AJAX - Silent Transaction Attacks

Friday, October 20, 2017 7:16 AM

Any system that silently processes transactions using a single submission is dangerous to the client.
ex.
If a normal web application allows a simple URL submission, a preset session attack will allow the attacker to complete a transaction without the user's authorization.

- In AJAX it gets worse: the transaction is silent; it happens with no user feedback on the page, so an injected attack script may be able to steal money from the client without authorization

Solution

- Inspect element on the 'Confirm' button

```
<pre></pre>
<input id="confirm" onclick="processData('http://localhost:8081/WebGoat/attack?Screen=218322538&menu=400');" value="Confirm" name="confirm" t
<br>
```

- As we can see, this calls the processData JavaScript function
- If we open the link to this code in the script tag we see…

```
function processData() {
    var accountNo = document.getElementById('newAccount').value;
    var amount = document.getElementById('amount').value;
    if (accountNo == '') {
        alert('Please enter a valid account number to transfer to.')
        return;
    }
    else if (amount == '') {
        alert('Please enter a valid amount to transfer.')
        return;
    }
    var balanceValue = document.getElementById('balanceID').innerHTML;
    balanceValue = balanceValue.replace(new RegExp('$'), '');
    if (parseFloat(amount) > parseFloat(balanceValue)) {
        alert('You can not transfer more funds than what is available in your balance.')
        return;
    }
    document.getElementById('confirm').value = 'Transferring'
    submitData(accountNo, amount, url);
    document.getElementById('confirm').value = 'Confirm'
    balanceValue = parseFloat(balanceValue) - parseFloat(amount);
    balanceValue = balanceValue.toFixed(2);
    document.getElementById('balanceID').innerHTML = balanceValue + '$';
}

function submitData(accountNo, balance) {
    var url = document.getElementById("url").value;
    url = url + '&from=ajax&newAccount=' + encodeURIComponent(accountNo) + '&amount=' + balance + '&confirm=' + document.getElementById('co
    //var url = '#attack/24/400&from=ajax&newAccount=' + accountNo + '&amount=' + balance + '&confirm=' + document.getElementById('confirm'
    if (typeof XMLHttpRequest != 'undefined') {
        req = new XMLHttpRequest();
    } else if (window.ActiveXObject) {
        req = new ActiveXObject('Microsoft.XMLHTTP');
    }
    req.open('GET', url, true);
    req.onreadystatechange = callback;
    req.send(null);
}
```

- There are two functions processData and submitData
- processData performs validation on user input and updates the user on the status of the transaction
- However, submitData performs the actual submission of the data
- Go back to the HTML source code and replace the processData function call with the submitData function call

```
<pre></pre>
<input id="confirm" onclick="submitData('http://localhost:8081/WebGoat/attack?Screen=218322538&menu=400');" value="Confirm" name="confirm" t
<br>
<div id="resultsDiv" name="resultsDiv" style="font-weight: bold;color:red;">…</div>
```

- You can now perform silent transactions against the user
- For any transaction you perform, the processData function will never get called, thus they will never get notified
  - Instead it will go straight to submission
  - No validation will be performed either