a new proposed model .

References

[1] Rohan Deshmukh , "Interactive Remote Authentication Dial In User Service (RADIUS) Authentication Server Model", International Conference on Wireless and Mobile Communications, ICWMC 2012.
[2] Hashmathur Rehman .Govardhan T. Venkat Narayana Rao ," Design and Implementation of RADIUS An Network Security Protocol" , Global Journal of Computer Science and Technology, P a g e 48 Vol. 10 Issue 7 Ver. 1.0 September 2010.
[3] Ang Gao, Wei Wei and Wenbo Shi , "Efficient Password-Proven Key Exchange Protocol against Relay Attack on Ad Hoc Networks",International Conference on Advanced Information Networking and Applications, 2010 IEEE.
[4] Xuguang Ren , Xin-Wen Wu , Kun Tang ,"TSPass: A Dynamic User Authentication Scheme Based On Time and Space", IJCSNS International Journal of Computer Science and Network Security, VOL.12 No.10, October 2012.
[5] madhuri and ramana lakshmi "attack patterns for detecting and preventing ddos and replay attacks", international journal of engineering and technology vol. 2(9), 4850-4859, 2010.
[6] Lan Sun, Zhao Luo, Yingjie Wu, Yilei Wang , "A Technique for Preventing Replay Attack in Road Networks", International Conference on Advanced Information Networking and Applications, 2012 IEEE.
[7] Hyun-Chul Kim1, Hong-Woo Lee1, Kyung-Seok Lee 1, Moon-Seog Jun1 ,"A Design of One-Time Password Mechanism using Public Key Infrastructure", 2008 IEEE.
[8] Xi Yang , "The Use of One-Time Password and RADIUS Authentication in a GSS-API Architecture" master thesis 2006.
[9] Mikko Saarinen, " Legacy User Authentication with IPSEC" master thesis 2004.
[10] Kenneth G. Paterson, Douglas Stebila " One-time-password-authenticated key exchange (full version) " , Australasian Conference on Information Security and Privacy (ACISP) 2010, Springer 2010.
[11] C. Rigney, S. Willens, A. Rubens Merit , rfc2865 "Remote Authentication Dial In User Service (RADIUS)", The Internet Society (2000).
[12] Daniel Szilagyi, Arti Sood, and Tejinder Singh, "Radius: A Remote Authentication Dial-In User Service" , JOURNAL, VOLUME 5, NUMBER 2, 2009.
[13] Florian Devic , Lionel Torres, Benoˆt Badrignans," Secure protocol implementation for remote bitstream update preventing replay attacks on FPGA", 2010 IEEE.
[14] Daniel Granlund, Christer ¦hlund ," A Scalability Study of AAA Support in Heterogeneous Networking Environments with Global Roaming Support ", 2011 IEEE.
[15] Jitendra Kumar Tyagi , A.K. Srivastava , Pratap Singh Patwal ," Remote User Authentication Scheme in Multi-server Environment using Smart Card ", International Journal of Computer Applications ,Volume 57– No.12, November 2012 .
[16] Chun-Ying Huang , Shang-PinMaa, Kuan-TaChen" , Using one-time passwords to prevent password phishing attacks ", 2011 Elsevier.
[17] Lanjun Dang , Weidong Kou , Yuxia xiao ,"An Improved Scheme for Unilateral Asymmetric Smart Card Authentication", International Conference on Advanced Information Networking and Applications, 2005 IEEE.
[18] Mohammed Misbahuddin, Mohammed Aijaz Ahmed, M.H. Shastri ,"A Simple and Efficient Techniques to Remote User Authentication Using Smart Cards", International Conference on Advanced Information Networking and Applications, 2006 IEEE.
[19] Havard Raddum, Lars Hopland Nestas, and Kjell Jrgen Hole," Security Analysis of Mobile Phones Used as OTP Generators ", IFIP International Federation for Information Processing 2010.
[20] Sung-Jae Lee, Jae Seong Lee, Mun-Kyu Lee, Sang Jin Lee, Doo-Ho Choi, and Dong Kyue Kim ,"Low-Power Design of Hardware One-Time Password Generators for Card-Type OTPs", ETRI Journal, Volume 33, Number 4, August 2011.
[21] Ms. E.Kalaikavitha M.C.A., M.Phil., 2,Mrs. Juliana gnanaselvi M.Sc., M.Phil," Secure Login Using Encrypted One Time Password (Otp) and Mobile Based Login Methodology ", International Journal Of Engineering And Science Vol.2, Issue 10 (April 2013), Pp 14-17.
[22] http://www.unixtimestamp.com/index.php(February,2014) online.
[23] R.SelvaBhuvaneshwari, P.Anuja," Secured Password Management Technique Using One-Time Password Protocol In Smartphone ", International Journal of Computer Science IJCSMC, Vol. 3, Issue. 3, March 2014, pg.976 – 981.
[24]Tamanna Saini" One Time Password Generator System" , International Journal of Computer Science, IJCSMC Volume 4, Issue 3, ISSN: 2277 128X , March 2014.