HTTP Parameter Pollution

https://campus.barracuda.com/doc/42049405/

Description

HTTP Parameter Pollution takes advantage of ambiguous application behavior in response to HTTP requests with multiple parameters of the same name. An attacker can insert duplicate parameters to elicit an unpredictable response allowing the application to be attacked.

Effects / Weaponization

Successful attacks poison an HTTP request typically by inserting a repeated parameter with an alternate value. The attack tries to take advantage of the possibility that validation of the parameters and execution of the request may rely on different values assigned to duplicate parameters of the same name. If the request is validated using an occurrence with an allowed value of a parameter, it may still execute with the alternate illegal value found elsewhere in the request parameters. In this way, an attacker can successfully insert illegal values causing unexpected and possibly dangerous application behavior.

By sending requests with various parameter configurations to a web application, an attacker can determine how duplicate parameters are validated and what the resulting request uses in each case. With this knowledge,the attacker can potentially pass validation checks with illegal parameters that will execute in the request.

Methods (Input Vectors)

URL Request Parameters

Examples

An attacker can determine the ambiguous behavior of requests containing duplicate parameters assigned different values. By sending multiple requests to a web application with varied parameter assignments containing first one parameter=value assignment, then another parameter=value2 assignment, and finally parameter=value&parameter=value2 and recording the application responses, an attacker may uncover a vulnerability which validates using a valid parameter assignment, but then executes using the invalid (dangerous) assignment.

GET /Vulnerable_Page.jsp?parameterA=value1&parameterA=value2

In this circumstance, the GET request expects only one parameter. The resulting behavior varies depending upon the web application. Assume that value1 is valid and value2 contains an attack (e.g., buffer overflow, or injection). If value1 is validated to verify that this is a legitimate request, value2 may still be used when the request is executed. This is the danger when allowing multiple assignments to the same parameter in a single request.

Prevention

Checking the number of individual parameter instances in a request and preventing requests with duplicate parameter name=value assignments prevents this attack.

Tags

PCI-DSS

See Also

```
<a href="http://cwe.mitre.org/data/definitions/89.html" target="_blank"
class="external" id="2faafad7">CWE 235</a>,<a
href="https://www.owasp.org/index.php/SQL_Injection" target="_blank"
class="external" id="2faafad7"> OWASP</a>, <a
href="http://projects.webappsec.org/w/page/13246963/SQL%20Injection"
target="_blank" class="external" id="2faafad7">WASC</a>
```