nu11secur1ty Update README.MD ... on Sep 7, 2021 History

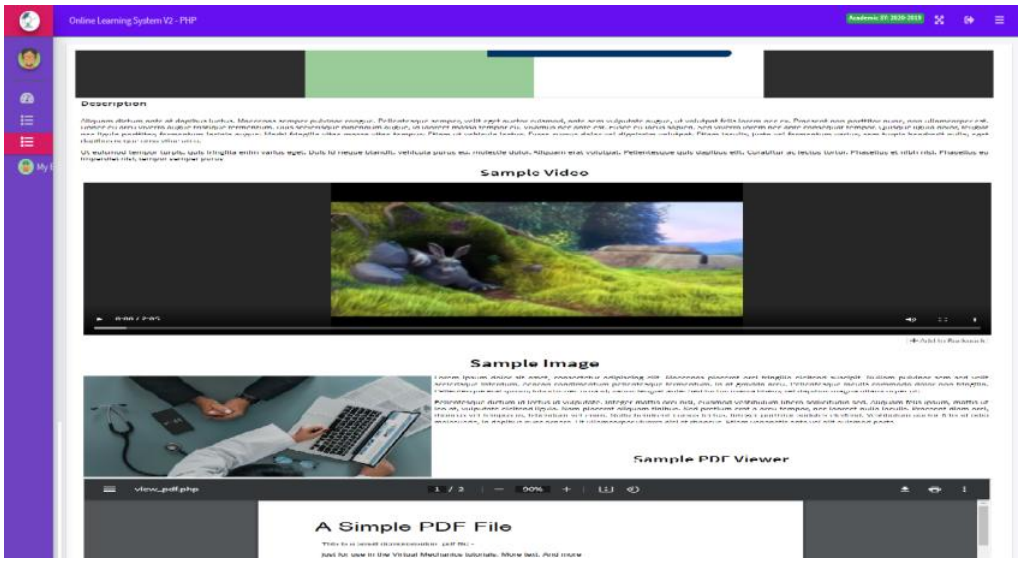
..	
docs	last year
PoC-CVE-nu11-07.py	last year
README.MD	last year
chromedriver.exe	last year
templeate_report.txt	last year
vulnerable-code.txt	last year

README.MD

CVE-nu11-07

VENDOR

- eLearning V2(by: oretnom23) is vulnerable from remote SQL-Injection-Bypass-Authentication



Description:

The eLearning V2(by: oretnom23) is vulnerable from remote SQL-Injection-Bypass-Authentication in 3 accounts of the system (admin, Faculty & Student) in app /elearning/classes/Login.php. remote SQL-Injection-Bypass-Authentication: https://portswigger.net/support/using-sql-injection-to-bypass-authentication. The parameter (username, faculty_id, and student_id) from the login form is not protected correctly and there is no security and escaping from malicious payloads. When the user will sending a malicious query or malicious payload to the MySQL server for those three accounts, he can bypass the login credentials and take control of these accounts.

- Vulnerable `PHP` app code in `/elearning/classes/Login.php`

```
public function login(){
        extract($_POST);

        $qry = $this->conn->query("SELECT * from users where username = '$username' and password = md5('$password') ");
        if($qry->num_rows > 0){
                foreach($qry->fetch_array() as $k => $v){
                        if(!is_numeric($k) && $k != 'password'){
                                $this->settings->set_userdata($k,$v);
                        }

                }
                $this->settings->set_userdata('login_type',1);
        $sy = $this->conn->query("SELECT * FROM academic_year where status = 1");
        foreach($sy->fetch_array() as $k =>$v){
                if(!is_numeric($k)){
                $this->settings->set_userdata('academic_'.$k,$v);
                }
```

```
                }
                return json_encode(array('status'=>'success'));
                }else{
                return json_encode(array('status'=>'incorrect','last_qry'=>"SELECT * from users where username = '$username' and pass
                }
        }
        public function flogin(){
                extract($_POST);

                $qry = $this->conn->query("SELECT * from faculty where  faculty_id = '$faculty_id' and `password` = '".md5($password)
                if($qry->num_rows > 0){
                        foreach($qry->fetch_array() as $k => $v){
                                if(!is_numeric($k)){
                                        $this->settings->set_userdata($k,$v);
                                }

                        }
                        $this->settings->set_userdata('login_type',2);
                        $sy = $this->conn->query("SELECT * FROM academic_year where status = 1");
                foreach($sy->fetch_array() as $k =>$v){
                        if(!is_numeric($k)){
                        $this->settings->set_userdata('academic_'.$k,$v);
                        }
                }
                        return json_encode(array('status'=>'success'));
                }else{
                return json_encode(array('status'=>'incorrect'));
                }
        }
        public function slogin(){
                extract($_POST);

                $qry = $this->conn->query("SELECT * from students where  student_id = '$student_id' and `password` = '".md5($password
                if($qry->num_rows > 0){
                        foreach($qry->fetch_array() as $k => $v){
                                if(!is_numeric($k)){
                                        $this->settings->set_userdata($k,$v);
                                }

                        }
```

- This vendor must STOP creating all these broken projects and vulnerable software programs, probably he is not a developer!
- [+] by @nu11secur1ty System Administrator - Infrastructure and Penetration Testing Engineer

Reproduce:

```
https://github.com/nu11secur1ty/CVE-nu11secur1ty/tree/main/vendors/oretnom23/CVE-nu11-07
```

Proof:

href

BR

@nu11secur1ty