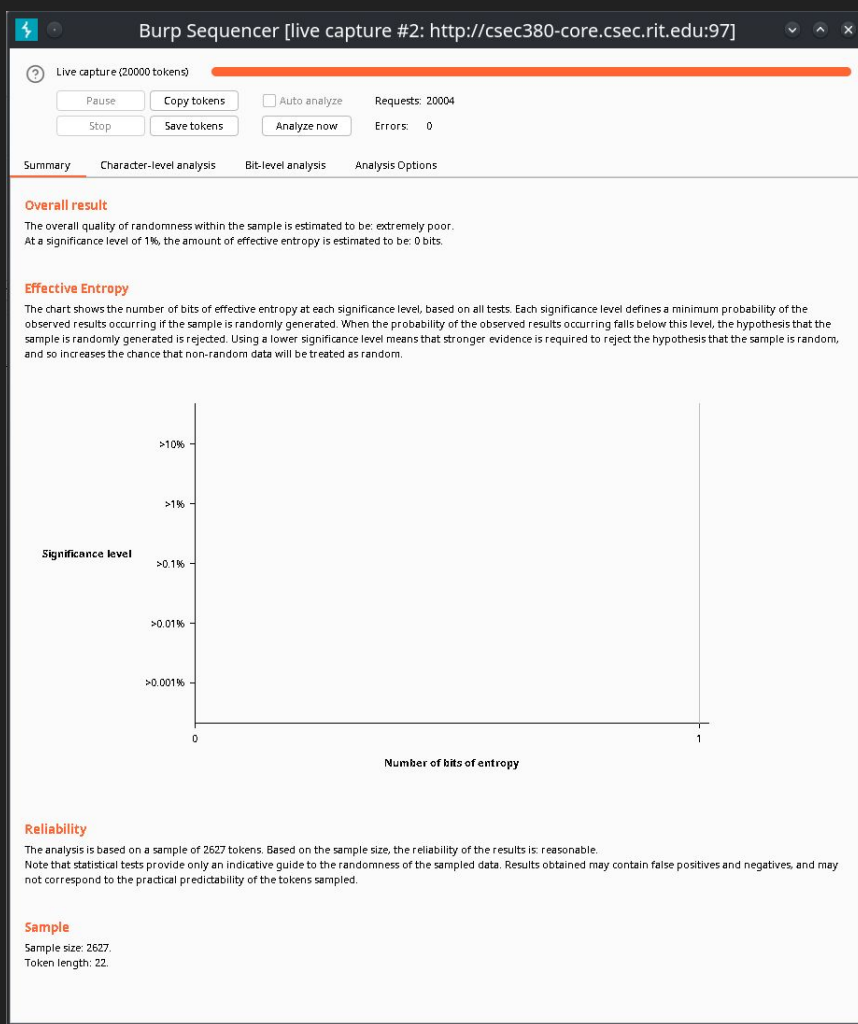


Armbook Session ID

Weakness

Max Fusco



0 bits of entropy

This is a very bad score for randomness, this means that the tokens are very easily guessable

The reason for this is that within a second, the same token will always be given out.

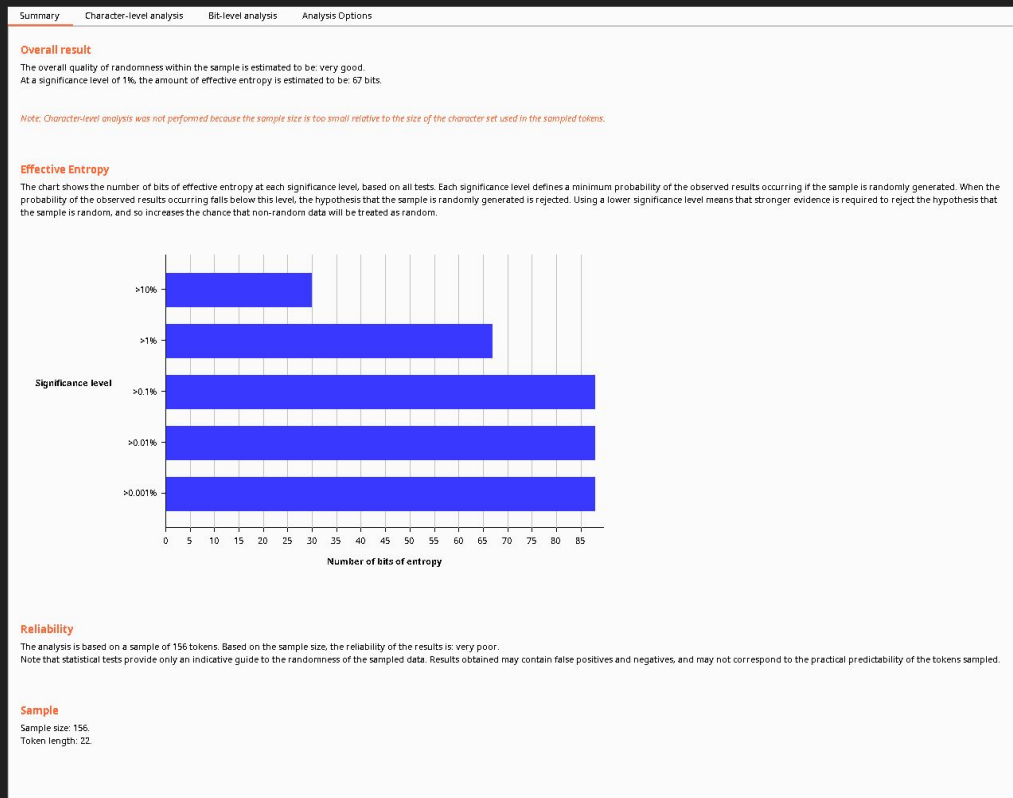
When doing this scan, it can do more than 1 per second so there was several duplicates

Unique Tokens

Generate more tokens and grouping all the tokens found in a single second, we get:

67 bits of entropy

This show the issues isn't with the hash function, but the timing that resulting from having multiple requests per second.




The Underlying Flaw

```
<input type="text" id="email" name="email" />
</td><td>
<input type="password" id="password" name="password" />
<input type="hidden" id="session_id" name="ARM_SESSION" value="<?php if(!isset($_REQUEST["ARM_SESSION"])){echo substr(md5(time()),0,22);}else{echo htmlentities($_REQUEST["ARM_SESSION"]);} ?>" />
</td><td>
<input type="submit" name="submit" value="login" />
```

session_ids are nothing more than the md5 hash of the current time

This current time using time() only is second specific

All users who visit the page within the same second will get the same session_id

Likewise you can generate all session_id from the past x seconds using: 

This can then get you access to all accounts
logged in as you have their session_id

```
<?php
$time = time();
$x = 100;
for ($i = 0; $i <= $x; $i++) {
    echo substr(md5($time - $i),0,22);
    echo "\n";
}
?>
```

Fix

Don't use time()!

Instead use something that incorporate random bytes like random_bytes()

```
<input type="password" id="password" name="password" />
<input type="hidden" id="session_id" name="ARM_SESSION" value="<?php if(!isset($_REQUEST["ARM_SESSION"])){echo substr(md5(random_bytes(22)),0,22);} ?>" />
</td><td>
<input type="submit" name="submit" value="login" />
</td></tr>
```

Also recommended, don't let the client to set their own session_id by including it in a form

Have a session be added server-side and returned to the user