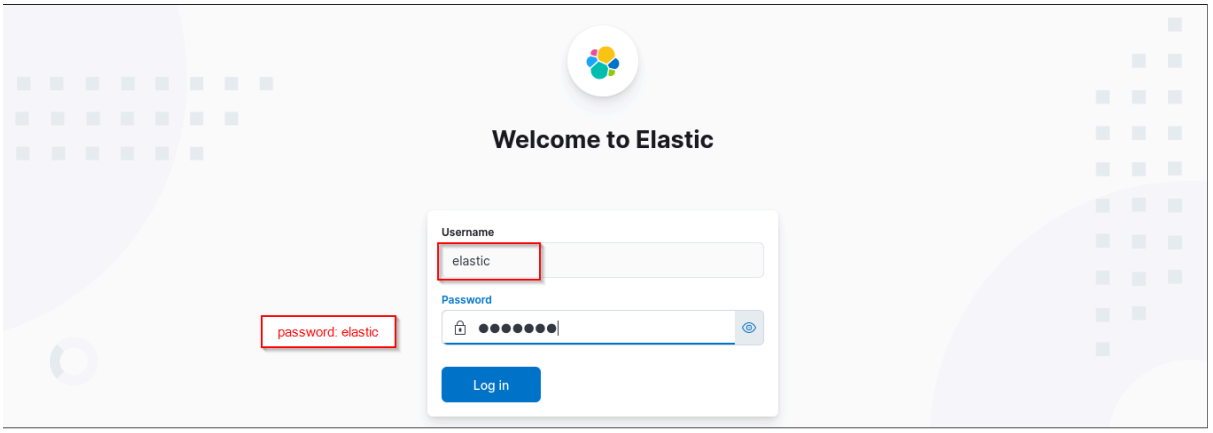
Advent of Cyber 2024

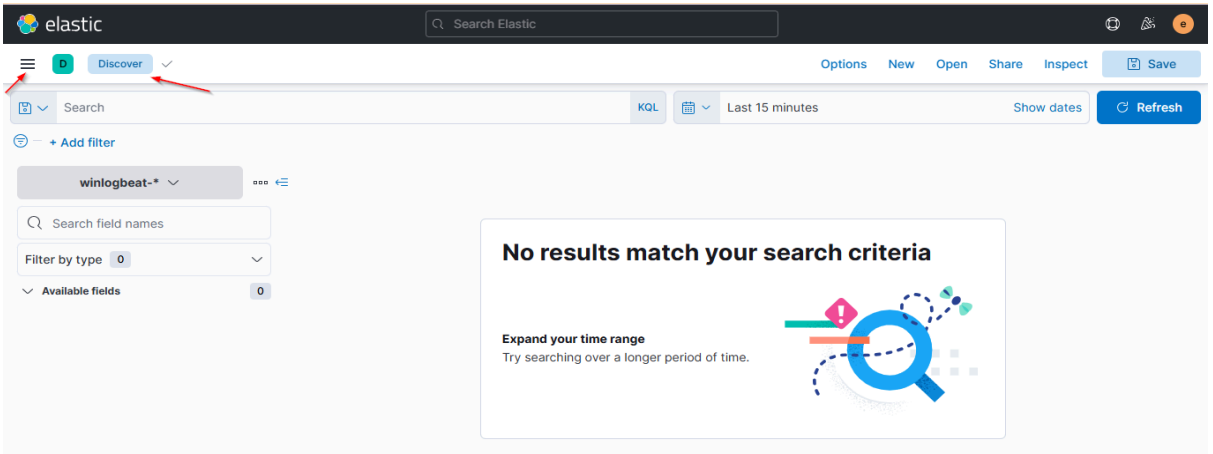
Dive into the wonderful world of cyber security by engaging in festive beginner-friendly exercises every day in the lead-up to Christmas!

Day 2: One man's false positive is another man's potpourri.

Step 1: Accessing the ELK Interface

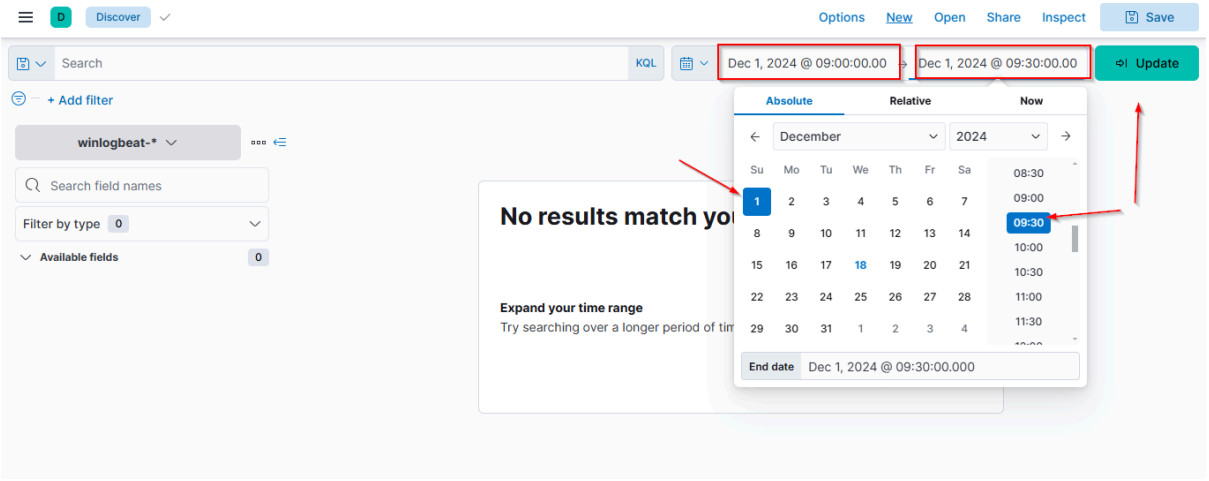
- ➔ I started by loading the URL https://10-10-254-35.p.thmlabs.com/ in my browser.
- ➔ Once logged in, I clicked on the menu in the top-left corner and navigated to the Discover tab.
- ➔ This took me to the event logs where I could start investigating.

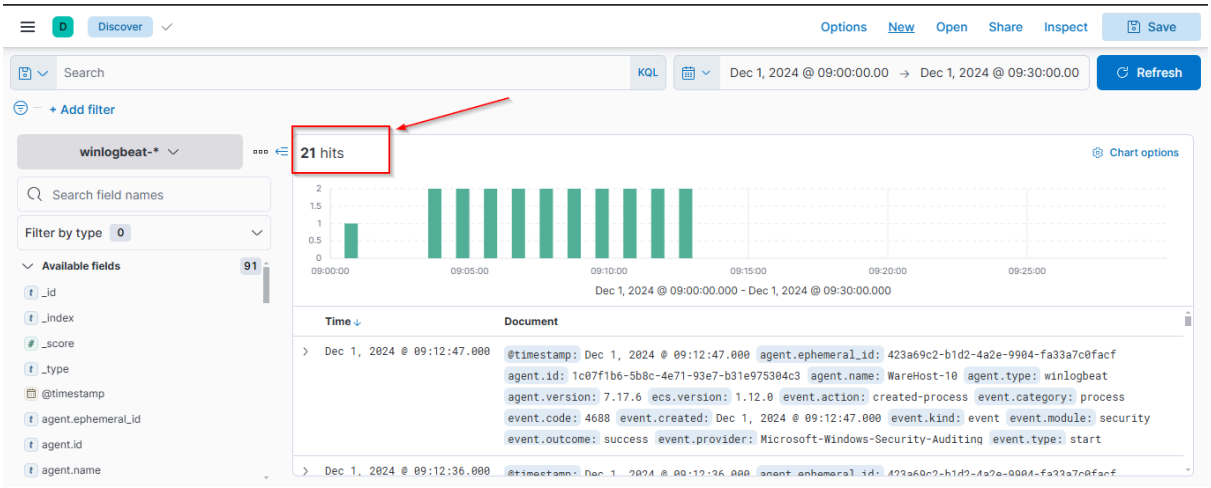




Step 2: Setting the Time Window

- ➔ According to the alert, the suspicious activity occurred between 09:00 and 09:30 on December 1st.
- ➔ So, I set the time frame by clicking the timeframe settings in the top-right corner, selecting the Absolute tab, and entering the start and end times. I clicked Update to apply the changes.
- ➔ After doing this, I saw 21 events in the specified time window.



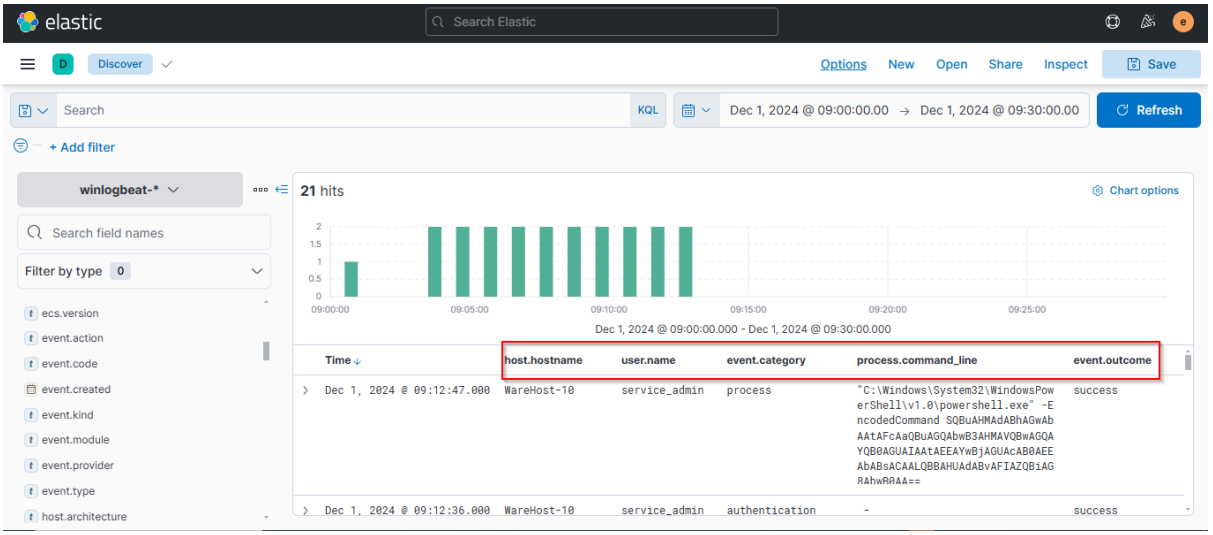


Step 3: Making the Events Readable

- ➔ At first glance, the events weren't very readable, so I decided to add a few columns to make it easier to analyze. I hovered over the field names in the left pane and added the following:

- host.hostname: To see which machine the command was run on.
- user.name: To identify the user who performed the activity.
- event.category: To confirm we're looking at the right event category.
- process.command_line: To check what PowerShell command was executed.
- event.outcome: To understand if the activity was successful.

➔ Once I added these fields, the results were much more digestible.



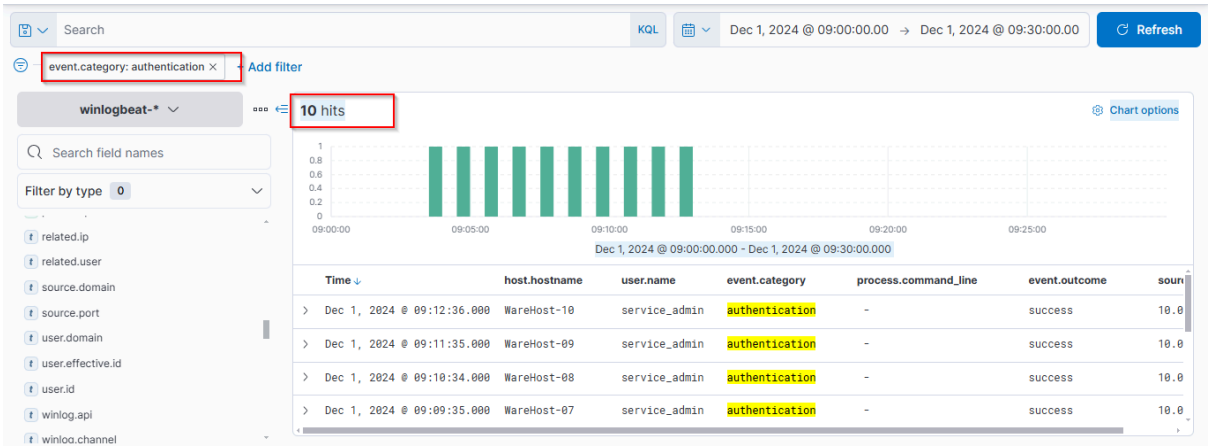
Step 4: Investigating the Events

➔ After reviewing the data, I found something interesting, someone had run the same encoded PowerShell command on multiple machines.
➔ I noticed that before every execution of the PowerShell command, there was a successful authentication event.
➔ This suggests that the attacker likely gained access before running the commands.
➔ To dig deeper, I decided to add the source.ip field as a column. This would show the IP address that triggered these commands. Since source.ip only appeared in authentication events, I filtered out the process events.

Step 5: Filtering for Authentication Events

➔ To narrow down the results, I clicked the plus (+) icon next to event.category and filtered for authentication events.

➔ This helped me see that the same IP address (10.0.11.11) kept appearing in the logs over the past few days, with a noticeable spike in activity at the end of the logs.

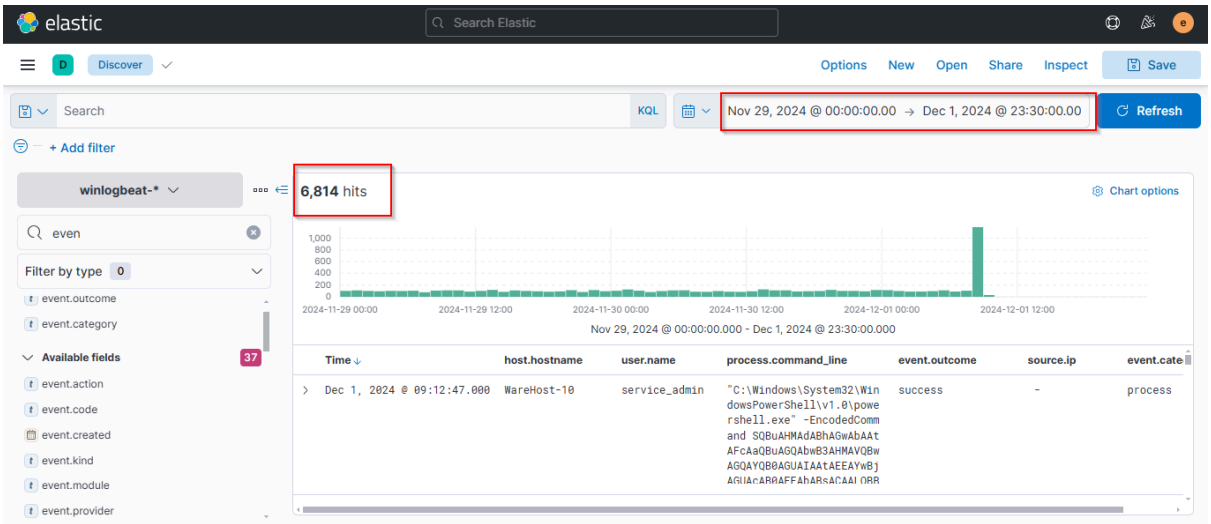


Step 6: Expanding the Time Window

➔ To get more context, I expanded the timeframe from November 29th to December 1st.

➔ After applying the new time filter, I saw over 6800 events over the course of three days, with the spike at the end clearly standing out.

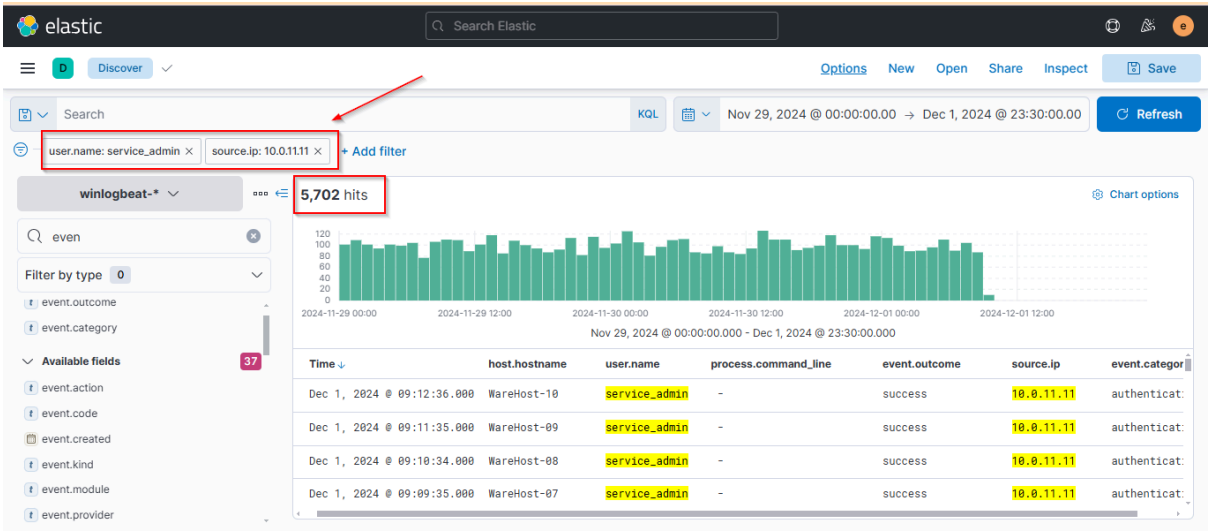
Note: Remember to remove the event.category filter before this step.

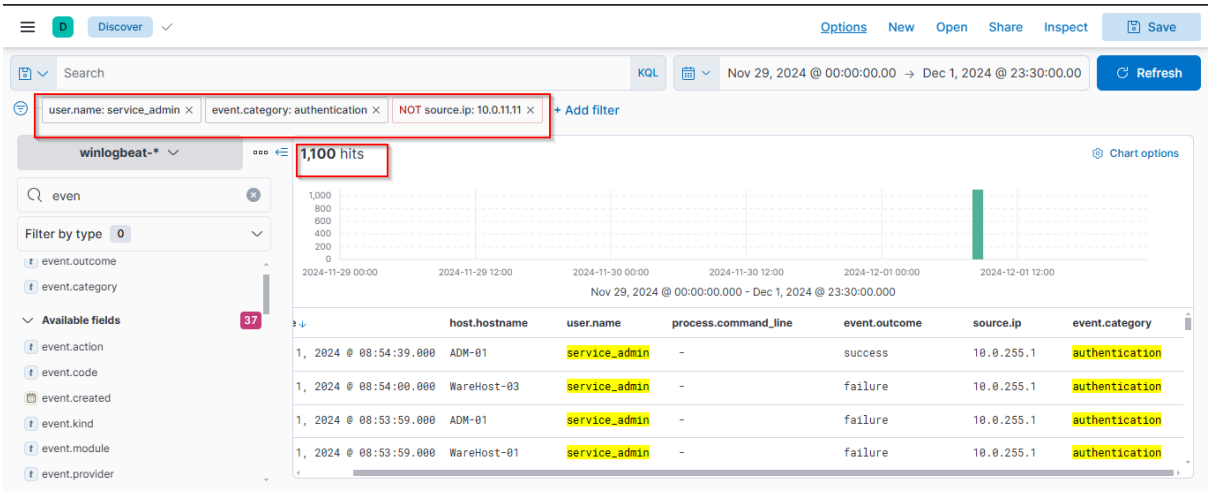


Step 7: Filtering by User and IP

➔ I then decided to focus on the service_admin user and IP address 10.0.11.11, so I added those filters to narrow my search.

➔ What I found was a mix of failed login attempts, but then, right after the spike, new authentication events came from a different IP address: 10.0.255.1.

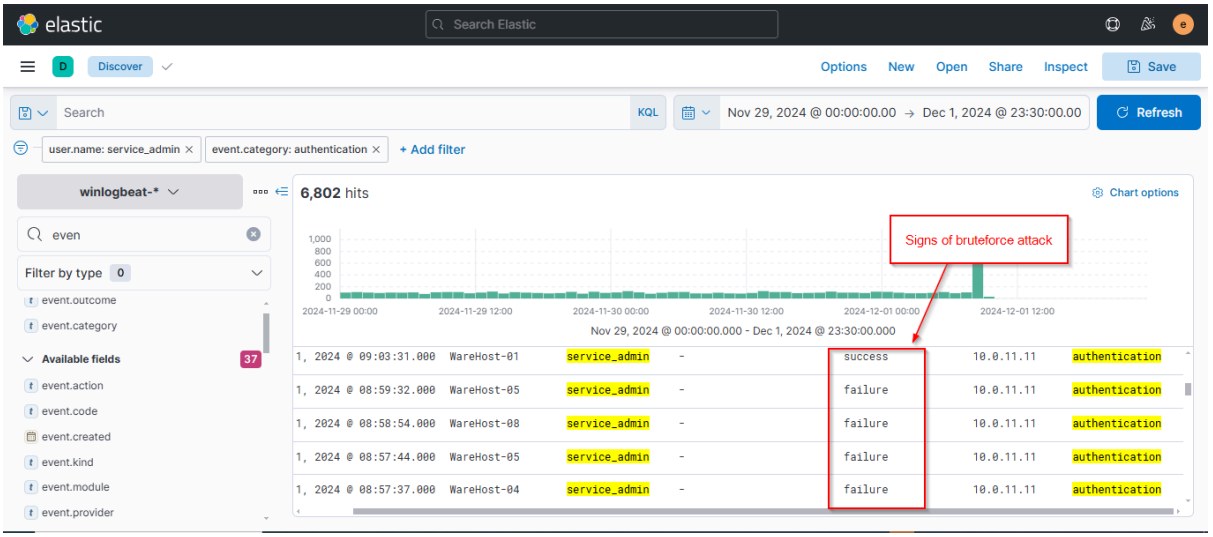


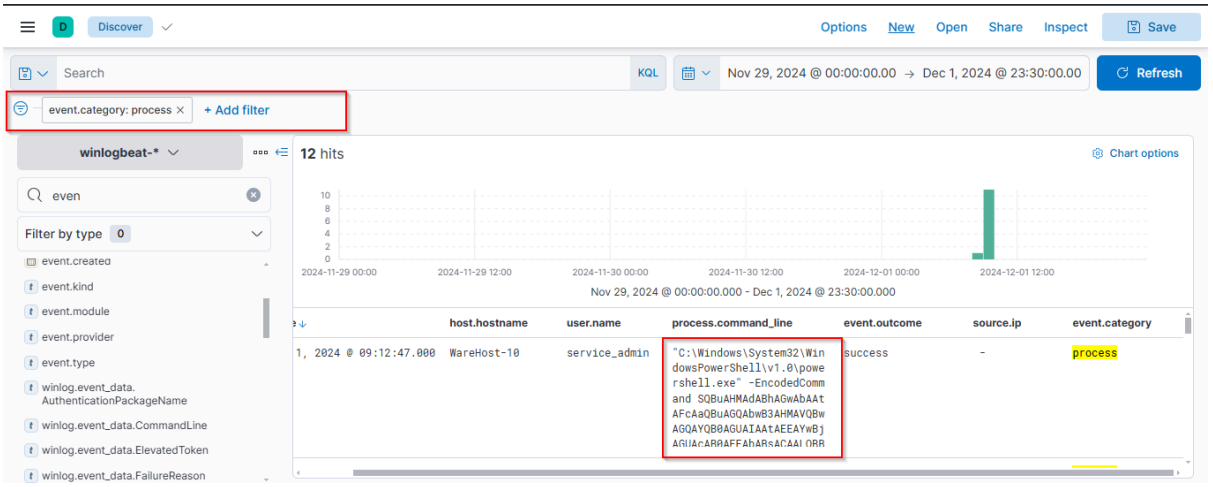


Step 8: Investigating the Brute-Force Attack

➔ I focused on the new IP address (10.0.255.1) and saw a series of failed login attempts, followed by a successful login and PowerShell command execution.

➔ This pattern suggested a brute-force attack, where the attacker tried multiple login attempts until successful, then executed the malicious PowerShell commands.

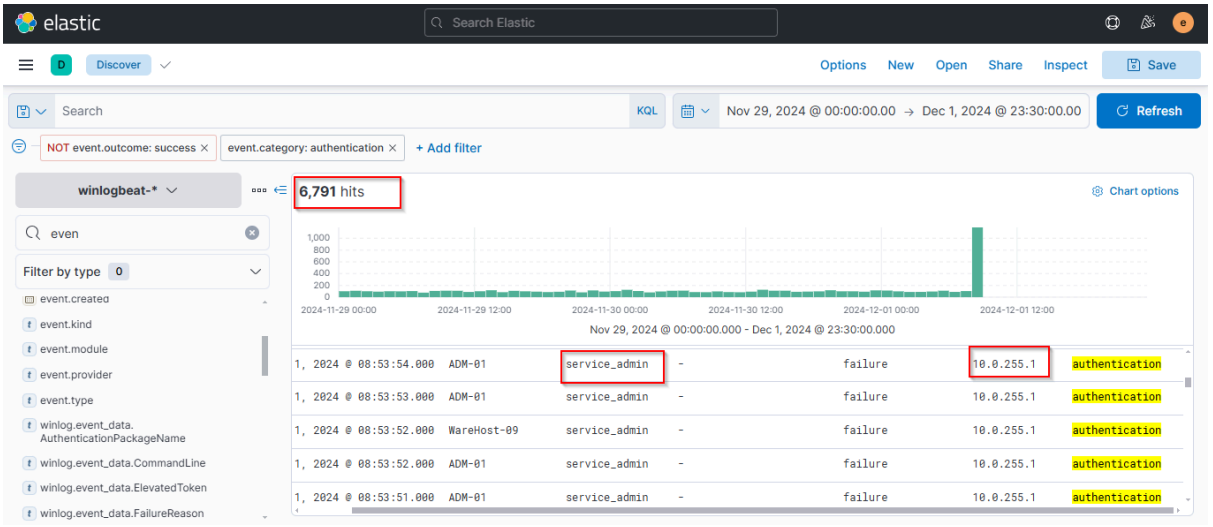




Question: What is the name of the account causing all the failed login attempts?
Answer: service_admin

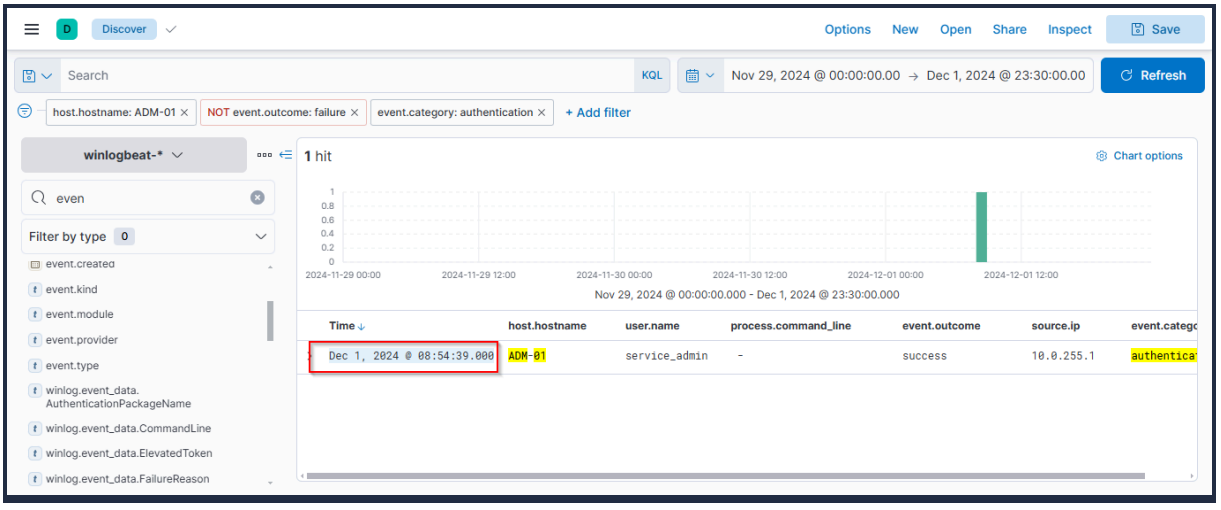
Question: How many failed logon attempts were observed?
Answer: 6791

Question: What is the IP address of Glitch?
Answer: 10.0.255.1



Question: When did Glitch successfully logon to ADM-01? Format: MMM D, YYYY HH:MM:SS.SSS

Answer: Dec 1, 2024 08:54:39.000



Step 9: Analyzing the PowerShell Command

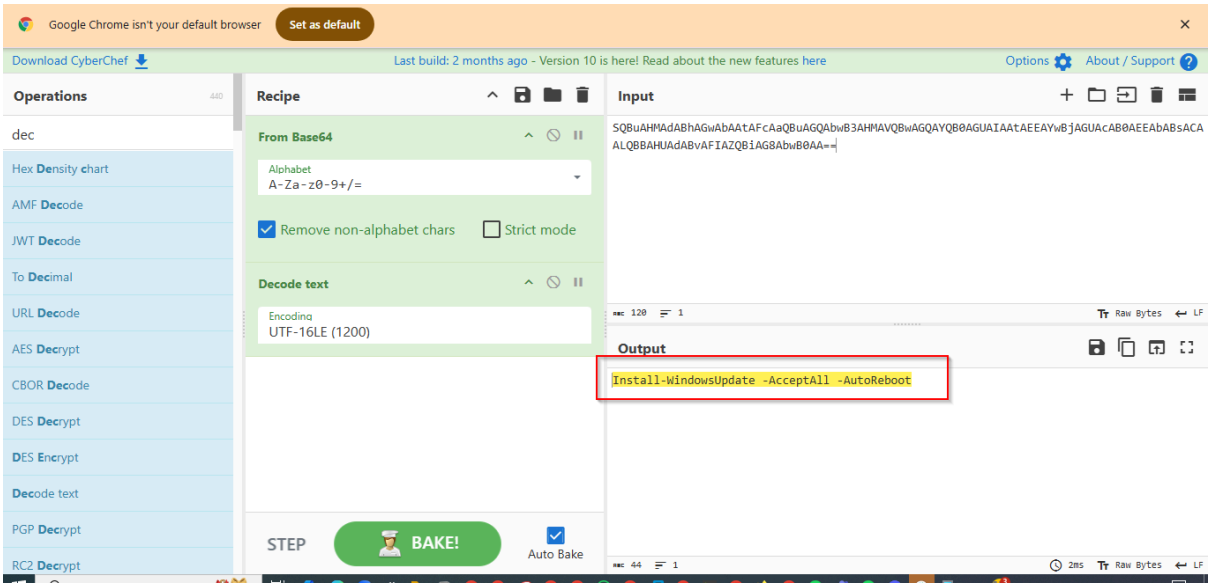
- ➔ At this point, I knew that the attacker had executed encoded PowerShell commands, but since they were Base64 encoded, I needed to decode them to understand their purpose.
- ➔ To decode the command, I used CyberChef, a tool that can help with various decoding techniques.

Step 10: McSkidy's Analysis

- ➔ I passed the investigation over to McSkidy, who took action by spinning up her own CyberChef instance to decode the PowerShell command locally.
- ➔ I used the FromBase64 recipe, and decoded the command.
- ➔ Then set the encoding to UTF-16LE, which is the encoding used by PowerShell for Base64.

Question: What is the decoded command executed by Glitch to fix the systems of Wareville?

Answer: Install-WindowsUpdate -AcceptAll -AutoReboot



END!!!