Securely sending information from a web form to an email address

Asked 12 years, 11 months ago Modified 9 years, 3 months ago

Viewed 3k times Part of PHP Collective

5

Is it possible to have a user enter information into a form on an HTTPS secured site, then send that information using PHP to an email address securely? How do you encrypt the email so it's secure between sending it from an HTTPS site and checking it via HTTPS email? How feasible is this and what are the potential pitfalls?

PHP php security

Share

Improve this question

Follow

edited Jan 27, 2012 at 1:53

deceze ♦

521k ● 88 ● 793 ● 936

asked Jan 27, 2012 at 1:31

swudev

283 ● 5 ● 18

5 A social security number should be treated with the highest security, which involves **not sending them to email addresses.** – animuson ♦ Jan 27, 2012 at 1:36

Good question, I think it should be generalized to "how to send confidential information securely in an email" or "how to

encrypt email" to be of more general use in the future.
– deceze ♦ Jan 27, 2012 at 1:37

As everyone else seems to state, please do not do this. Sending info except from one HDD to another on the same server is bad news for us. People break into computers for this information so why would you ever willingly send it out into the open? – HenryGuy Jan 27, 2012 at 1:41

You *can* send stuff securely in an email by encrypting the content. The question is, how feasible is it to do this while allowing the receiver to decrypt it. I'd be happy if everybody could focus on this question. – deceze ♦ Jan 27, 2012 at 1:44

People know their own SSNs and have no business seeing anyone else's, so what's the point in emailing one?
– FtDRbwLXw6 Jan 27, 2012 at 1:48

2 Answers

Sorted by: Highest score (default)

6

✓

You could encrypt the email with PGP or S/MIME. These will require special support in your client to decrypt. Most webmail providers don't have this (though there may be e.g., Firefox extensions to make it work).

Other than end-to-end encryption like PGP or S/MIME, it really isn't doable, because of email's clear-text legacy:

- It is possible to transmit e-mail between MTAs using TLS (with the SMTL STARTTLS extension). However, there is no way to mark a message as requiring it. You can configure *your* MTA to require STARTTLS, but not anyone else's. (Note that unless you have root, you can't even configure your MTA.)

The default and normal configuration of MTAs will *happily* accept a message over TLS, and relay it plain text.

- Its normal to add relays when needed. In practice, they are frequent. So your message may take a couple of hops before final delivery. You can force TLS for the first hop, but none of the others. You can check `Received:` headers to see how many hops a message took, but that may change at any point (and it may change from message to message for operational reasons).
- Generally speaking, mail while in transit is written to disk. The normal MTA process is accept message, write to disk, confirm receipt, send message to next MTA, wait for confirm, delete from disk. You can certainly configure your MTA to store these on disk encrypted, but generally that's not done.
- Once "delivered", mail is usually stored in plaintext on disk. Your web mail provider probably does so.
- If you ever use a non-webmail client to access the email, its very likely it downloads messages to disk, in plain text. Then you have unencrypted messages sitting on a random desktop PC. The MUA may even transfer messages over the Internet in plain text.
- Email is easy to access anywhere, from any machine. Especially webmail. It will be very tempting for you, or someone else in your company, to log in to the account the messages are sent to from random devices. Can you guarantee *none* of those

> devices has any malware—key logger, session hijacker, screen scraper, etc.—installed? Can you guarantee no user will *ever* ignore the certificate warning, enabling a man-in-the-middle attack?

Except for a few of the very largest webmail providers, they probably don't have sufficient security to protect confidential identity data. E.g., if you call them up and say you forgot your password, how do they handle that? Or, if someone *else* calls them up and pretends to be you.

Lastly, even with PGP or S/MIME, you're left with the problem that email, in the real world, can and does get lost. Your site may send a message, get confirmation that the message is accepted for delivery, and then that message just never arrive—not even to a spam folder. That generally isn't acceptable for valuable data.

note: If you encrypt the message with PGP or S/MIME, using sensible settings (e.g., 3072-bit or higher RSA keypair, AES cipher), then it doesn't matter if the email message (ciphertext) is disclosed; it is unreadable without the private key. The crypto used is fairly similar to TLS (but unfortunately doesn't offer things like perfect forward security as its an offline protocol). End-to-end crypto protects against insecurities between the endpoints. Not that I'd advocate printing the cihertext in a newspaper, but even that should still be secure. You must, of course, keep the machine which holds the private key (and decrypts the emails) secure.

Share Improve this answer

Follow

edited Jan 27, 2012 at 2:25

answered Jan 27, 2012 at 1:58

derobert

51k ● 15 ● 97 ● 126

When you are saying "message is stored and transferred in plaintext", that still means the information itself is encrypted using PGP or S/MIME. It'd be helpful to say something about the chances of cracking that encryption, should somebody gain access to the encrypted blob through the "plaintext" storage. – deceze ♦ Jan 27, 2012 at 2:11

1 @deceze: OK, I think I've addressed that in my "note:" paragraph at the end. – derobert Jan 27, 2012 at 2:20

2

Only send the last four digits. Such as !!!!-&&-8590. Send them a secure link to view the Number online if necessary. Also remember that its unlawful in some areas to store Social Security Numbers or any sensitive information in plaintext, always encrypt or encode your data.

Share Improve this answer

Follow

answered Jan 27, 2012 at 1:51

Adam Fowler

1,751 ● 1 ● 17 ● 19