Experiential Learning Workshop on XSS and CSRF

Oct 18, 2019

Dr. Ram P Rustagi
Professor, CSE Dept
KSIT, Bangalore
rprustagi@ksit.edu.in

Resources & Acknowledgements

- Resources
  - https://rprustagi.com/ELNT/Experiential-Learning.html
  - Articles in ACCS Journal
    - https://acc.digital/experiential-learning-of-networking-technologies-4/
    - www.github.com/rprustagi
  - slides: https://www.rprustagi.com/workshops/others
  - Examples of programs
    - https://www.rprustagi.com/workshops/programs
  - Example Web pages
    - https://www.rprustagi.com/workshops/wev

Common Vulnerabilities in Web Appl^{n}

- What is vulnerability?
- How attackers find vulnerabilities
  - Penetration testing
- Common web application vulnerability
  - SQL Injection
  - XSS Attack
  - CSRF attack
  - Broken authentication and session hijacking
  - Insecure configuration (misconfiguration)
  - Direct insecure reference to objects
    - URL manipulation (redirecting to malicious website)

XSS and Javascript Attack

- XSS: Cross Site Scripting
  - Attacker inserts malicious javascripts in web applications e.g. blogs, comments
- Attack types
  - Persistent attacks
  - Non-persistent
    - Client based
  - DOM based attacks

XSS Attack

- S1: Attacker inserts malicious scripts in web server code
- S2: User accesses (browses) web server
- S3: User is silently directed to some other website (controlled by attacker)
  - Cookies of the victims are sent to attacker's site
  - Any other information user is typing

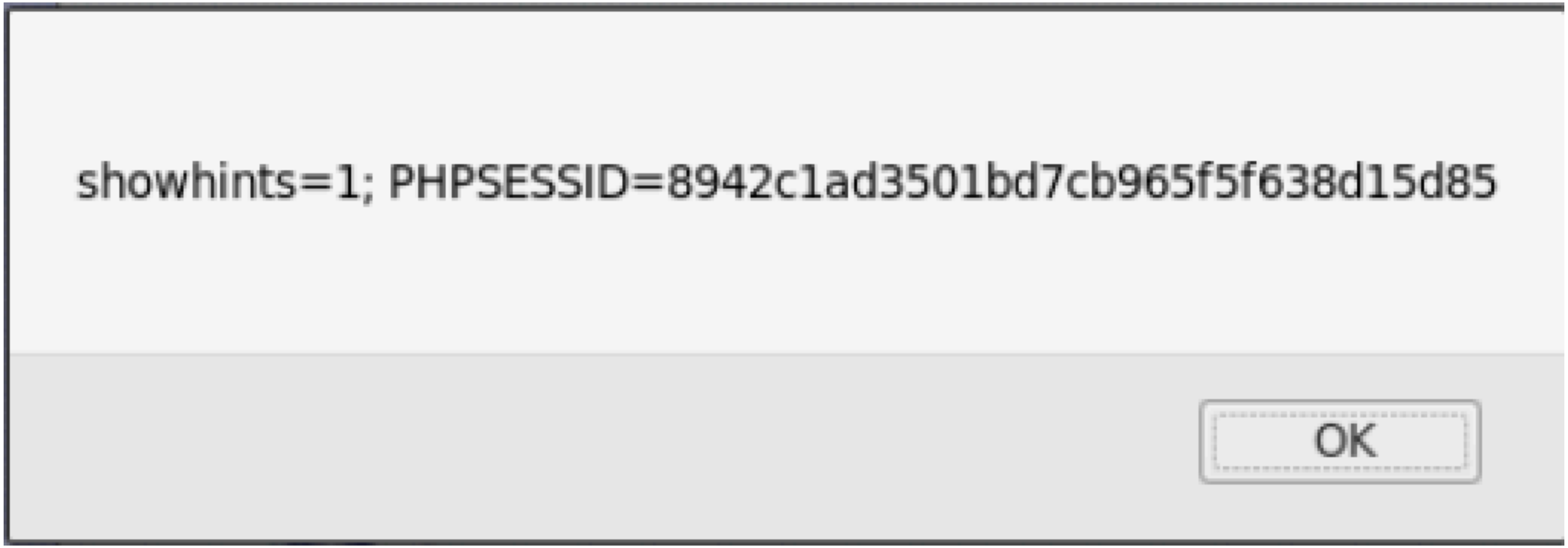
XSS Attack

- Open Kali linux browser
  - Normal browsers are fortified and thus prevent such attacks.
- Access mutillidae web page
  - OWASP 2017—>A7 XSS attacks —> First order —>DNS Lookup
  - Enter some javascript with alert() e.g. `<script>alert(“website hacked”);</script>`
  - Click `DNS Lookup`
  - Should see the alert pop-up, implying website in vulnerable.

XSS Attack

- Access mutillidae web page
  - OWASP 2017—>A7… —>DNS Lookup
  - Enter some javascript with alert() e.g.

    ```
    <script>alert(document.cookie)</
    script>
    ```



```
showhints=1; PHPSESSID=8942c1ad3501bd7cb965f5f638d15d85
```

Bug Bounty program

- https://hackerone.com/bug-bounty-programs
  - A number of websites offers this program.
  - Identify the vulnerabilities and make your money.
  - This is perfectly legal as long as you comply with terms of websites.
  - Examples of website offering this program
    - Android
    - Apache httpd
    - Blogger
    - Chrome
    - cPanel
    - …

XSS Attack: Persistent attack

- Kali Linux browser: Access multillidae
  - OWASP 2017->A7 XSS ->Persistent-Add to Blog
  - Enter following script

```
<script>window.location=”http:/
/www.rprustagi.com”;</script>
```

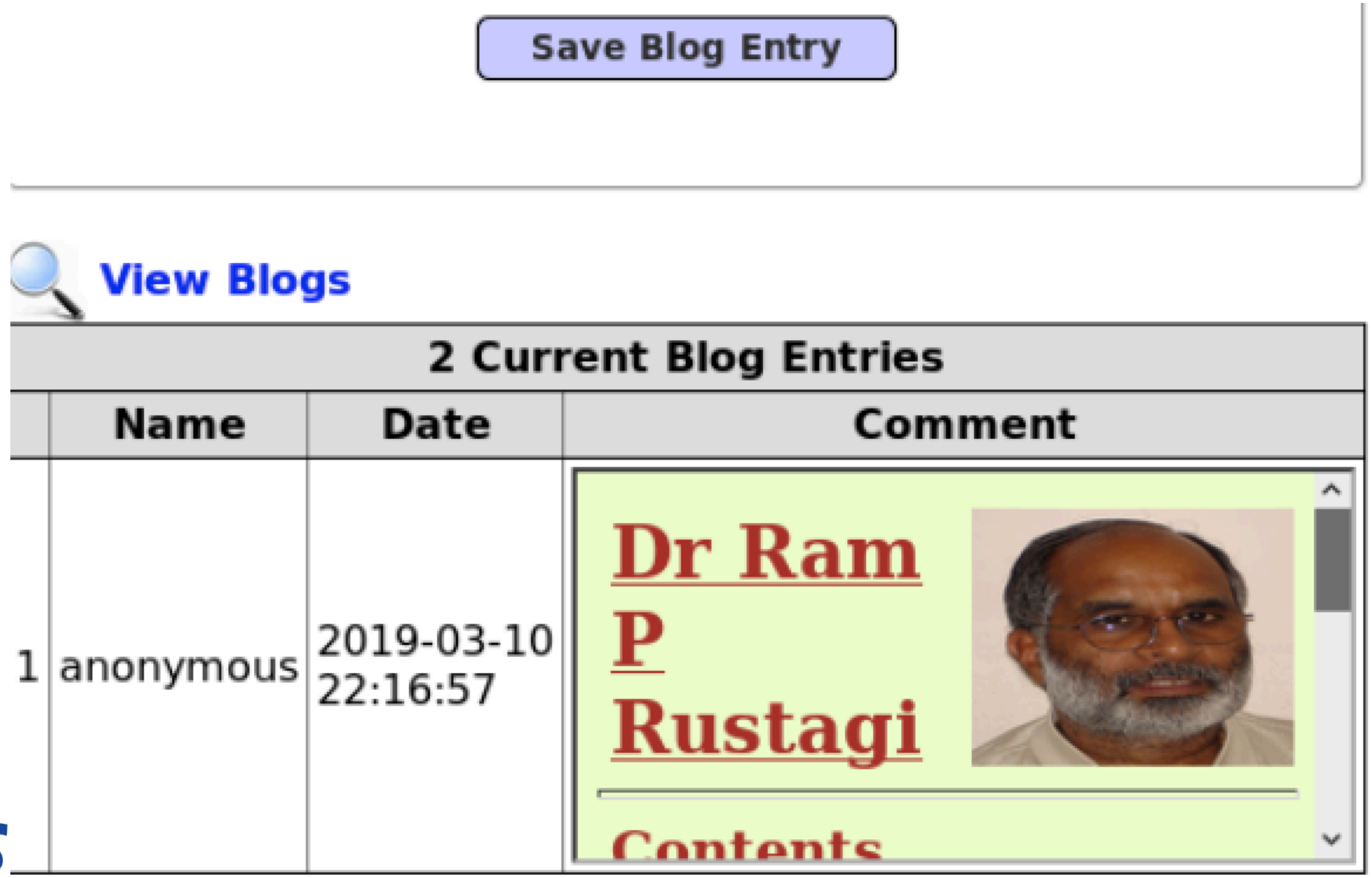
  - Click “Add to Blog”
- Browser is redirected to the specified website.
- Even if you press Browser back button, it comes back to this specified website.
  - Hence it is called persistent attack
- Summary: a user can be directed to any ***phishing*** website

XSS Attack: Persistent attack

- Kali Linux browser: Access multillidae
  - OWASP 2017->A7 XSS ->Persistent-Add to Blog
  - Enter following script

```
<iframe src=”http://www.rprustagi.com” />
```

- Result: content of website is inserted in blogs



2 Current Blog Entries			
	Name	Date	Comment
1	anonymous	2019-03-10 22:16:57	Dr Ram P Rustagi Contents

XSS Attack: Persistent attack

- Using malicious scripts, attacker can play havoc
  - One needs to know javascript extensively.
  - Capture users's email,
    - password,
    - other details.
  -

XSS Attack: Prevention

- XSS attack prevention techniques
  - Front end checking
    - At the front end (browser itself) ensure no unwanted content is entered.
  - Always validate the input.
    - Ensure no unwated information is present, especially no scripting kind of information.
  - Always sanitize the input. Input is along expected lines and no special characters etc,. are used.

Prevention: XSS Attack

- Front end (browser) should check before sending the data to application, prevent following
  - javascript
  - HTML elements
  - URL links
- Input Validation
  - Back end application needs to perform all required checking.
  - User can use command line tool (wget, curl etc) to send the data. User may not use the browser
    - Check for HTML content, javascripts, URLs etc.

Prevention: XSS Attack

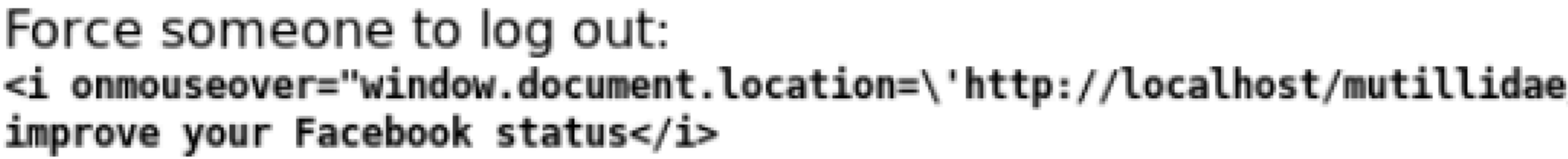
- Data sanitization
  - Today, application accepts all kind of data including HTML text etc.
  - Santization: allow only trusted content.
- Study cheatsheets for XSS attacks
  - https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
  - https://gbhackers.com/top-500-important-xss-cheat-sheet/
  -

CSRF Attacks

- CSRF: Cross Site Request Forgery
  - Attacker executes some unwanted action to trick the user
  - example:
    - A user is trying to transfer some information (e.g. username/password) to another user.
    - Attacker can trick the user to send this information to the her/himself (attacker)
- CSRF attack methodology
  - Attacker redirects the user to malicious website
  - Steals the cookies of the victim
  - Uses these cookies to carry out harmful action

CSRF Attacks: Mutillidae

- Kali linux: access mutillidae
  - Reset the DB
- Register yourself e.g. admin/admin
- Access
  - mutillidae->OWASP 2013->A8 CSRF->Add blog
  - Look at Hints and Videos
  - Look at the script
    - Force someone to logout



Force someone to log out:

```
<i onmouseover="window.document.location=\'http://localhost/mutillidae
improve your Facebook status</i>
```

CSRF Attacks: Mutillidae

- Take the mouse to new blog entry
  - notice that you have been logged out
  - Add to any other website, and you will be taken there.
  - Notice that you are logged out from current account.

Hands On 3:

- Carry out XSS Attacks
- Explore all possibilities of mutillidae
- Use cheatsheet of owasp for XSS attack and carry out these attacks locally on mutillidae or your own developed website
  - e.g. all the web content that you develop for projects.
- Carry out CSRF attacks

CSRF Attack Prevention

- Same site cookies
  - – Server when receives the cookies in a web req,
  - – Verifies that origin of this request is from same IP and Port number (to which cookies were issued)
  - – If request origin does not match
    - web request is rejected,.
  - – One of the best ways to prevent CSRF attack

DoS Attacks

- Goal: Prevent access to application to genuine users
- Methodology
  - – Flood the network: Choke it with too much data
  - – Overload the application (send too many requests)
- Non-technical example: Access to PES University
- Tool: `hping3` (available by default on full kali)
  - – Can be installed on regular linux
    - sudo apt install hping3
- Use regular ping with options `-l 100 -f`
  - –

DoS Attacks

- Use hping3 to cause attacks like
  - SYN Flooding attacks (-S), Simple flooding attacks
  - Random source (—rand-source)
  - Fragmentation attacks
  - Gather TCP sequence number of target hosts (—seqnum)
  - TCP/UDP packets with bad checksum
  - Set FIN flag
- Example
  - hping3 -c 100000 -d 120 -S -w 64 -p 80 --flood —rand-source <some website>
  - hping3 -S --flood -V <some website>

Password Attacks

- use ncrack
- Applications->password attacks -> ncrack
- Full Kali Linux
  - User burpsuite
  - Can filter and replay login with different username./password
- Prevention
  - Use Captcha
  - Different URLs to different users
  - Limited login attempts

Thanks
Question, Comments, Suggestions

Summary

- ?

Thank You