Journal of Mobile, Embedded and Distributed Systems, vol. VI, no. 4, 2014
ISSN 2067 – 4074

PayPal Transactions Security

Răzvan TOADER
IT&C Security Master
Department of Economic Informatics and Cybernetics
The Bucharest University of Economic Studies
ROMANIA

Abstract: Recent threads to prominent organizations and companies have greatly increased the need for information security. Many measures have been designed and developed to guard against threats from outsider attacks. Technologies are actively implemented to prohibit such attacks that could actively prohibit rogue connections. In this paper, common vulnerabilities for PayPal transactions identified as well as solutions for defending against them.

Key-Words: PayPal, Security, Malware, Public-Key, Private-Key, Cryptography, XSS, Man-in-the-Middle

1. Introduction

PayPal is a Web-based e-commerce business that allows money transfers and payments to be made through the Internet. It offers a secure method to transfer funds between individuals or business electronically. A special feature of PayPal is that it does not transmit any financial information over the Internet as a bank would do. It is similar to an escrow service as PayPal acts as the middleman holder of the money between both parties involved in a transaction.
In this paper I will focus on the specific case of Cross-Site Scripting attacks (XSS) against the security of web applications. The XSS attack is very similar to SQL Injection due to the fact that both attacks inject code into the web application that poses a threat. A study documented by Symantec in 2007 showed that roughly 84% of websites are susceptible to cross site scripting.

2. Problem Formulation

2.1 PayPal Public Key Encryption

PayPal uses the public key cryptography technology to encrypt your PayPal button created from your account that is used in a website.
Public key cryptography provides a way to prove the identity in the online world. This cryptographic algorithm uses two keys (public-key and private-key) which are bits of data that are mathematically related to one another. This type of cryptographic algorithm only works only if you keep the private key confidential, while the public key can be made available. The public keys are distributed inside a digital certificate.
Digital certificates represents a file that contains information about the public key (name of the company that owns the public key, the third party company that distributed the certificate and the certificate expiration date) and the key itself. The third party involved in signing the certificate is referred as a certificate authority (CA). The CA signs the digital certificate, and the consumers can validate that the public key is valid by using the public certificate of the CA to verify the digital signature.

```
<form action="https://www.sandbox.paypal.com/cgi-bin/webscr"
method="post" target="_top">
<input type="hidden" name="cmd" value="_s-xclick">
<input type="hidden" name="hosted_button_id"
value="P78HHB5SFVYRW">
<input type="image"
src="https://www.sandbox.paypal.com/en_US/i/btn/btn_buynow_LG.gif"
border="0" name="submit" alt="PayPal - The safer, easier way to pay
online!">
<img alt="" border="0"
src="https://www.sandbox.paypal.com/en_US/i/scr/pixel.gif" width="1"
height="1">
</form>
```

Figure 1. Encrypted PayPal button

Even though you can generate an encrypted PayPal button either from your PayPal account or through the PayPal API, many websites still use the standard

HTML form for a Buy Now button or Subscription button.
The code for the encrypted PayPal button is presented in the image below (Figure 1).
You can see in Figure 1 that the input with the name "*hosted_button_id*" holds the encrypted value for the product. This includes the business name, item name, price, currency shipping and tax values entered in PayPal.
This type of PayPal button is secure because it does not display any information about the merchant or about the item itself.
Another type of PayPal button is the one where you can enter all the variables in plaintext as seen below (Figure 2).

```
<form action="https://www.sandbox.paypal.com/cgi-bin/webscr" method="post">
<input type="hidden" name="cmd" value="_cart">
<input type="hidden" name="business" value="razvan.toader@dizertatie.ro">
<input type="hidden" name="item_name" value="shirt">
<input type="hidden" name="item_number" value="123">
<input type="hidden" name="amount" value="15.00">
<input type="hidden" name="first_name" value="Razvan">
<input type="hidden" name="last_name" value="Toader">
<input type="hidden" name="address1" value="Teiul Doamnei 110">
<input type="hidden" name="address2" value="Apt 5">
<input type="hidden" name="city" value="Bucharest">
<input type="hidden" name="zip" value="19312">
<input type="image" name="submit" border="0"
src="https://www.paypalobjects.com/en_US/i/btn/btn_buynow_LG.gif"
alt="PayPal - The safer, easier way to pay online">
</form>
```

Figure 2. Plaintext PayPal code

In Figure 2 you can see that the personal information about your account is now visible: business (the merchant account), amount (the price of the item), first name, last name, address 1, address 2 city, zip (information about the buyer).
All the above information is in plaintext and can be easily altered for malicious intents.
For example using a man-in-the-middle attack as seen in Figure 3, the attacker intercepts the communication between the victim and the Web Server. Using different techniques, the attacker can split the original connection into two new connections, one between the victim and the attacker and the other between the attacker and the Web Server. Once the connection is intercepted, the attacker can read and alter the data from the intercepted connection. Once the attacker intercepts the connection he can alter the merchant account by adding his own PayPal account, so the transaction goes into his account, instead of the merchants. He can also modify the amount so that the victim is charged more.

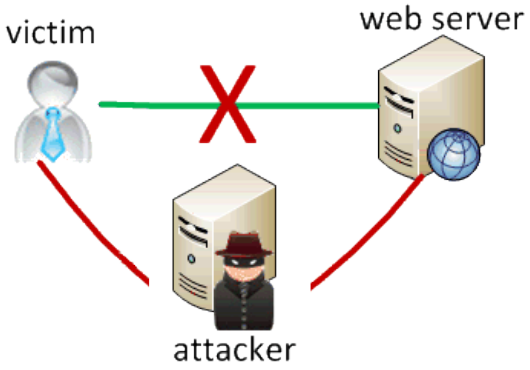

Figure 3. Attacking scenario

This type of alteration is made through cross site scripting. For instance let's say that there is a website that displays the HTML form showed in Figure 2. The attacker can insert the following code in the webpage:

```
window.onload = function(){
    document.getElementsByTagName('form')[0].addEventListener("submit", function(event){
        event.preventDefault();
        var inputs = document.getElementsByTagName('input');
        for(var i=0; i<inputs.length; i++){
            if(inputs[i].name === 'business')
                inputs[i].value = 'attacker@dizertatie.ro';
        }
        this.submit();
    });
};
```

Figure 4. Code injection

This code finds the form and modifies the merchant account so the victim is actually sending money to the attacker.

2.2 Cross-site-scripting

Cross-site-scripting is a code injection attack that allows an attacker to execute malicious JavaScript in another user's browser.
The attacker does not directly target a specific victim. Instead, he exploits a website's vulnerability that the victim visits, in order to get the website to deliver the malicious JavaScript for him. To the victim's browser, the malicious JavaScript appears to be a legitimate part of the website, and thus acted as an unintentional accomplice to the hacker.

2.2.1 The consequences of malicious JavaScript

Among many other things, the ability to execute arbitrary JavaScript in another

ISSN 2067 – 4074

user's browser allows an attacker to perform the following types of attack:

- Cookie theft: the attacker can access the victim's cookies associated with the website, send them to his own server and use them to extract sensitive information like session ID, which can be used to enter PayPal's website (PayPal uses cookies when a user loges in with his account) as though he was the actual owner of the account
- Keylogging: the attacker can register a keyboard event listener, and then send all of the user's keystrokes to his own server, potentially recording personal or financial information
- Phishing: the attacker can insert a fake login form into the page, set the form's action attribute to target his own server and then trick the user into submitting sensitive information.

For security purposes PayPal introduced an option for using a PayPal Security Key during the login process. This security measure is actually an OTP authentication system that generates a random temporary security code (which is displayed on the PayPal Security Key card) that must be used together with the account's holder username and password to complete the sign in process.

2.3 The SSL Man in the middle attack

In this case, the attackers intrude into the network and establish a successful man in the middle connection. The attackers can watch the HTTPS traffic and wait for the targeted website to respond to some browsers HTTPS request. When a browser sends a HTTPS request, the server responds by sending its digital certificate as part of the SSL handshake protocol. At this moment, the attackers can grab this certificate and note down various information as the domain name, expiration date etc. The attackers can now create their self-signed certificate using the information from above. From this point forward, the attackers intercept each browser request and respond with the fake certificate. As a normal response to this situation, the browser pops up a warning to the user, which in most cases is ignored and therefore the attack was successful. Further, the attackers establish a separate HTTPS connection with the server to complete the request, and the result of the response is delivered to the victim's browser. This gives the attackers full control over the SSL traffic and helps them steal personal information. Since the attackers are not breaking the request-response chain, this kind of attack becomes tough to detect the data theft.

2.3.1 NSA Man in the middle

NSA (National Security Agency) is known for using man in the middle attack. Because NSA has a secret partnership with the US telecoms companies, NSA placed secret servers at key places in the internet backbone. This placement ensured that they can react faster then any other websites can. Exploiting these speed differences these servers could impersonate a visited website to the victim before the legitimate website could respond, thereby tricking the victim into thinking they are on the right website. This kind of attack is very difficult to execute for any other attackers because it requires a privileged position in the internet backbone. NSA uses these fast servers to execute a packet injection attack that redirects the victim to one of the NSA servers.

2.4 Packet Crafting

Packet crafting is a task that is methodically carried out to penetrate into a network's infrastructure. There are four steps involved in crafting a packet:

- Packet assembly;
- Packet editing;
- Packet playing;
- Packet analysis.

2.4.1 Paced assembly

This is the first step in the crafting process, where an attacker decides which network needs to be cracked, tries to gather possible vulnerability information and creates the packets to be sent. The packet is then checked for accuracy, especially to ensure that the attack is as "invisible" on the network as possible, to go undetected.

2.4.2 Packet editing

In this step, usually a dry run on the assembled packet is tested and based on the results gathered, the packet is corrected before moving to the next step. In the editing phase, the focus is usually to gather the maximum amount of information by injecting the minimum number of packets into the network.

2.4.3 Packet playing

Once the correct packet or a stream of packets is created, "packet playing" sends it onto the network, and collects the resultant packets to perform further analysis and correlation. This is the step where the actual attack is performed. If the attack is not successful, the attackers go back to editing phase to change the attack scenario.

3. Problem Solution

3.1 Protecting against phishing attacks

Some attackers will place a fake browser address bar over the real one, so it will appear as though you are on the real PayPal website. But even if the URL contains the word "*paypal*" it may not be a PayPal site. Some examples of fake PayPal addresses:

- http://signin.paypal.com@10.19.32.4/
- http://83.16.123.18/pp/update.htm?=
- https://www.paypal.com/=cmd_login_ success;
- www.secure-paypal-com.

Always log in to PayPal by opening a new browser and type www.paypal.com and make sure you are using HTTPS although the PayPal website automatically redirects you to the HTTPS version of the site.
Another measure of protection against phishing attacks is to make sure that the certificate is issued by a trusted third party.

3.2 Protecting against XSS attacks

There are several ways to protect yourself against XSS attacks:
Try and keep the system and the applications up-to-date with patches and updates that are securely configured and meant to protect you from malware.
Allowing all JavaScript to run opens a user to XSS attacks. Of course restricting all JavaScript from a webpage can result in a poor user experience, but you can restrict the JavaScript that comes from domains that you don't trust.
Nowadays, browsers come with some kind of XSS filters to protect users against malicious websites. These extra security measures should be enabled when available.
For developers, there are two basic techniques used to sanitize data: blacklisting and whitelisting.

3.2.1 Blacklisting

It seems reasonable to perform validation by defining a forbidden pattern that should not appear in user input. If a string matches the pattern, it is marked as invalid. An example would be to allow users to submit custom URLs with any protocol except "JavaScript:".
Blacklisting has two major drawbacks:

- Complexity: accurately describing the set of all possible malicious strings is a very complex task;
- Staleness: even if a perfect blacklist is developed, it would fail if a new feature allowing malicious use would be added to the browser.

As pointed above, blacklisting as a strategy is highly discouraged. Whitelisting is a much safer approach as I will describe below/

3.2.2 Whitelisting

Whitelisting is essentially the opposite of blacklisting: instead of defining a forbidden pattern, a whitelist approach defines an allowed pattern and marks input as invalid if does not match the pattern.
In contrast with the blacklisting example before, an example of whitelisting would be to allow users to submit custom URLs containing only the protocols "http:", "https:" nothing else. This approach would automatically mark a URL as invalid if it would have the protocol "JavaScript:"

ISSN 2067 – 4074

Compared to blacklisting, there are two major advantages of using whitelisting:

- Simplicity: accurately describing a set of safe strings is generally much easier then identifying the set of all malicious strings;
- Longevity: unlike a blacklist, a whitelist will generally not become obsolete when a new feature is added to the browser.

3.2.2 Content Security Policy

The disadvantage of protecting against cross-site-scripting attacks by using secure input handling is that even a single lapse of security can compromise the website. A recent web standard called Content Security Policy (CSP) can mitigate this risk.

CSP is used to contain the browser viewing your page so that it can only load resources from trusted sources. This means that even if an attacker succeeds in injecting malicious content into the website, CSP can prevent it from being executed.

By default, browsers do not enforce CSP. To enable CSP on a website, pages must be served with an additional HTTP header: "Content-Security-Policy". Any page served with this header will have its security policy respected by the browser loading it, providing that the browser supports CSP.

3.3 Protecting against Packet Crafting

A common user would probably not even recognize packet injection attack, but it is the job of the server administrator to protect the webserver against such attacks.

To protect the Web servers, the administrator should implement the latest content filtering firewall which is capable of performing a state full packet inspection, and is equipped to detect and prevent denial of service attacks. Configuring firewalls, switches and routers properly to prevent networks from crafting attacks. Packet crafting attacks typically can happen outside the firm's local area network, which demands a carefully designed perimeter defence security system for network infrastructure.

4. Conclusion

When you make transaction with PayPal over the internet be sure that the website has a SSL certificate, this way you are sure that the site is secured and verified by a trusted third party. Make sure that the PayPal website is indeed PayPal by verifying that the certificate is indeed issued to PayPal.

Also you can make sure that the websites you visit have defined a CSP

Acknowledgement

Parts of this paper were presented at The 7th International Conference on Security for Information Technology and Communications (SECITC 2014), Bucharest, Romania, 12-13 June 2014.

References

[1] Damon Williams, Pro PayPal E-Commerce, 2007, pp. 55-65.

[2] Deborah Morley, Charles Parker, Understanding Computers Today and Tomorrow, 2013, 14th Edition, pp. 443-445

[3] Javier Lopez, Bernhard Hämmerli, Critical Information Infrastructures Security: Second International Workshop, CRITIS 2007, Benalmadena-Costa, Spain, October 3-5, 2007

[4] Jeremiah Grossman, Robert Hansem, Petko D.Petkov, XSS Attacks Cross Site Scripting Exploits and Defense, Syngress Publishing Inc., 2007

[5] Michelle Savage, The PayPal Official Insider Guide to Internet Security: Spot Scams and protect your web business 2013

[6] Mark Gregory, David Glance, Security and the Networked Society 2013

[7] Sudahanshu Kairab, A Practical Guide to Security Assessments, 2004

[8] https://www.paypal.com/us/webapps/mpp/security/what-is-phishing#spoof_websites

[9]http://www.nsa.gov/ia/_files/factsheets/XSS_IAD_Factsheet_Final_Web.pdf

[10] Lech Janczewski, Andre M. Colarik, Cyber Warfare and Cyber Terrorism, Idea Group Inc, 2008