

Re: socketServer questions

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2005-10/msg01237.html>

- *From:* rbt <rbt@xxxxxxxxxxxxxxxxxxxx>
 - *Date:* Mon, 10 Oct 2005 10:19:55 -0400
-

On Mon, 2005-10-10 at 05:54 -0700, Paul Rubinhttp: wrote:

> rbt <rbt@xxxxxxxxxxxxxxxxxxxx> writes:
>>> I don't understand the question. HMAC requires that both ends share a
>>> secret key; does that help?
>>
>> That's what I don't get. If both sides have the key... how can it be
>> 'secret'? All one would have to do is look at the code on any of the
>> clients and they'd then know everything, right?
>
> Yes, clients have to keep the key secure.
>
>>> What do you mean by verification?
>>
>> I'm trying to keep script kiddies from tampering with a socket server. I
>> want the server to only load a valid or verified string into its log
>> database and to discard everything else.
>
> If the clients can keep a secret key secure, then use hmac. Note that
> if there's lots of clients, they shouldn't all use the same secret key.
> Instead, for client #i, let that client's key be something like
> hmac(your_big_secret, str(i)).digest()
> and the client would send #i as part of the string.

How is this different from sending a pre-defined string from the client that the server knows the md5 hash of? The clients know the string, the server knows the hash of that string.

Also, could this not be done both ways? So that, if an attacker figures out the string he's supposed to send from a client to the server (which he could easily do). He could not easily figure out the string the server should send back as all he would have is the hash of that string.

So, before the actual data is sent from the client to the server. The client would send it's secret string that the server would verify and then if that worked, the server would send its own secret string that the client must verify. We'd have two secret strings instead of one.

> You'd use

Re: socketServer questions

- > #i to recompute the client's key and then use that derived key to
- > verify the string. This is called "key derivation" or "key
- > diversification". If an attacker gets hold of that client's key and
- > starts hosing you, you can disable that key without affecting the
- > other ones. (The client is issued only the derived key and never sees
- > the big secret).

This is interesting. I didn't know that was possible.

- >
- >> Strings could come to the socket server from anywhere on the Net from
- >> any machine. This is outside my control. What is there to prevent a
- >> knowledgeable person from finding the py code on a client computer,
- >> understanding it and then being able to forge a string that the server
- >> will accept?
- >
- > Yes, if you're concerned about insecure clients, you have a much more
- > complex problem. But your x..z..y scheme is far worse than hmac.
- > Once the attacker figures that out, there's no security at all.

I dropped the x,y,z scheme after your first response ;)

- >
- > What is the actual application, if you can say? Depending on the
- > environment and constraints, various approaches are possible.

Nothing important. It just logs network data. It's an anti-theft program for laptops that phones home data like this: public and private IP(s), MAC addy, date, time, etc. Maybe I'm putting too much thought into it. Python encourages good design and I try to follow that encouragement when coding... even for trivial things such as this.

.

• *Follow-Ups:*

- ◆ **Re: socketServer questions**
◇ *From: Paul Rubin*

• *References:*

- ◆ **socketServer questions**
◇ *From: rbt*
- ◆ **Re: socketServer questions**
◇ *From: Paul Rubin*
- ◆ **Re: socketServer questions**
◇ *From: rbt*
- ◆ **Re: socketServer questions**
◇ *From: Paul Rubin*
- ◆ **Re: socketServer questions**
◇ *From: rbt*

Re: socketServer questions

◆ [Re: socketServer questions](#)

◇ *From:* Paul Rubin

◆ [Re: socketServer questions](#)

◇ *From:* rbt

◆ [Re: socketServer questions](#)

◇ *From:* Paul Rubin

- Prev by Date: [Re: Merging sorted lists/iterators/generators into one stream of values...](#)
- Next by Date: [TurboGears slashdotted](#)
- Previous by thread: [Re: socketServer questions](#)
- Next by thread: [Re: socketServer questions](#)
- Index(es):
 - ◆ [Date](#)
 - ◆ [Thread](#)