

Underhanded Crypto Contest Results

Contest Recap

Two categories:

1. GnuPG key leaking.
2. Password hashing (or auth.) backdoor.

Announced: July 27th (but yesterday for some).

Deadline: This morning at 8am.

Results

- Few, but high-quality submissions.
 - 3 GnuPG patches.
 - 2 password authentication backdoors.
- All submissions will be online soon.

Thanks Jean-Philippe Aumasson!

The winner of the GnuPG category is...

GnuPG Winner

ctz (Joseph Birr-Pixton)

Summary:

- DSA needs random nonces in $[0, Q)$.
- Non-random nonce = recover private key.
- GnuPG 1.4 does this:
 1. Pick random nonce N . Set MSB.
 2. If $N \geq Q$, re-gen high 32 bits.
- Patch: The nonce needs to be kept secret, so zero it!

```

diff --git a/cipher/dsa.c b/cipher/dsa.c
index e23f05c..e496d69 100644
--- a/cipher/dsa.c
+++ b/cipher/dsa.c
@@ -93,6 +93,7 @@ gen_k( MPI q )
     progress('.');

+     if( !rndbuf || nbits < 32 ) {
+         if (rndbuf) memset(rndbuf, 0, nbytes);
+         xfree(rndbuf);
+         rndbuf = get_random_bits( nbits, 1, 1 );
+     }
@@ -115,15 +116,18 @@ gen_k( MPI q )
     if( !(mpi_cmp( k, q ) < 0) ) {          /* check: k < q
*/
+         if( DBG_CIPHER )
+             progress('+');
+         memset(rndbuf, 0, nbytes);
+         continue; /* no */
+     }
+     if( !(mpi_cmp_ui( k, 0 ) > 0) ) { /* check: k > 0 */
+         if( DBG_CIPHER )
+             progress('-');
+         memset(rndbuf, 0, nbytes);
+         continue; /* no */
+     }
+     break; /* okay */
+ }
+ memset(rndbuf, 0, nbytes);
+ xfree(rndbuf);
+ if( DBG_CIPHER )
+     progress('\n');

```

The winner of the password authentication category is...

Password Auth. Winner

Scott Arciszewski

Summary:

- Problem: User enumeration side channel.
- Fix: Compare password with a random value.
- But... Random value comes from rand().
 - rand() is not cryptographically secure.
 - Some rand() output (so attacker can recover state) is available in cache-busting URL.

```
class TimingSafeAuth
{
    private $db;
    public function __construct(\PDO $db)
    {
        $this->db = $db;

        $garbage = noise();

        $this->dummy_pw = password_hash($garbage, PASSWORD_DEFAULT);
    }

    // Returns the user's user ID, or false.
    public function authenticate($username, $password)
    {
        $stmt = $this->db->prepare("SELECT * FROM users WHERE username = :username");
        if ($stmt->execute(['username' => $username])) {
            $row = $stmt->fetch(\PDO::FETCH_ASSOC);
            // Valid username
            if (password_verify($password, $row['password'])) {
                return $row['userid'];
            }
            return false;
        } else {
            // Returns false
            return password_verify($password, $this->dummy_pw);
        }
    }
}
```

```
if ($_SESSION['userid'] == 1) {  
    echo "Welcome great leader!\n";  
    echo "<hr />";  
    echo "Administrative features:  
...";  
} else {  
    echo "Welcome, peon.\n";  
}
```

In PHP, true == 1.

Thanks

- The 5 participants.
- Jean-Philippe for judging.
- Crypto & Privacy Village for giving us a venue.
- You for listening.

Stay tuned for the 2016 contest!

underhandedcrypto.com