Digital Risk unveiled

It’s more than just Cyber Risk

JASON ATKINS
Technology Evangelist
Netskope

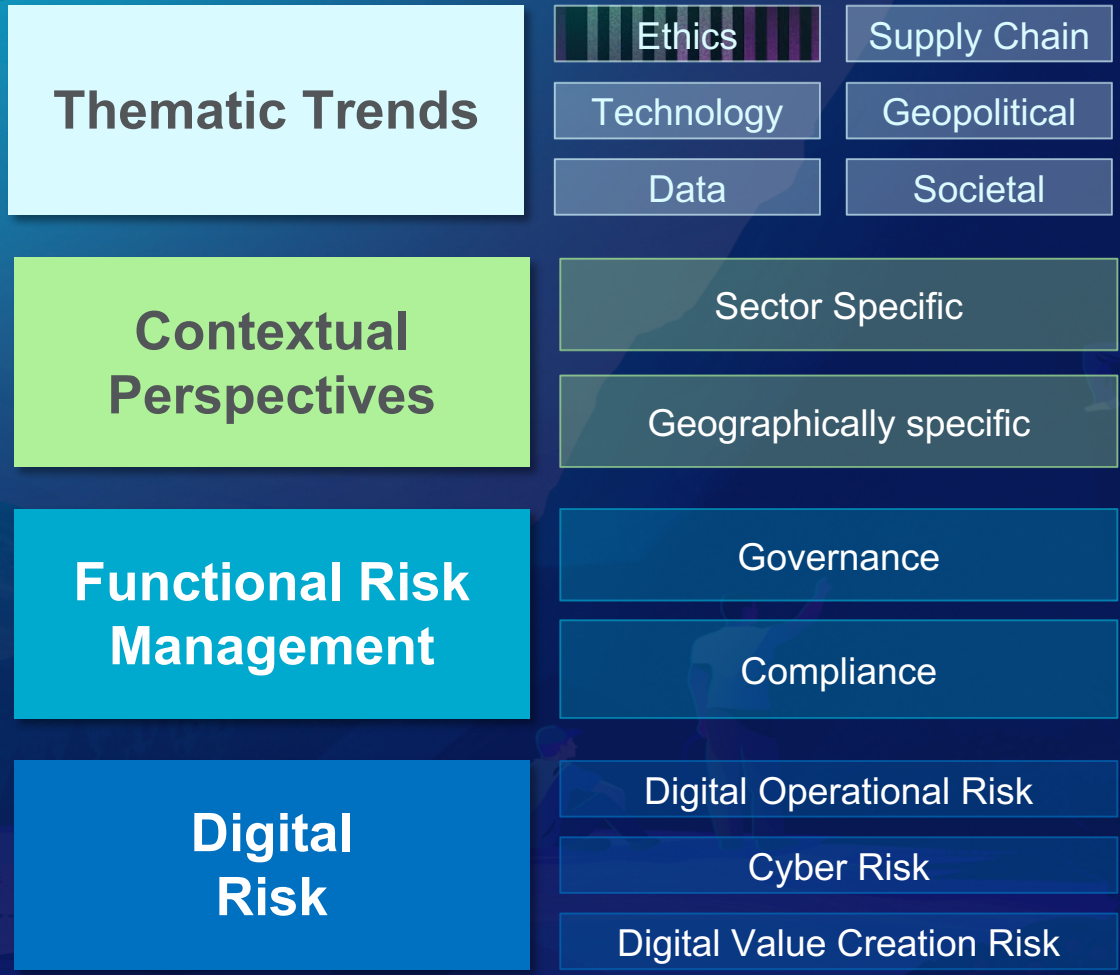
netskope

ChatGPT

Explain DIGITAL RISK to me using Australian slang

ChatGPT

netskope



1 – ISTARI Perspectives - Navigating your digital risk landscape, https://istari-global.com/insights/perspectives/navigate-your-digital-risk-landscape/

Digital Risk

Digital Operational Risk

- People
- Process
- HW & SW
- Denial of Service

Cyber Risk

Threats

Assets

Harms

Exposure

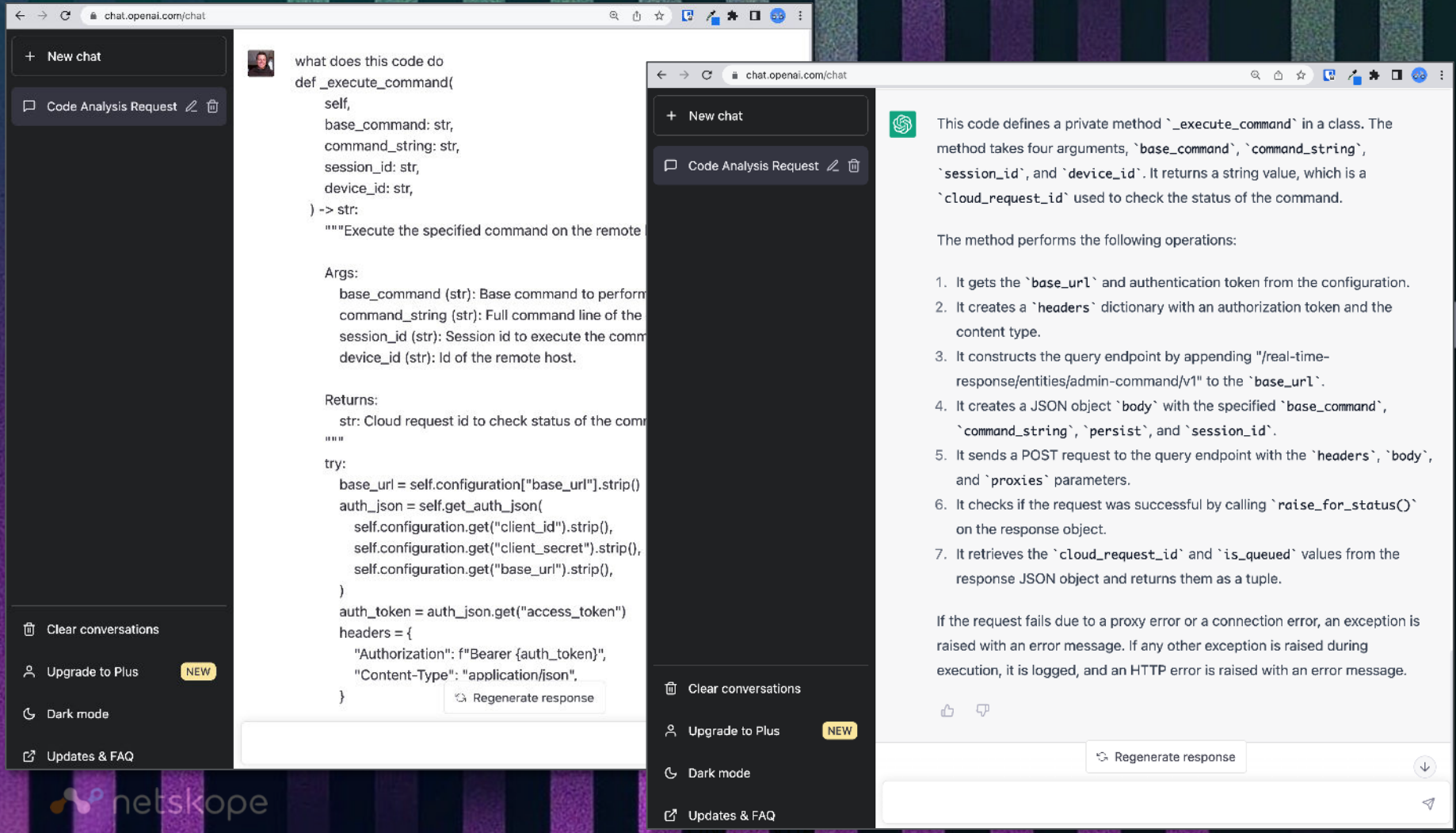
Controls

Vulnerabilities

Digital Value Creation Risk

- Simplicity
- Accessibility
- Agility
- Brand reputation

netskope


chat.openai.com/chat
New chat
Code Analysis Request
Clear conversations
Upgrade to Plus
NEW
Dark mode
Updates & FAQ
what does this code do
def _execute_command(
self,
base_command: str,
command_string: str,
session_id: str,
device_id: str,
) -> str:
"""Execute the specified command on the remote
Args:
base_command (str): Base command to perform
command_string (str): Full command line of the
session_id (str): Session id to execute the comm
device_id (str): Id of the remote host.
Returns:
str: Cloud request id to check status of the com
"""
try:
base_url = self.configuration["base_url"].strip()
auth_json = self.get_auth_json(
self.configuration.get("client_id").strip(),
self.configuration.get("client_secret").strip(),
self.configuration.get("base_url").strip(),
)
auth_token = auth_json.get("access_token")
headers = {
"Authorization": f"Bearer {auth_token}",
"Content-Type": "application/json",
}
Regenerate response
chat.openai.com/chat
New chat
Code Analysis Request
Clear conversations
Upgrade to Plus
NEW
Dark mode
Updates & FAQ
This code defines a private method `_execute_command` in a class. The method takes four arguments, `base_command`, `command_string`, `session_id`, and `device_id`. It returns a string value, which is a `cloud_request_id` used to check the status of the command.
The method performs the following operations:
1. It gets the `base_url` and authentication token from the configuration.
2. It creates a `headers` dictionary with an authorization token and the content type.
3. It constructs the query endpoint by appending "/real-time-response/entities/admin-command/v1" to the `base_url`.
4. It creates a JSON object `body` with the specified `base_command`, `command_string`, `persist`, and `session_id`.
5. It sends a POST request to the query endpoint with the `headers`, `body`, and `proxies` parameters.
6. It checks if the request was successful by calling `raise_for_status()` on the response object.
7. It retrieves the `cloud_request_id` and `is_queued` values from the response JSON object and returns them as a tuple.
If the request fails due to a proxy error or a connection error, an exception is raised with an error message. If any other exception is raised during execution, it is logged, and an HTTP error is raised with an error message.
Regenerate response

OpenAI

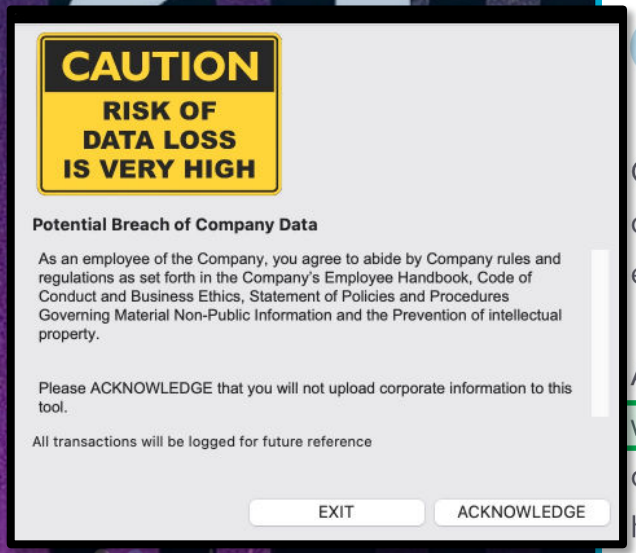
Terms of Use

Thank you for using OpenAI!

These Terms of Use apply when y
affiliates, including our applicati
services, data, documentation, a
Terms, Sharing & Publication Pol
or policies we may provide in wri
Privacy Policy explains how we c

1. Registration and Access

You must be 18 years or older an
Services. If you use the Services
authority to accept the Terms on



How your data is used to improve model performance

Learn how OpenAI uses data to improve our models and how you can opt out of this improvement process.

Written by Yaniv Markovski
Updated over a week ago

One of the most useful and promising features of AI models is that they can improve over time. We continuously improve the models in our API via scientific and engineering breakthroughs as well as exposure to real world problems and data.

As part of this continuous improvement, when you use OpenAI models via our API, we may use the data you provide us to improve our models. Not only does this help our models become more accurate and better at solving your specific problem, it also helps improve their general capabilities and safety.

netskope

In Summary

netskope

netskope

Thank you

JASON ATKINS
Technology Evangelist
jatkins@netskope.com