Exploiting Put Method

Asfiya $ha!kh Apr 27, 2019 · 4 min read

Seems like a boring topic? But mind you, its NOT!

An attacker could get a local or root shell on the system using publicly accessible put method also known as one of Webdav method.

WebDAV is a term given to a collection of HTTP methods.HTTP requests can use a range of methods other than the standard GET and POST methods.

WebDAV can be used to manipulate files on the web server. Given the nature of the functionality, if these are accessible by low-privileged users, they may provide an effective avenue for attacking an application.

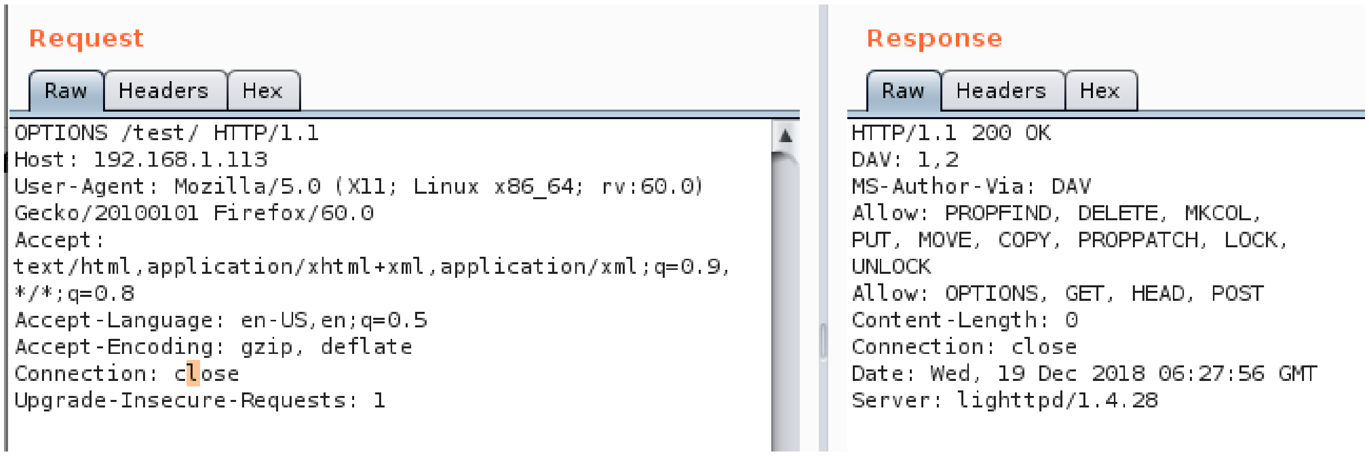
Aahh… the bookish things … lets explore more

Here are some methods to look for:
- PUT uploads the attached file to the specified location.
- DELETE deletes the specified resource.
- COPY copies the specified resource to the location given in the Destination header.
- MOVE moves the specified resource to the location given in the Destination header.
- SEARCH searches a directory path for resources.

- PROPFIND retrieves information about the specified resource, such as author, size, and content type.

Okay … explained it on a big note … Lets check how we can actually perform the attack.

We can use the OPTIONS method to list the HTTP methods that are permitted in a particular directory.



```
OPTIONS /test/ HTTP/1.1
Host: 192.168.1.113
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0)
Gecko/20100101 Firefox/60.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,
*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200 OK
DAV: 1,2
MS-Author-Via: DAV
Allow: PROPFIND, DELETE, MKCOL,
PUT, MOVE, COPY, PROPPATCH, LOCK,
UNLOCK
Allow: OPTIONS, GET, HEAD, POST
Content-Length: 0
Connection: close
Date: Wed, 19 Dec 2018 06:27:56 GMT
Server: lighttpd/1.4.28
```

This response indicates that several of the powerful methods listed previously are in fact allowed.However we will be looking at exploiting put method.

The PUT method is particularly dangerous. If you upload arbitrary files within the web root, the first target is to create a backdoor script on the server that will be executed by a server-side module, thereby giving the attacker full control of the application, and often the web server itself.

If the PUT method appears to be present and enabled,application will respond with 201 created response as shown in below Exhibit.

you may receive 405 status code if you attempt to use the PUT method where it is not supported. 405 Method Not Allowed indicates that the method used in the request is not supported for the specified URL.

We can even upload reverse shell directly instead of backdoor as shown in below Exhibit.

Note that permissions are likely to be implemented per directory, so recursive checking is required in an attack. Tools such as DAVTest can be used to iteratively check all directories on the server for the PUT method.

Observe the difference between below tests.

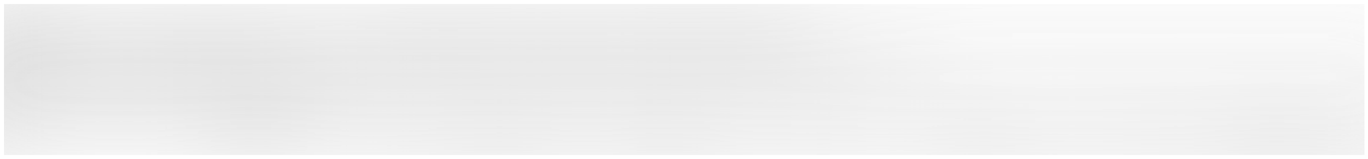


Image courtesy — https://tools.kali.org/web-applications/davtest

As http://192.168.1.209/ has put method enabled, it shows different response than http://mdsec.net/public which has disabled put method access.

Also note that For WebDAV instances where end users are permitted to upload files, it is relatively common for uploading server-side scripting language extensions specific to that server's environment to be forbidden. The ability to upload HTML or JAR files is much more likely, and both of these allow attacks against other users to be conducted.

Huhhh…. too much of information … lets summarize the testing steps to be followed -
a. Attempt to use the PUT method to upload a benign file
b. If this is successful, try uploading a backdoor script using PUT.
c. If the necessary extension for the backdoor to operate is being
blocked, try uploading the file with a .txt extension and using the
MOVE method to move it to a file with a new extension.(If MOVE doesn't work try COPY method)
d. If any of the preceding methods fails, try uploading a JAR file, or a file
with contents that a browser will render as HTML.
e. Recursively step through all the directories using a tool such as
davtest.

There are some webdav testing tools such as cadaver for kali linux and phTagr for windows, which can help along.

Remediation

Only GET and POST HTTP methods should be allowed and other unused methods should be blocked. If required these extra methods should only be enabled with credentials and public access denied.

References

http://stackoverflow.com/questions/320959/disabling-put-trace-delete-request-in-apache-tomcat-6-0
https://forums.asp.net/t/1703115.aspx?Disable+http+OPTIONS+method
http://stackoverflow.com/questions/320959/disabling-put-trace-delete-request-in-apache-tomcat-6-0

If you are still reading this, Then let me tell you , Its the end. See you next time..!!

Hacking Security Pentesting Penetration Testing Webdev