# Dangerous HTTP methods

| | |
|---|---|
| Tags | checklist web |
| Property | |

Introduction
POST and GET
PUT
DELETE
CONNECT
Finding these request methods
Conclusion

## Introduction

While most HTTP methods are safe or are implemented in a safe manner, unfortunately, we can not say the same for every server out there. They might implement request methods incorrectly which may even lead to critical vulnerabilities such as remote code execution or firewall bypass.

Usually, the POST and GET methods are safe but they might be implemented incorrectly which leads to logic vulnerabilities. The same can not be said for the PUT, DELETE and CONNECT methods, however. Let’s explore how exactly these methods can lead to an issue.

## POST and GET

The POST and GET methods are used often and are safe in general unless implemented incorrectly. The POST method can make changes to the data but not to the server which is why it is considered safe in general. The GET method can request data from the server.

# PUT

The PUT method can be very dangerous as it allows attackers to upload files to the server which might contain malicious code. This can go wrong in various ways. Of course, there is the obvious way where an attacker will try to upload a reverse shell and then trigger it somehow but there’s also a lesser-known exploit that plagues the PUT method. An attacker can use this method for phishing as they can upload HTML pages with links to malicious pages or malicious login forms.

An example of an attack can be when PUT is enabled in a JBOSS server as the attacker can upload a JSP reverse shell and actually get remote code execution on the server. Please note this can much more than just a reverse shell.

# DELETE

The DELETE method can easily be abused to delete important files from the server if it is misconfigured. This includes files that stop the attacker from accessing certain resources such as admin pages such as the .htaccess file. This would allow the attacker unauthorized access to all certain resources.

# CONNECT

This HTTP method can be used to create a P2P connection over HTTP. this has many implications and can lead to bypassing a firewall for example.

# Finding these request methods

We have several options available to use to grab the HTTP methods.

- telnet ip 80(or the HTTP(s) port if it’s different). This can return an allow header that contains all the allowed HTTP methods
- with NMAP with can use either the -sC or --script=htttp-methods.nse.
- We can send a request with the method OPTIONS.

# Conclusion

It’s best to disable the HTTP methods CONNECT, PUT and DELETE and maybe even OPTIONS which disables the attacker from viewing the HTTP methods but that should not be your only prevention method as security through obscurity does not work.