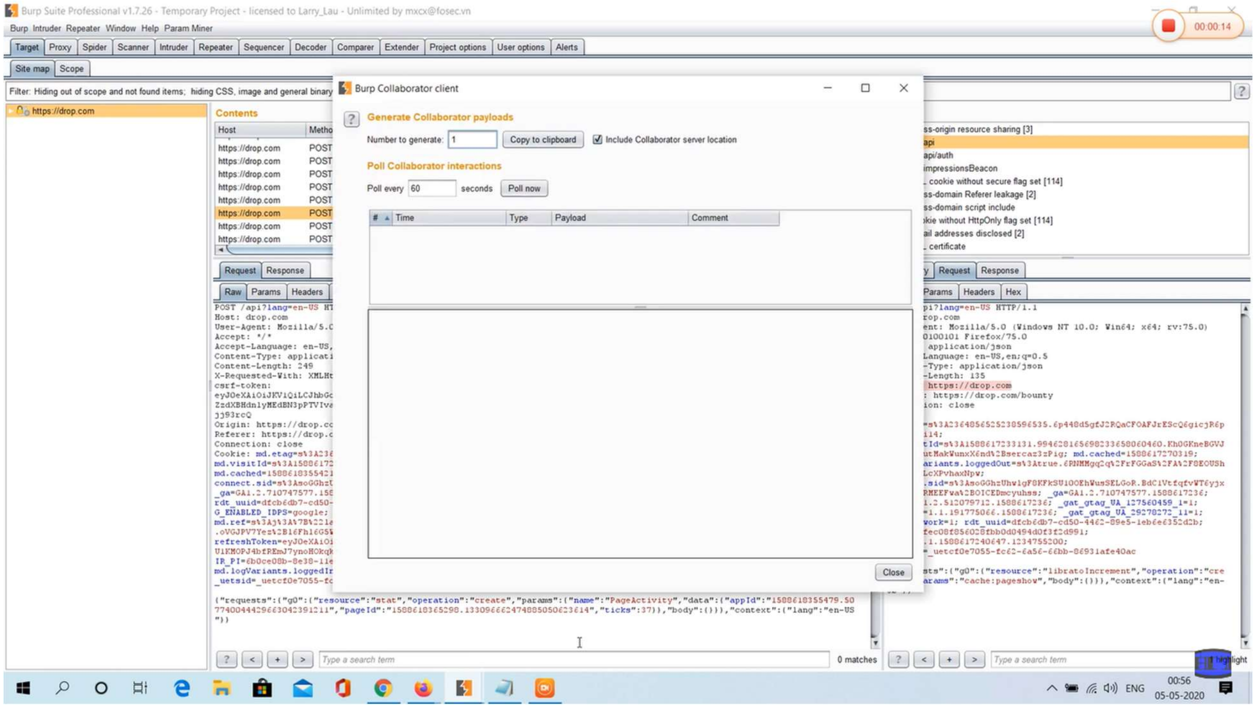
Day 12: Identifying and Exploiting SSRF Vulnerabilities

The objective was to find a Server-Side Request Forgery (SSRF) vulnerability within a web application and demonstrate how it can be exploited using the Burp Suite tool. This report outlines the steps taken and the findings of this practical exercise.


Burp Suite Professional v1.7.26 - Temporary Project - licensed to Larry_Lau - Unlimited by mxcx@fosec.vn
Burp Intruder Repeater Window Help Param Miner
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Site map Scope
00:00:14
Burp Collaborator client
Generate Collaborator payloads
Copy to clipboard
Poll Collaborator interactions
Poll now
# Time Type Payload Comment
Close
Contents
Host Metho
Request Response
Raw Params Headers Hex
Type a search term
0 matches
1 highlight
ENG
00:56
05-05-2020

Filter: Hiding out of scope and not found items; hiding CSS, image and general binary

Number to generate: 1

☑ Include Collaborator server location

Poll every 60 seconds

https://drop.com

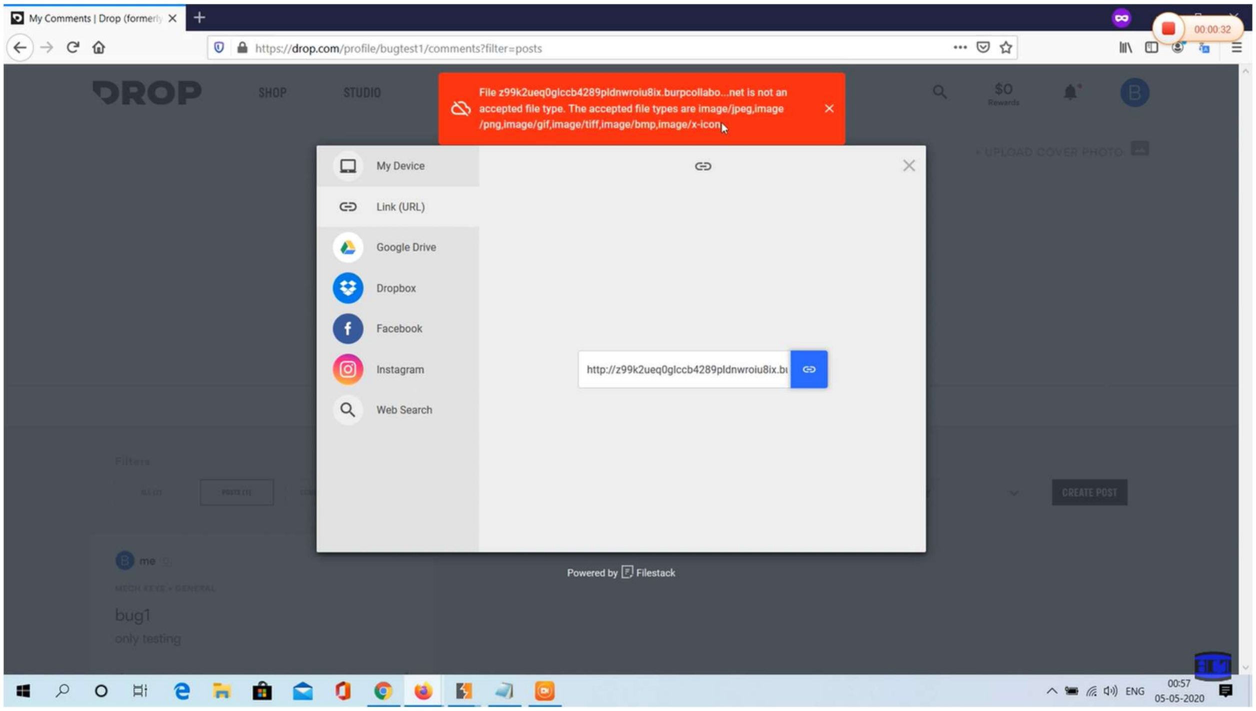
Host	Metho
https://drop.com	POST
https://drop.com	POST
https://drop.com	POST
https://drop.com	POST
https://drop.com	POST
https://drop.com	POST
https://drop.com	POST
https://drop.com	POST

ss-origin resource sharing [3]
api
api/auth
impressionsBeacon
. cookie without secure flag set [114]
ss-domain Referer leakage [2]
ss-domain script include
okie without HttpOnly flag set [114]
ail addresses disclosed [2]
. certificate

```
POST /api?lang=en-US HT
Host: drop.com
User-Agent: Mozilla/5.C
Accept: */*
Accept-Language: en-US,
Content-Type: applicati
Content-Length: 249
X-Requested-With: XMLHt
csrf-token:
eyJ0eXAiOiJKV1QiLCJhbGc
ZzdXBHdnlyMEdBN3pPTVIva
jj93rcQ
Origin: https://drop.cc
Referer: https://drop.c
Connection: close
Cookie: md.etag=s%3A23€
md.visitId=s%3A15886172
md.cached=1588618355421
connect.sid=s%3AsoGGhzU
_ga=GA1.2.710747577.158
rdt_uuid=dfcb6db7-cd50-
G_ENABLED_IDPS=google;
md.ref=s%3Aj%3A%7B%22le
.oVGJPV7Yez%2B16Fh16G5%
refreshToken=eyJ0eXAiOi
U1KMOPJ4bfREmJ7ynoHOkqk
IR_PI=6b0ce08b-8e38-11e
md.logVariants.loggedIr
_uetsid=_uetcf0e7055-fc

{"requests":{"g0":{"resource":"stat","operation":"create","params":{"name":"PageActivity","data":{"appId":"1588618355479.50
7740044429663042391211","pageId":"1588618365298.1330966624748850506236l4","ticks":37}},"body":{}}},"context":{"lang":"en-US
"}}
```

```
pi?lang=en-US HTTP/1.1
rop.com
ent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:75.0)
0100101 Firefox/75.0
 application/json
Language: en-US,en;q=0.5
-Type: application/json
-Length: 135
 https://drop.com
: https://drop.com/bounty
ion: close

=s%3A23648565252385965 35.6p448d5gfJ2RQaCFOAFJrEScQ6gicjR6p
i14;
tId=s%3A1588617233131.99462816569823365806046 0.KhOGKneBGVJ
utMakWunxX6nd%2Bsercaz3zPig; md.cached=1588617270319;
ariants.loggedOut=s%3Atrue.6RNMMgq2q%2FrFGGaS%2FA%2F8EOUSh
LcXPvhaxNpw;
.sid=s%3AsoGGhzUhwlgF8KFkSU1OOEhWusSELGoR.BdC1VtfqfvWT6yjx
RMEEFwa%2BOICEDmcyuhss; _ga=GA1.2.710747577.1588617236;
1.2.512079712.1588617236; _gat_gtag_UA_127560459_1=1;
=1.1.191775066.1588617236; _gat_gtag_UA_29278272_11=1;
work=1; rdt_uuid=dfcb6db7-cd50-4462-89e5-1eb6e6352d2b;
fec08f856028fbb0d0494d0f3f2d991;
.1.1588617240647.1234755200;
=_uetcf0e7055-fc62-6a56-66bb-86931afe40ac

sts":{"g0":{"resource":"libratoIncrement","operation":"cre
arams":"cache:pageshow","body":{}}},"context":{"lang":"en-
```

I selected a web application for testing, focusing on areas where user-supplied input could influence URL requests.

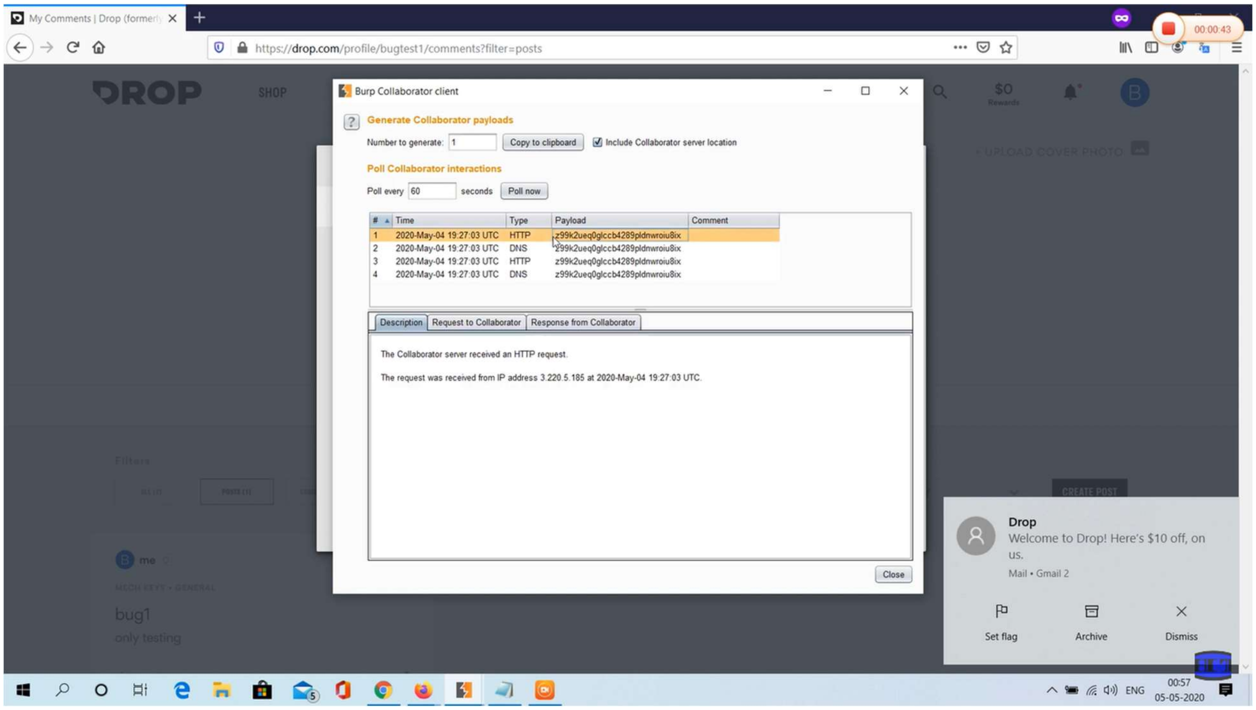


https://drop.com/profile/bugtest1/comments?filter=posts

File z99k2ueq0glccb4289pldnwroiu8ix.burpcollabo...net is not an accepted file type. The accepted file types are image/jpeg,image/png,image/gif,image/tiff,image/bmp,image/x-icon

http://z99k2ueq0glccb4289pldnwroiu8ix.bu

Using Burp Suite, I conducted initial testing of the application to identify any potential entry points for SSRF.



Burp Collaborator client

Generate Collaborator payloads

Number to generate: 1 Copy to clipboard ☑ Include Collaborator server location

Poll Collaborator interactions

Poll every 60 seconds Poll now

#	Time	Type	Payload	Comment
1	2020-May-04 19:27:03 UTC	HTTP	z99k2ueq0glccb4289pldnwroiu8ix	
2	2020-May-04 19:27:03 UTC	DNS	z99k2ueq0glccb4289pldnwroiu8ix	
3	2020-May-04 19:27:03 UTC	HTTP	z99k2ueq0glccb4289pldnwroiu8ix	
4	2020-May-04 19:27:03 UTC	DNS	z99k2ueq0glccb4289pldnwroiu8ix	

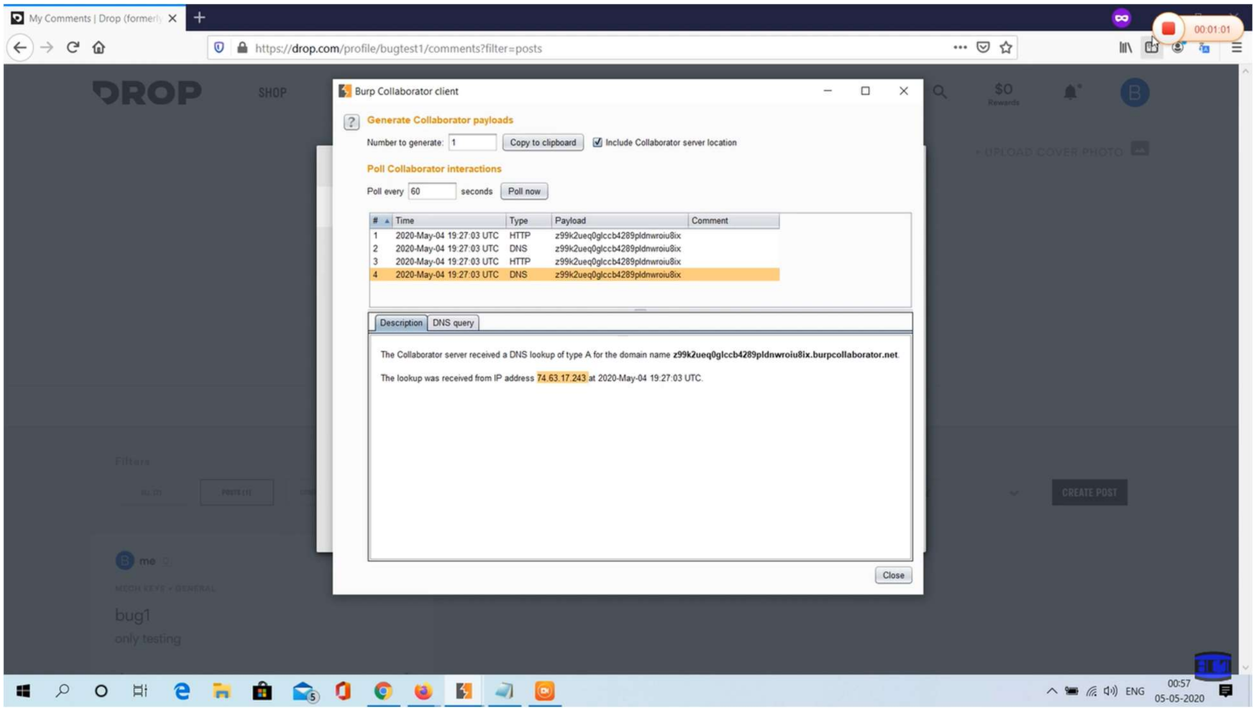
Description | Request to Collaborator | Response from Collaborator

The Collaborator server received an HTTP request.

The request was received from IP address 3.220.5.185 at 2020-May-04 19:27:03 UTC.

Drop
Welcome to Drop! Here's $10 off, on us.
Mail • Gmail 2

I crafted payloads that included URLs pointing to internal and external resources, keeping an eye out for any unusual behaviors or responses from the application.



Generate Collaborator payloads

Number to generate: 1 ☑ Include Collaborator server location

Poll Collaborator interactions

Poll every 60 seconds

#	Time	Type	Payload	Comment
1	2020-May-04 19:27:03 UTC	HTTP	z99k2ueq0glccb4289pldnwroiu8ix	
2	2020-May-04 19:27:03 UTC	DNS	z99k2ueq0glccb4289pldnwroiu8ix	
3	2020-May-04 19:27:03 UTC	HTTP	z99k2ueq0glccb4289pldnwroiu8ix	
4	2020-May-04 19:27:03 UTC	DNS	z99k2ueq0glccb4289pldnwroiu8ix	

The Collaborator server received a DNS lookup of type A for the domain name **z99k2ueq0glccb4289pldnwroiu8ix.burpcollaborator.net**.

The lookup was received from IP address 74.63.17.243 at 2020-May-04 19:27:03 UTC.

I injected crafted payloads into input fields or parameters, where I suspected SSRF vulnerabilities might exist.

I carefully monitored the application's responses, looking for any signs of successful SSRF exploitation, such as responses containing data from internal resources or external service requests initiated by the server.

Conclusion:

A critical security concern that can have significant consequences if not properly addressed. By using Burp Suite as a tool, I was able to locate and demonstrate how the vulnerability can be exploited.

It's essential to emphasize that SSRF vulnerabilities can pose substantial risks to web applications and their users. Responsible disclosure of such vulnerabilities to application owners is crucial to ensure prompt remediation and to protect the application's security.

Note: I'm not felling well so I craft this report in hurry and I don't know will u like it or not, sorry for this and I also missed day 11 task.