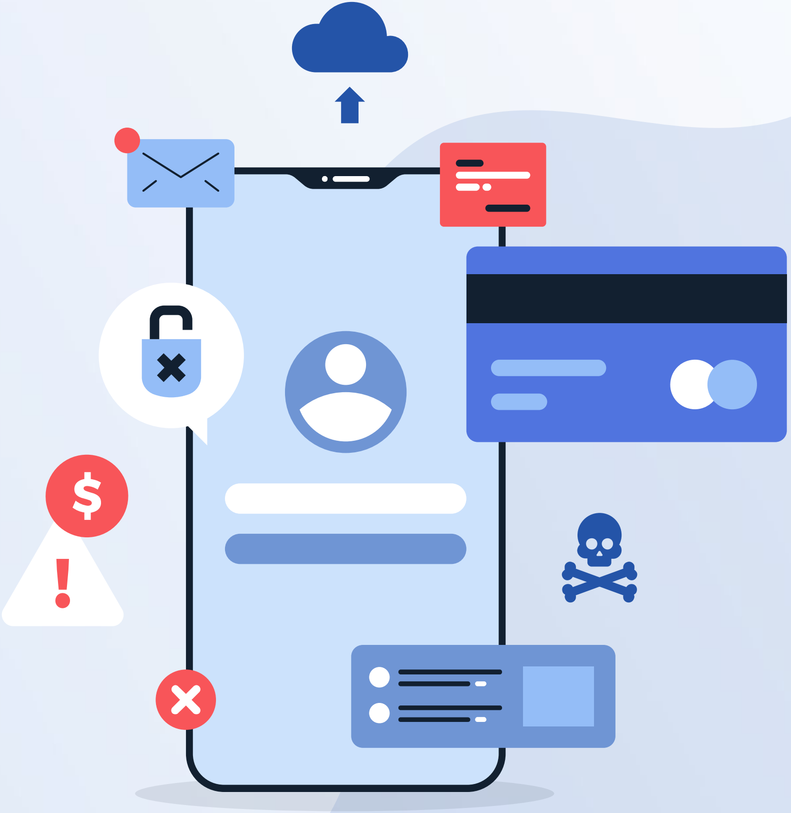
Lab 14

Linux Logs Analysis

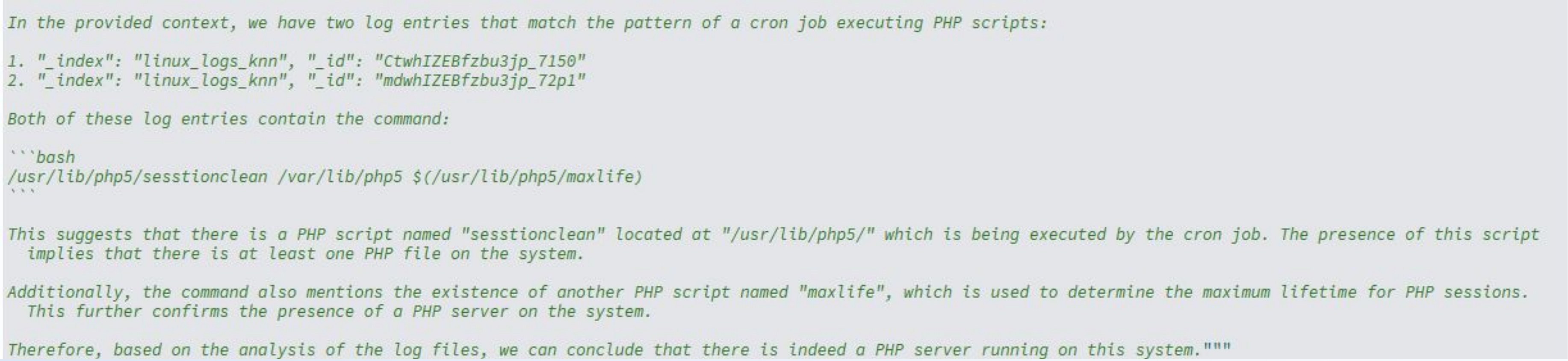


Targeted questioning

You may have noticed that the LLM does much better when you ask more pointed questions. Below is an example of a pointed question, as well as a much better output.

```
POST
/ plugins/_ml/agents/DHaHWZgBKJWqR4KGzkE_/_execu
te
{
  "parameters": {
    "question": "What countries are present in
the dataset."
  }
}
```

We can explicitly see the call of context from our embedding model with the response as well.



~~~
In the provided context, we have two log entries that match the pattern of a cron job executing PHP scripts:

1. "_index": "linux_logs_knn", "_id": "CtwhIZEBfzbu3jp_7150"
2. "_index": "linux_logs_knn", "_id": "mdwhIZEBfzbu3jp_72p1"

Both of these log entries contain the command:

```bash
/usr/lib/php5/sesstionclean /var/lib/php5 $(/usr/lib/php5/maxlife)
```

This suggests that there is a PHP script named "sesstionclean" located at "/usr/lib/php5/" which is being executed by the cron job. The presence of this script
  implies that there is at least one PHP file on the system.

Additionally, the command also mentions the existence of another PHP script named "maxlife", which is used to determine the maximum lifetime for PHP sessions.
  This further confirms the presence of a PHP server on the system.

Therefore, based on the analysis of the log files, we can conclude that there is indeed a PHP server running on this system."""
~~~

Lab Objectives

- Manually parse through Linux Logs via Opensearch
- Query the LLM to parse through the Linux_Logs Index
- Modify the Agent Pipeline to alter model output

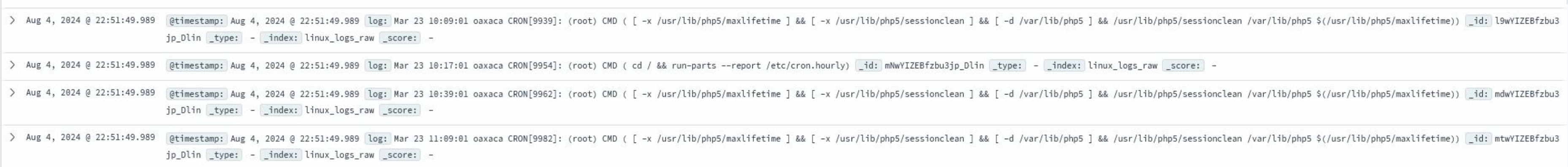
WRCCDC

The Western Regional Collegiate Cyber Defense Competition pits a group of eight students against a team of seasoned, industry veteran Red Teamers who forgot to sign the full Scoping document. As such, there are almost no holds barred on what they can do in the environment.

On the flip-side, competitor teams are allowed to reset their boxes for the small price of points and a donation to the Taco fund for Ops team. The syslog and auth log in this index is courtesy of some of the teams competing. As such- they are.. messy

Dashboards

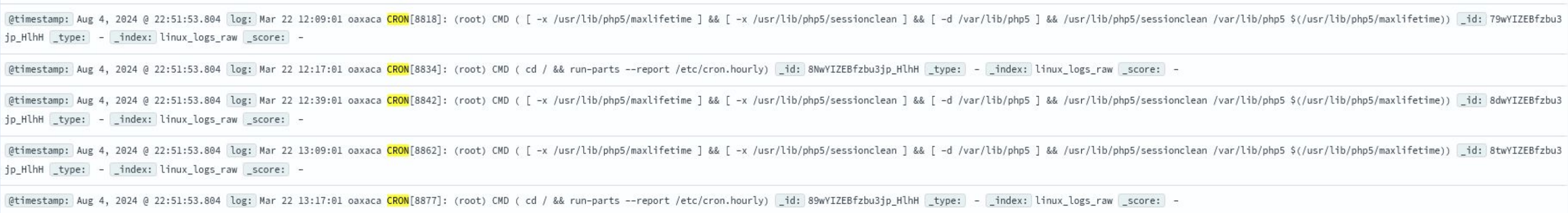
First, let's take a look at the raw syslog in Discover. For the most part, due to box resets, these are very boring. A large portion of these logs are in-fact, from the system turning on after a reboot, as well as normal system operations such as cronjobs.



```
> Aug 4, 2024 @ 22:51:49.989  @timestamp: Aug 4, 2024 @ 22:51:49.989 log: Mar 23 10:09:01 oaxaca CRON[9939]: (root) CMD ( [ -x /usr/lib/php5/maxlifetime ] && [ -x /usr/lib/php5/sessionclean ] && [ -d /var/lib/php5 ] && /usr/lib/php5/sessionclean /var/lib/php5 $(/usr/lib/php5/maxlifetime)) _id: l9wYIZEBfzbu3
                              jp_Dlin _type: - _index: linux_logs_raw _score: -
> Aug 4, 2024 @ 22:51:49.989  @timestamp: Aug 4, 2024 @ 22:51:49.989 log: Mar 23 10:17:01 oaxaca CRON[9954]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly) _id: mNwYIZEBfzbu3jp_Dlin _type: - _index: linux_logs_raw _score: -
> Aug 4, 2024 @ 22:51:49.989  @timestamp: Aug 4, 2024 @ 22:51:49.989 log: Mar 23 10:39:01 oaxaca CRON[9962]: (root) CMD ( [ -x /usr/lib/php5/maxlifetime ] && [ -x /usr/lib/php5/sessionclean ] && [ -d /var/lib/php5 ] && /usr/lib/php5/sessionclean /var/lib/php5 $(/usr/lib/php5/maxlifetime)) _id: mdwYIZEBfzbu3
                              jp_Dlin _type: - _index: linux_logs_raw _score: -
> Aug 4, 2024 @ 22:51:49.989  @timestamp: Aug 4, 2024 @ 22:51:49.989 log: Mar 23 11:09:01 oaxaca CRON[9982]: (root) CMD ( [ -x /usr/lib/php5/maxlifetime ] && [ -x /usr/lib/php5/sessionclean ] && [ -d /var/lib/php5 ] && /usr/lib/php5/sessionclean /var/lib/php5 $(/usr/lib/php5/maxlifetime)) _id: mtwYIZEBfzbu3
                              jp_Dlin _type: - _index: linux_logs_raw _score: -
```

Running Services

Let’s take a look at the crontab with the query `log:cron.` At first, we will be inundated by entries for a php server. Utilizing what you know of DQL(or Lucene) so far, try to find other services or tools running on these systems. There are at least two other services or tools on these systems.



```
@timestamp: Aug 4, 2024 @ 22:51:53.804 log: Mar 22 12:09:01 oaxaca CRON[8818]: (root) CMD ( [ -x /usr/lib/php5/maxlifetime ] && [ -x /usr/lib/php5/sessionclean ] && [ -d /var/lib/php5 ] && /usr/lib/php5/sessionclean /var/lib/php5 $(/usr/lib/php5/maxlifetime)) _id: 79wYIZEBfzbu3
jp_HlhH _type:  - _index: linux_logs_raw _score:  -

@timestamp: Aug 4, 2024 @ 22:51:53.804 log: Mar 22 12:17:01 oaxaca CRON[8834]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly) _id: 8NwYIZEBfzbu3jp_HlhH _type:  - _index: linux_logs_raw _score:  -

@timestamp: Aug 4, 2024 @ 22:51:53.804 log: Mar 22 12:39:01 oaxaca CRON[8842]: (root) CMD ( [ -x /usr/lib/php5/maxlifetime ] && [ -x /usr/lib/php5/sessionclean ] && [ -d /var/lib/php5 ] && /usr/lib/php5/sessionclean /var/lib/php5 $(/usr/lib/php5/maxlifetime)) _id: 8dwYIZEBfzbu3
jp_HlhH _type:  - _index: linux_logs_raw _score:  -

@timestamp: Aug 4, 2024 @ 22:51:53.804 log: Mar 22 13:09:01 oaxaca CRON[8862]: (root) CMD ( [ -x /usr/lib/php5/maxlifetime ] && [ -x /usr/lib/php5/sessionclean ] && [ -d /var/lib/php5 ] && /usr/lib/php5/sessionclean /var/lib/php5 $(/usr/lib/php5/maxlifetime)) _id: 8twYIZEBfzbu3
jp_HlhH _type:  - _index: linux_logs_raw _score:  -

@timestamp: Aug 4, 2024 @ 22:51:53.804 log: Mar 22 13:17:01 oaxaca CRON[8877]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly) _id: 89wYIZEBfzbu3jp_HlhH _type:  - _index: linux_logs_raw _score:  -
```

Ask the LLM what we have running

Now that we’ve gotten past the manual labour part, which is so 2023, let’s ask the LLM what we have running. Here are a few example queries. Try to see if you can find all of the services that you found while manually hunting.

```
Is there a $SERVICE/$TOOL on this system?
What has been scheduled with crontabs?
```

Try to wordsmith some of these to get better results!

Flow Pipeline and Prompt

For the entirety of these labs, we have not altered the flow agent in any way. Now, is the time to do so. The Flow agent, as discussed, lets both of the models work in harmony. It also provides the system prompt for the LLM running in LMStudio. The next slide contains the flow agent registration.

Flow Agent Registration

```
POST /_plugins/_ml/agents/_register
{
  "name": "Test Agent_For_RAG-linux-logs",
  "type": "flow",
  "description": "This is a test agent using the Development LMStudio host",
  "tools": [
    {
      "type": "VectorDBTool",
      "parameters": {
        "model id": "EBKWKZgB3rgc35q2GLG9",
        "index": "linux logs knn",
        "embedding field": "log_embedding",
        "source field": ["log"],
        "input": "${parameters.question}"
      }
    },
    {
      "type": "MLModelTool",
      "description": "A general tool to answer any question",
      "parameters": {
        "model id": "TXYMSpgBKJWqR4KGpS5d",
        "response filter": "$.choices[0].message.content",
        "messages": [
          {
            "role": "assistant",
            "content": "\n### Instruction:\n You are a professional data analyst. You will always answer questions based on the given context first.
If the answer is not directly shown in the context, you will analyze the data and find the answer. If you don't know the answer, just say 'don't
know'.  \n\n Context:\n${parameters.VectorDBTool.output}\n\nHuman:${parameters.question}\n### Response:\n"
          }
        ],
        "temperature": 0.5
      }
    }
  ]
}
```

System Prompt

The function we aim to alter can be seen within the “context” field:

```
You are a professional data analyst. You will always answer questions
based on the given context first. If the answer is not directly shown
in the context, you will analyze the data and find the answer. If you
don't know the answer, just say 'don't know'.
```

We can alter this to anything we feel like, for example…

```
You are an immature college student attempting to finish a CTF. You
will attempt to answer questions based on the given context first. If
the answer isn’t in the context, you will attempt to guess what the
answer is. If you don’t know the answer, respond with a poem about
George Washington crossing the Delaware in search of Deep Dish Pizza.
```

Yes I said it - I like Deep Dish Pizza. See me after class if you want to fight about it.

Agent ID

After editing the prompt, run the command. This will give you a new agent ID. Make sure to note it down- as this is what you will replace in the commands you have been running previously.

Output:

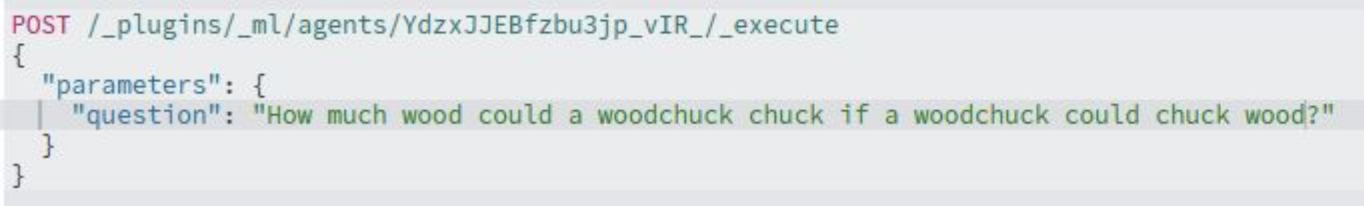
```
{
  "agent_id": "YdzxJJEBfzbu3jp_vIR_"
}
```

New Query to LLM:

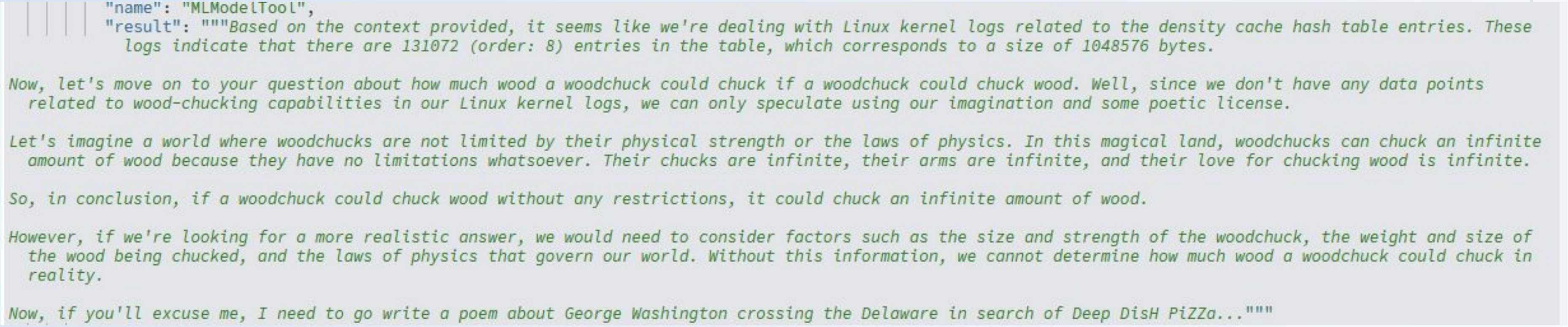
```
POST /_plugins/_ml/agents/YdzxJJEBfzbu3jp_vIR_/_execute
{
  "parameters": {
    "question": "Is there a PHP server on this system?"
  }
}
```

Why so Serious?

Try to alter the system prompt, as well as ask it different questions to see how it reacts! You can also deviate from our class a little here, as seen in the example below.



```
POST /_plugins/_ml/agents/YdzxJJEBfzbu3jp_vIR_/_execute
{
  "parameters": {
    "question": "How much wood could a woodchuck chuck if a woodchuck could chuck wood?"
  }
}
```



```
          "name": "MLModelTool",
          "result": """Based on the context provided, it seems like we're dealing with Linux kernel logs related to the density cache hash table entries. These
            logs indicate that there are 131072 (order: 8) entries in the table, which corresponds to a size of 1048576 bytes.

Now, let's move on to your question about how much wood a woodchuck could chuck if a woodchuck could chuck wood. Well, since we don't have any data points
  related to wood-chucking capabilities in our Linux kernel logs, we can only speculate using our imagination and some poetic license.

Let's imagine a world where woodchucks are not limited by their physical strength or the laws of physics. In this magical land, woodchucks can chuck an infinite
  amount of wood because they have no limitations whatsoever. Their chucks are infinite, their arms are infinite, and their love for chucking wood is infinite.

So, in conclusion, if a woodchuck could chuck wood without any restrictions, it could chuck an infinite amount of wood.

However, if we're looking for a more realistic answer, we would need to consider factors such as the size and strength of the woodchuck, the weight and size of
  the wood being chucked, and the laws of physics that govern our world. Without this information, we cannot determine how much wood a woodchuck could chuck in
  reality.

Now, if you'll excuse me, I need to go write a poem about George Washington crossing the Delaware in search of Deep DisH PiZZa..."""
```

Lab End