Subject: Safe web authentication
Posted by dolik.rce on Sat, 21 Feb 2015 21:44:40 GMT
View Forum Message <> Reply to Message

Hi everyone!

TL;DR version: Looking for a method of secure authentication on insecure network to be implemented in U++ Skylark.

After writing a couple projects in Skylark, I found myself again and again thinking about the same thing: How to handle authentication?

There is currently no support for user registration and login in Skylark, even though it is a very common task. I had to implement it couple times, every time slightly different, but I never 100% liked it. So that is problem number one... The solution to this is actually quite simple, I could easily take one of my implementations, turn it into simple package, let's say Skylark/Login, and let everyone reuse it instead of reinventing the wheel all the time.

The real problem - let's call it problem number two - is what exactly should this package do. As I already said, I implemented various solutions and many more authentication algorithms exist, some of which I probably never heard of. I guess that to have something universal, it should be actually simple. The first thing that comes to mind is probably the most common scheme: client sends password, server combines it with some salt and computes a hash, which is then compared to the value stored in database.

This is simple to implement, but it doesn't seem secure enough to me. First of all, the password is send to the server. This is usually solved by using HTTPS, but that does not always guarantee safety (especially after the recent Lenovo Superfish scandal). So I'd be much happier to use some more advanced challenge-response algorithm, possibly encryption based one, which doesn't require to send any sensitive information over the network. There is many such protocols, but even after hours of googling and reading cryptography articles I haven't found anything that could be easily used in web environment. Most of the algorithms rely on preshared secret keys and do not address the problem of user registration.

It really bugs me that I can't figure this out. So I'm finally getting to the point of this post: I'd like to ask for your opinions and/or tips on how to implement the "Ultimate authentication"? (pun intended).

To sum it up, I'm looking to implement user registration and login procedure that: Can be used in regular web browser (that is only html and javascript on the client side) Doesn't send password (or its equivalent) with the request, so it can be used on unsafe networks Is not vulnerable to MITM, replay and similar attacks Doesn't require user to use other services or devices (that rules out oauth, OTP tokens etc.)
Your ideas are more than welcomed

Best regards,
Honza

Subject: Re: Safe web authentication
Posted by mirek on Sun, 22 Feb 2015 10:20:43 GMT
View Forum Message <> Reply to Message

dolik.rce wrote on Sat, 21 February 2015 22:44Hi everyone!

TL;DR version: Looking for a method of secure authentication on insecure network to be implemented in U++ Skylark.

After writing a couple projects in Skylark, I found myself again and again thinking about the same thing: How to handle authentication?

There is currently no support for user registration and login in Skylark, even though it is a very common task. I had to implement it couple times, every time slightly different, but I never 100% liked it. So that is problem number one... The solution to this is actually quite simple, I could easily take one of my implementations, turn it into simple package, let's say Skylark/Login, and let everyone reuse it instead of reinventing the wheel all the time.

The real problem - let's call it problem number two - is what exactly should this package do. As I already said, I implemented various solutions and many more authentication algorithms exist, some of which I probably never heard of. I guess that to have something universal, it should be actually simple. The first thing that comes to mind is probably the most common scheme: client sends password, server combines it with some salt and computes a hash, which is then compared to the value stored in database.

This is simple to implement, but it doesn't seem secure enough to me. First of all, the password is send to the server. This is usually solved by using HTTPS, but that does not always guarantee safety (especially after the recent Lenovo Superfish scandal). So I'd be much happier to use some more advanced challenge-response algorithm, possibly encryption based one, which doesn't require to send any sensitive information over the network. There is many such protocols, but even after hours of googling and reading cryptography articles I haven't found anything that could be easily used in web environment. Most of the algorithms rely on preshared secret keys and do not address the problem of user registration.

It really bugs me that I can't figure this out. So I'm finally getting to the point of this post: I'd like to ask for your opinions and/or tips on how to implement the "Ultimate authentication"? (pun intended).

To sum it up, I'm looking to implement user registration and login procedure that:
Can be used in regular web browser (that is only html and javascript on the client side) Doesn't send password (or its equivalent) with the request, so it can be used on unsafe networks Is not vulnerable to MITM, replay and similar attacks Doesn't require user to use other services or devices (that rules out oauth, OTP tokens etc.)

Your ideas are more than welcomed

Best regards,
Honza

First things first: It would be nice to have google login integrated with this (possible even others, but google is essential).

As for safety, I guess 'standard' is to send salt (random string) from server (server remembers it), append password to this salt in client and send back hash of whole (I am afraid that means using javascript and having some safe has library available - I guess SHA1 is not considered safe anymore, so perhaps first step is to find some nice new hash, perhaps SHA256, add it to Core, find javascript library...)

Mirek

Subject: Re: Safe web authentication
Posted by dolik.rce on Sun, 22 Feb 2015 11:05:35 GMT
View Forum Message <> Reply to Message

mirek wrote on Sun, 22 February 2015 11:20First things first: It would be nice to have google login integrated with this (possible even others, but google is essential).
It should not be hard to add option to login using Google, Facebook or other service. I guess most of them use oauth, right?

mirek wrote on Sun, 22 February 2015 11:20As for safety, I guess 'standard' is to send salt (random string) from server (server remembers it), append password to this salt in client and send back hash of whole (I am afraid that means using javascript and having some safe has library available - I guess SHA1 is not considered safe anymore, so perhaps first step is to find some nice new hash, perhaps SHA256, add it to Core, find javascript library...)

It is not a problem to implement pretty much anything in javascript (e.g.: I saw an RSA implementation just yesterday). The real problem with this approach is that it means the password is stored on the server in plaintext. It could be hashed, but that solves only part of the problem (the case when your database is compromised).

I'd prefer to use an approach where server doesn't know the password at all. As far as I know, the only way to do this is using asymmetric cryptography, where server has only public key of the client and the private key never leaves the clients computer.

Honza

Subject: Re: Safe web authentication
Posted by Oblivion on Sun, 22 Feb 2015 12:22:17 GMT
View Forum Message <> Reply to Message

Hello guys,

I'm not directly interested in skylark, and this maybe off-topic, since Honza ruled out the OauthX protocol, but as for a client-side authentiacation mechanism, I was playing with the OAuth2

protocol and already wrote an experimental Auth package.
Currently it can authenticate using Google Accounts, though the current implementation is somewhat hacky (since we don't have any web/html engine, I had to use workarounds) and not for production environments.
But if you are interested in it, I can redesign tha package it and publish it around this summer. (My main goal is to implement OpenID, which is built on top of OAuth2)

Regards,
Oblivion

Subject: Re: Safe web authentication
Posted by dolik.rce on Sun, 22 Feb 2015 19:17:20 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Sun, 22 February 2015 13:22I'm not directly interested in skylark, and this maybe off-topic, since Honza ruled out the OauthX protocol, but as for a client-side authentiacation mechanism, I was playing with the OAuth2 protocol and already wrote an experimental Auth package.
Currently it can authenticate using Google Accounts, though the current implementation is somewhat hacky (since we don't have any web/html engine, I had to use workarounds) and not for production environments.
But if you are interested in it, I can redesign tha package it and publish it around this summer. (My main goal is to implement OpenID, which is built on top of OAuth2)

Hi Oblivion,

I didn't want to rule out oauth completely. Perhaps I worded it wrong. What I meant is that the login should not force user to use other services. Some users prefer to have accounts separated as much as possible, other choose more convenient login via single service. So having oauth/openid/facebook/twitter/... as optional is perfectly ok for me.

These are (in theory) simple to implement and there is not much to discuss about. I'd be definitely interested to see your code.

Honza
