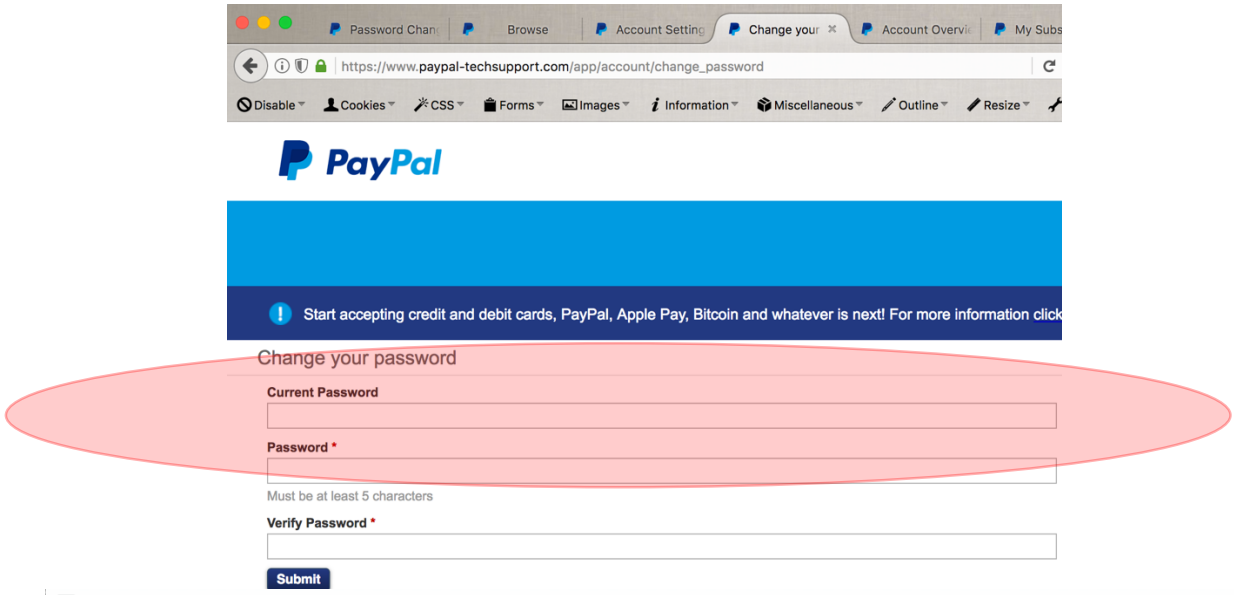
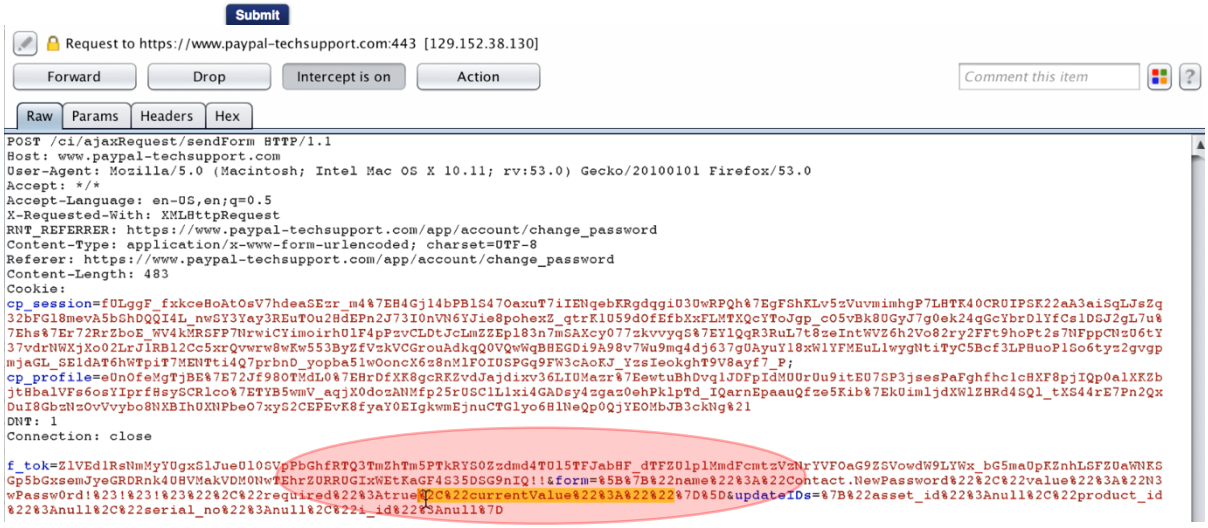
Bypassing the Current Password Protection at PayPal Tech-Support Portal





Dec 26th, 2017

@YoKoAcc (yk@firstsight.me)

[English Version]

Revision Detail

Version	Date	Detail
0.1	Jul 09th, 2017	Initial Report to PayPal
0.2	Aug 03rd, 2017	Sent the report with the .pdf format (previously was sent in general text format via PayPal Bug Bounty Program)
0.3	Dec 26th, 2017	• Added the lesson learned section (as references); • Added the explanation from the proof of concept video; • Added the additional note for the rewarding timeline; • Change the report structure from the original one that sent to PayPal

Table of Contents

Table of Figures

List of Table

I. ABSTRACT

As could be seen previously at another report, for completing the support to all of PayPal's merchant, PayPal provides the technical support portal (located at: https:/www.paypal-techsupport.com) for their merchant to communicate each other when they would like to discuss about the integration, feedback about the needed new feature, or any technical issue that could be face by PayPal's merchant.

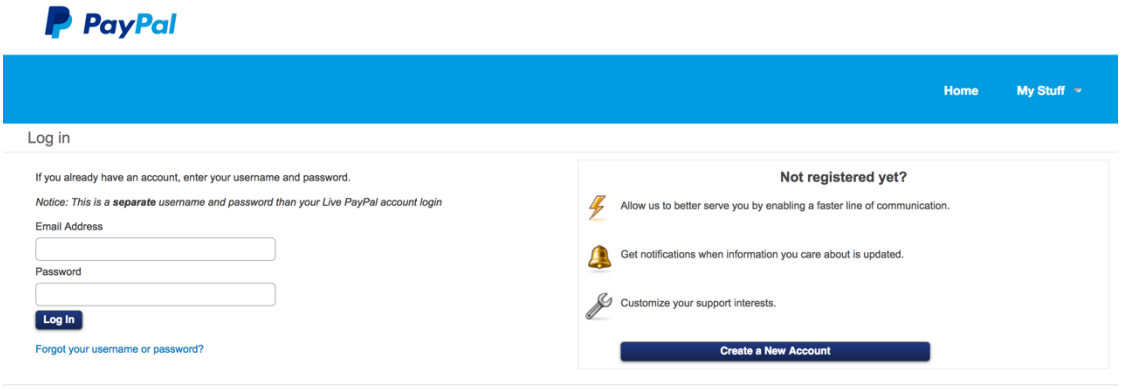


Figure 1 PayPal Technical Support Portal

Just like a common support portal, for facilitating their customer to track the issue, PayPal providing the feature that could be used by their customer to registering themselves with their own account. With the ability to create their own account with their own credential (password), generally users will meet one of the famous common feature such as "Change Password" feature.

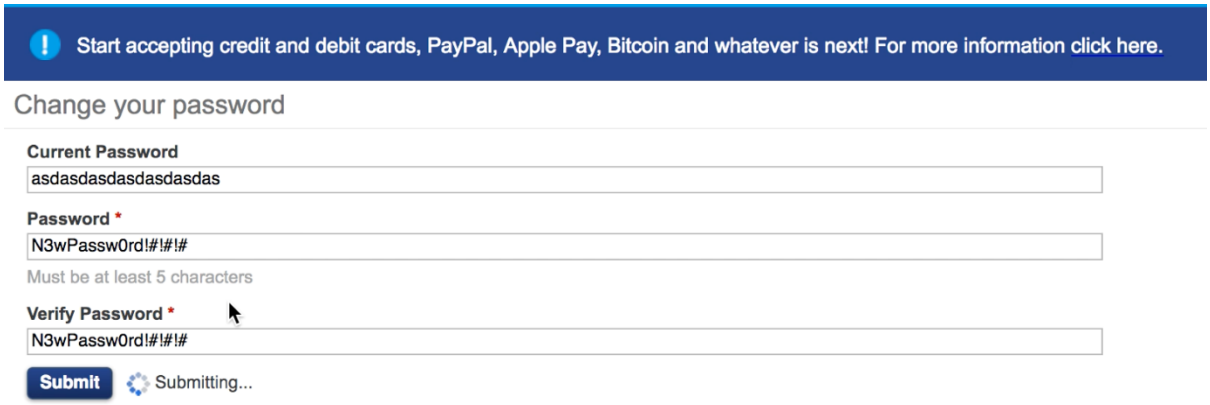


Figure 2 Change Password Feature

But the problem exists when the "Change Password" feature didn't works well to protecting the customer from unauthorized changes. In this case, the Attacker could bypass the "Current Password" Protection feature at application to change the victim's password. In other words, without supply the current password / the knowledge of current password, the Attacker could change the victim's password.

II. INTRODUCTION

Not much thing that could be explain at this part since this we are very sure if the readers are really familiar with the “change password feature” that protected by “current password” field. By this consideration, then we could go directly about the flow that provides by PayPal to use this feature.

The change password feature at the support portal could be found at https://www.paypal-techsupport.com/app/account/change_password or commonly we could see this feature from clicking "My Stuff" URL at https://www.paypal-techsupport.com/app/account/overview.

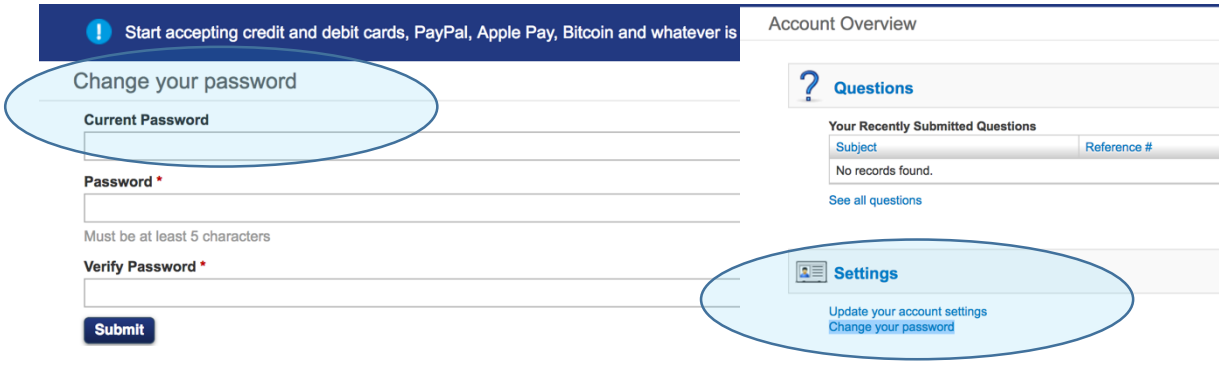


Figure 3 Change Password Feature (Left) and the Location of the Feature (Right)

When user trying to change their Password, normally the application will send a request into https://www.paypal-techsupport.com/ci/ajaxRequest/sendForm with several POST Parameter. Here is the example of the request:

```
POST /ci/ajaxRequest/sendForm HTTP/1.1
Host: www.paypal-techsupport.com
Accept: */*
REDACTED
Content-Length: 477
Cookie: <cookies_over_here>
Connection: close

f_tok=ZlVacmlsQ05nSjV0X3JWU0t4d21QRGYzRGJia3J0WHpGUGJpZWczSWNFTHVmRnpHMm56aUl_U05zUFZtbUdQTnhBM29CaXFSUUptbERpU2NGWmlQZWtjSFZtRzRVUzZKaDlnZDF_Z05sR2ZRU01lblVvQUxIdE05NWxnSjFNRVR_b0pveDRJQzNrU0tnIQ!!&form=%5B%7B%22name%22%3A%22Contact.NewPassword%22%2C%22value%22%3A%22Passw0rd!%23%25!%23%25%22%2C%22required%22%3Atrue%2C%22currentValue%22%3A%22Passw0rd!%23%25%22%7D%5D&updateIDs=%7B%22asset_id%22%3Anull%2C%22product_id%22%3Anull%2C%22serial_no%22%3Anull%2C%22i_id%22%3Anull%7D
```

Table 1 Request for Password Changes

If we try to decode the POST parameter, then we will find it like this:

f_tok=ZlVacmlsQ05nSjV0X3JWU0t4d21QRGYzRGJia3J0WHpGUGJpZWczSWNFTHVmRnpHMm56aUl_U05zUFZtbUdQTnhBM29CaXFSUUptbERpU2NGWmlQZWtjSFZtRzRVUzZKaDlnZDF_Z05sR2ZRU01lblVvQUxIdE05NWxnSjFNRVR_b0pveDRJQzNrU0tnIQ!!&form=**[{"name":"Contact.NewPassword","value":"Passw0rd!#%!#%","required":true,"currentValue":"Passw0rd!#%"}]&updateIDs={"asset_id":null,"product_id":null,"serial_no":null,"i_id":null}**

Table 2 Decoded POST Parameter

As we could see from those full decoded parameter, there is a common interesting part at the "Form" parameter. There is "**currentValue**" parameter that act as the parameter to receive the input of previous password that type by user to change their password. In those decoded value, it tells us if the current password is "**Password!#%**" and the new password is "**Password!#%!#%**".

The problem in this situation is: if we remove the "**currentValue**" parameter and leave it only "**Contact.NewPassword**", then the application still processing the request and change the user's password without the needs to validating the current password.

III. PROOF OF CONCEPT

As stated earlier, we should remove the "**currentPassword**" parameter completely to executing this PoC. Please note, the "complete" word in here means the **%2C%22currentValue%22%3A%22%22** parameter (which is: **,"currentValue":""**)

Here are the step by step to reproducing the issue:

3.1. The first one is put a random password at the “Current Password” Field. For example, **asdasdasdasdasdasdas**.

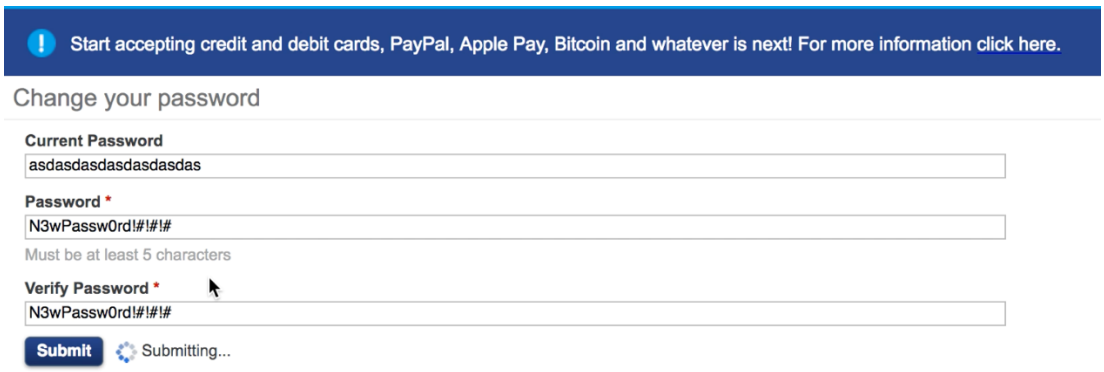


Figure 4 Submitting Random Old Password

3.2. The second one is put the new password at the rest of the field just like the picture above;

3.3. Setting up the Burpsuite interceptor and make it “on” so the request could be intercept later;

3.4. Back to the browser and send the request by click the “submit” button or press the “enter” key;

3.5. After the request has been sent, then go to the interceptor again and see the request.

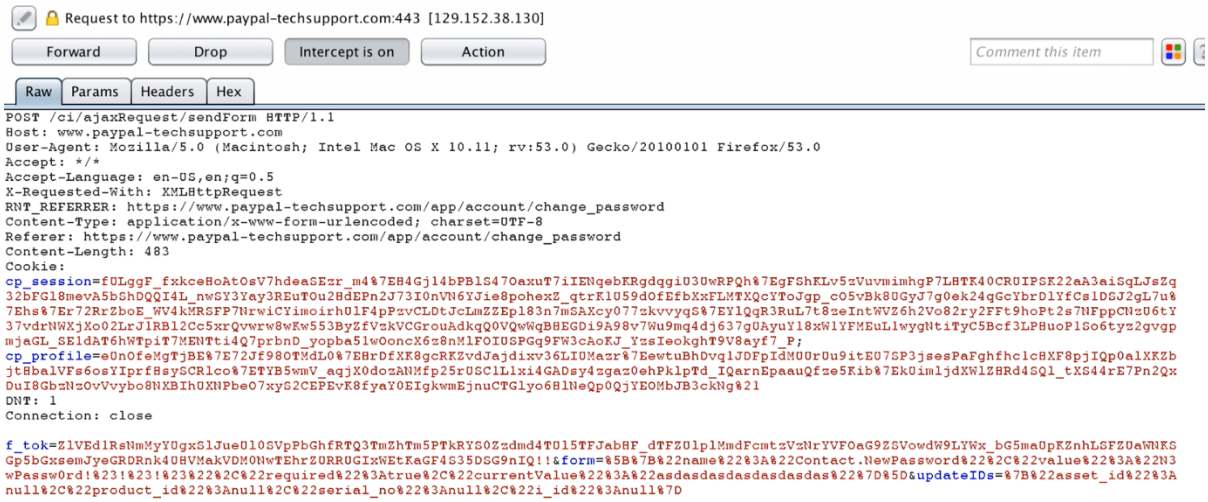


Figure 5 Hold the Request with Interceptor

As we could see, there is a “**currentValue**” parameter with the asdasdasdasdasdasdas value. It could be seen with: **%2C%22currentValue%22%3A%22asdasdasdasdasdasdas%22**

So, all the things that we should conduct is remove completely those parameter from **%2C%22currentValue** (which is **,"currentValue**) until **asdas%22** (which is **asdas"**).

If we trying to decode the POST Parameter, then it just leave this:

```
&form=[{"name":"Contact.NewPassword","value":"N3wPassw0rd!#!#!#","required":true}]&u
pdateIDs={"asset_id":null,"product_id":null,"serial_no":null,"i_id":null}
```

Table 3 Modify the POST Parameter

Please kindly note: When we send the modify request, the application will showing an error. But it doesn’t matter since at the backend process, the password has been changed completely.

IV. PROOF OF CONCEPT VIDEO

For completing the explanation, we upload the ~~unlisted~~ video at Youtube that could act as Proof of Concept related this report: https://www.youtube.com/watch?v=QGBpjDDs9pY

As a support of explanation, here are some information that could be helpful to looking the video:

4.1. The account of the victim is circle.idts2@hotmail.com

4.2. The current password is Sup3rN3wPassw0rd!#

4.3. Attacker put a random password, which is asdasdasdasdasdasdas;

4.4. Attacker tries to change the password into the new one, which is N3wPassw0rd!#!#!#

4.5. Attacker send the request and intercepting it with interceptor;

4.6. Attacker remove the **asdasdasdasdasdasdas** value from the request;

4.7. Attacker remove the **%2C%22currentValue%22%3A%22%22** parameter from the request;

4.8. Attacker send the request to server;

4.9. Application shows an error;

4.10. Attacker refresh the application and get logout automatically;

4.11. Attacker tries to login with the new password that made by the Attacker itself;

4.12. Attacker success to login.

V. LESSON LEARNED

One of the very useful lesson to be learned is we should try to spare our time to read any research that conduct by another researcher. As an information, trick was inspired by the both of research that conduct by Henry Hoggard (PayPal 2FA Bypass) and Suleman Malik (Password Validation bypass at Blackberry).

VI. ADDITIONAL NOTE

The initial bounty was sent on August 10th, 2017. And the final bounty was sent on October 27th, 2017.