HTTP Parameter Pollution

Summary

HTTP Parameter Pollution (HPP) is a Web attack evasion technique that allows an attacker to craft a HTTP request in order to manipulate web logics or retrieve hidden information. This evasion technique is based on splitting an attack vector between multiple instances of a parameter with the same name (?param1=value&param1=value). As there is no formal way of parsing HTTP parameters, individual web technologies have their own unique way of parsing and reading URL parameters with the same name. Some taking the first occurance, some taking the last occurance, and some reading it as an array. This behavior is abused by the attacker in order to bypass pattern-based security mechanisms.

Tools

No tools needed. Maybe Burp or OWASP ZAP.

How to test

HPP allows an attacker to bypass pattern based/black list proxies or Web Application Firewall detection mechanisms. This can be done with or without the knowledge of the web technology behind the proxy, and can be achieved through simple trial and error.

```
Example scenario.
WAF - Reads first param
Origin Service - Reads second param. In this scenario, developer trusted WAF and did
not implement sanity checks.

Attacker -- http://example.com?search=Beth&search=' OR 1=1;## --> WAF (reads first
'search' param, looks innocent. passes on) --> Origin Service (reads second 'search'
param, injection happens if no checks are done here.)
```

Table of refence for which technology reads which parameter

When ?par1=a&par1=b

Technology	**Parsing Result**	**outcome (par1=)**
ASP.NET/IIS	All occurrences	a,b
ASP/IIS	All occurrences	a,b
PHP/Apache	Last occurrence	b
PHP/Zues	Last occurrence	b
JSP,Servlet/Tomcat	First occurrence	a
Perl CGI/Apache	First occurrence	a
Python Flask	First occurrence	a
Python Django	Last occurrence	b
Nodejs	All occurrences	a,b
Golang net/http - `r.URL.Query().Get("param")`	First occurrence	a

Technology	Parsing Result	outcome (par1=)
Golang net/http - `r.URL.Query()["param"]`	All occurrences	a,b
IBM Lotus Domino	First occurrence	a
IBM HTTP Server	First occurrence	a
Perl CGI/Apache	First occurrence	a
mod_wsgi (Python)/Apache	First occurrence	a
Python/Zope	All occurences in array	['a','b']

References

- HTTP Parameter Pollution - Imperva
- HTTP Parameter Pollution in 11 minutes | Web Hacking - PwnFunction
- How to Detect HTTP Parameter Pollution Attacks - Acunetix