**Problem**

<https://github.com/domainmod/domainmod/issues/122>

reset.php handles “resetting”/changing an existing user’s password. The reset functionality uses PHP’s **time()** to derive the new password for the user that is attempting to change their password. This method of using time() to create a temporary password is **very deterministic** and allows an attacker to invoke the reset password functionality and reliably determine what the new password is, thus allowing for an account takeover.

**reset.php**

$new\_password = substr(md5(**time()**), 0, 8);

**Fix**

**reset.php**

new\_password = $user->**generatePassword(30);**

$new\_hash = $user->generateHash($new\_password);

<https://github.com/domainmod/domainmod/blob/986d8198d567471c46fa666b8a5318c76834f877/classes/DomainMOD/User.php#L80>

public function generatePassword($password\_length = 72)

{

$character\_pool = 'abcefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890';

return substr(str\_shuffle($character\_pool), 0, $password\_length);

}

public function generateHash($password)

{

return password\_hash($password, PASSWORD\_DEFAULT);

}

public function setDarkMode()

{

$stmt = $this->deeb->cnxx->prepare("

UPDATE user\_settings

SET dark\_mode = :dark\_mode,

update\_time = :update\_time

WHERE user\_id = :user\_id");

$timestamp = $this->time->stamp();

$stmt->bindValue('dark\_mode', $\_SESSION['s\_dark\_mode'], \PDO::PARAM\_INT);

$stmt->bindValue('update\_time', $timestamp, \PDO::PARAM\_STR);

$stmt->bindValue('user\_id', $\_SESSION['s\_user\_id'], \PDO::PARAM\_INT);

$stmt->execute();

return;

}

} //@formatter:on