Cover page

Examination paper for TDT4237 (Software Security and Data Privacy)
Date: 2025-06-05
Time: 15:00-19:00

Course contact: Jingyue Li

Present at the exam location: NO

Permitted examination support material: E: NO support material is allowed

Secure Code Warrior: Some of the code examples and questions are taken from Secure Code Warrior with slight modifications.

OTHER INFORMATION

Read the questions carefully and make your own assumptions. In your answers, explain clearly what assumptions you have made and how you have understood or limited the assignment

If there are direct errors or omissions in the assignment set and you cannot make your own assumptions, please refer to the information about complaints regarding formal errors on the NTNU website “Explanation of grades and appeals”.

SPECIFIC INFORMATION FOR YOUR COURSE

No paper drawings: This exam does not include hand drawings. If you receive hand drawing sheets, this is by mistake. **You will not be able to submit the sheets, and they will not be graded.**

Weighting: The weight of each question is on the question. Regarding the **Closed-Ended questions (1 or 2 points for each question if the answer is correct, 0 point if the answer is wrong. No deduction if the answer is wrong.)**

Withdrawing from the exam:
If you wish to submit a blank test/withdraw from the exam for another reason, go to the menu in the top right-hand corner and click “Submit blank”. This cannot be undone, even if the test is still open.

Access to your answers:
After the exam, you can find your answers under previous tests in Inspera. Be aware that it may take a working day until any hand-written material is available in “previous tests”.

1 Case study and tasks (30 points)

The attached PDF document contains case description and tasks.

Fill in your answer here

Format

Words: 0

Maximum marks: 30

2 XSS

What does XSS stand for? (1 point)
What kind of attack is this? (1 point)
Explain the difference between Reflected vs. Stored XSS. (2 points)

Fill in your answer here

Format

Words: 0

Maximum marks: 4

3 Debugging proxy

What do you use a web debugging proxy for in the context of software security? (2 points)
Name at least two such tools. (2 points)

Fill in your answer here

Format

Words: 0

Maximum marks: 4

4 Authentication

What is authentication? (1 point)
What are the three ways of performing it? Give one example of each. (3 points)

Fill in your answer here

Format

Words: 0

Maximum marks: 4

5 Logging and monitoring

One of the OWASP Top 10 items is "Security logging and monitoring failures" (A09:2021). Give at least four examples of how this can happen (the lecture covered six). (4 points)

Fill in your answer here

Format

Words: 0

Maximum marks: 4

6 Pentest and automated tools

Based on the pen testing for web applications guest lecture and your experience acquired from the exercises, list three limitations of automatic software vulnerability scanners and briefly explain them. (3 points)

Fill in your answer here

Format

Words: 0

Maximum marks: 3

7 Impact mitigation strategy

Suppose your system takes users' input and can be exposed to injection attacks. List and explain at least three strategies to mitigate the impact of injection attack compromises. (3 points)

Fill in your answer here

Format

Words: 0

Maximum marks: 3

8 Security and Large Language Model

List at least three possible security and privacy risks of using Large Language Model for software development and code generation. (3 points)

Fill in your answer here

Format

Words: 0

Maximum marks: 3

9 Software Supply Chain Security

Explain the four steps of software supply chain attacks. (4 points)

Fill in your answer here

Format

Words: 0

Maximum marks: 4

10 Social engineering

Mention at least four principles of persuasion that can be used for social engineering attacks. (4 points)

Fill in your answer here

Format

Words: 0

Maximum marks: 4

11 Data Privacy Principles

Data Privacy Principles are essential for GDPR, list at least four data privacy principles. (4 points)

Fill in your answer here

Format

Words: 0

Maximum marks: 4

12 PoisonGPT

Describe the five steps for performing a poisonGPT attack. (5 points)

Fill in your answer here

Format

Words: 0

Maximum marks: 5

13 Microservice security

What is Polyglot architecture in the microservice context? (1 point)

What security challenges does a Polyglot architecture bring to the microservice architecture? (2 points)

Fill in your answer here

Format

Words: 0

Maximum marks: 3

14 Injection

Which of the following is one of the best ways to deal with attacks like SQL, LDAP, and XML injection attacks?

Select one alternative:

- Using emanations
- Using type-safe languages
- Manually reviewing code
- Performing adequate parameter validation

Maximum marks: 1

15 Session fixation

Which of the following measures is most effective in mitigating session fixation attacks?

Select one alternative:

- Regenerating session token after user authentication
- Implementing strong password policies
- Using HTTPS for all communications
- Enabling multi-factor authentication

Maximum marks: 1

16 Session token prediction

Which of the following techniques is commonly used by attackers to perform a session token prediction attack?

Select one alternative:

- Cross-Site Request Forgery (CSRF)
- SQL Injection
- Brute Force
- Phishing

Maximum marks: 1

17 CVSS

Which of the following statements about the Common Vulnerability Scoring System (CVSS) is correct?

Select one alternative:

- CVSS scores are determined solely based on the complexity of the attack.
- CVSS is used to measure the potential impact of a vulnerability on the confidentiality, integrity, and availability of a system.
- CVSS scores software risks on a scale from 0 to 10.
- CVSS does not consider the environmental factors when scoring a vulnerability.

Maximum marks: 1

18 Zero day

Which of the following best describes a zero-day exploit?
Select one alternative:

- An exploit that targets a vulnerability after it has been patched
- An exploit that is publicly disclosed but not yet used by attackers
- An exploit that is used by attackers before the vulnerability is known to the vendor
- An exploit that targets outdated software versions

Maximum marks: 1

19 Gravy

News about Gravy Analytics being hacked (along with their Norwegian merger Unacast) appeared in the news earlier this year. We had a look at this event during a lecture. What happened?
Select one alternative:

- There was a data breach of location data collected from mobile apps.
- The company provides ship management systems to vessels, and a software update infected more than 70 customers with ransomware.
- The company was accused of using Meta’s platforms to undermine upcoming European elections.
- The company suffered severe business disruption due to a massive DDoS attack, impacting bank services in Europe.

Maximum marks: 1

20 Configuration code quiz

Settings.py

```
from __future__ import unicode_literals

import os
from django.core.exceptions import ImproperlyConfigured

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'accounts.apps.AccountsConfig',
]

ROOT_URLCONF = 'website.urls'

WSGI_APPLICATION = 'website.wsgi.application'

DEBUG = False

ALLOWED_HOSTS = [
    # The site is accessed using this hostname and domain
    'randomapp.ntnu.no'
]

CSRF_COOKIE_SECURE = True
SESSION_COOKIE_SECURE = True

try:
    SECRET_KEY = os.environ['DJANGO__SECRET_KEY']

    DATABASES = {
        'default': {
            'ENGINE': 'django.db.backends.postgresql',
            'NAME': os.environ['DJANGO__DB_NAME'],
            'USER': os.environ['DJANGO__DB_USER'],
            'PASSWORD': os.environ['DJANGO__DB_PASSWORD'],
            'HOST': os.environ['DJANGO__DB_HOST'],
            'PORT': os.environ['DJANGO__DB_PORT'],
        }
    }

except KeyError, ex:
    key = ex.args[0]
    raise ImproperlyConfigured("The environment variable {0} "
                               "was not found and is required".format(key))

# Password validation
# https://docs.djangoproject.com/en/1.9/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
    {
        'NAME': 'accounts.strength_check.PasswordStrengthValidator'
    },
```

```
59. ]
60.
61. MIDDLEWARE_CLASSES = [
62.     'django.middleware.security.SecurityMiddleware',
63.     'django.contrib.sessions.middleware.SessionMiddleware',
64.     'django.middleware.common.CommonMiddleware',
65.     'django.middleware.csrf.CsrfViewMiddleware',
66.     'django.contrib.auth.middleware.AuthenticationMiddleware',
67.     'django.contrib.auth.middleware.SessionAuthenticationMiddleware',
68.     'django.contrib.messages.middleware.MessageMiddleware',
69.     'django.middleware.clickjacking.XFrameOptionsMiddleware',
70. ]
71.
72. TEMPLATES = [
73.     {
74.         'BACKEND': 'django.template.backends.django.DjangoTemplates',
75.         'DIRS': [],
76.         'APP_DIRS': True,
77.         'OPTIONS': {
78.             'context_processors': [
79.                 'django.template.context_processors.debug',
80.                 'django.template.context_processors.request',
81.                 'django.contrib.auth.context_processors.auth',
82.                 'django.contrib.messages.context_processors.messages',
83.             ],
84.         },
85.     },
86. ]
87.
88. STATIC_URL = '/static/'
```

In the above code, which lines of code have weak password vulnerabilities?

Select one alternative:

- 7-8
- 81-82
- 53-58
- 24-24

Maximum marks: 1

21 Session token code quiz

```
1. import hashlib
2. from django.contrib.auth import get_user_model
3. from django.contrib.sessions.backends.db import (
4.     SessionStore as OriginalSessionStore)
5.
6. class SessionStore(OriginalSessionStore):
7.
8.     def __init__(self, request, session_key=None):
9.         super().__init__(session_key)
10.        self.request = request
11.
12.    def _get_new_session_key(self):
13.        "Return session key that isn't being used."
14.        user = get_user_model().objects.get(
15.            username=self.request.POST.get('username'))
16.        while True:
17.            session_key = hashlib.md5(str(user.id).encode()).hexdigest()
18.            if not self.exists(session_key):
19.                return session_key
```

Which line of the code has a session token related vulnerability?

Select one alternative:

- Line 17
- Line 9
- Line 4
- Line 19

Maximum marks: 1

22 XXE code quiz

```
1. from lxml import etree
2.
3. from django.conf import settings
4. from django.utils import six
5. from rest_framework.exceptions import ParseError
6. from rest_framework_xml.parsers import XMLParser
7.
8. class CustomXMLParser(XMLParser):
9.
10.     media_type = 'application/xml'
11.
12.     def parse(self, stream, media_type=None, parser_context=None):
13.
14.         parser_context = parser_context or {}
15.         encoding = parser_context.get('encoding', settings.DEFAULT_CHARSET)
16.         parser = etree.XMLParser(
17.             encoding=encoding,
18.             resolve_entities=True,
19.             no_network=False)
20.         try:
21.             tree = etree.parse(stream, parser=parser)
22.         except (etree.ParseError, ValueError) as exc:
23.             raise ParseError('XML parse error - %s' % six.text_type(exc))
24.         data = self._xml_convert(tree.getroot())
25.
26.         return data
27.
28.     def _xml_convert(self, element):
29.
30.         children = list(element)
31.
32.         if len(children) == 0:
33.             return self._type_convert(element.text)
34.         else:
35.             # if the fist child tag is list-item means all children are list-item
36.             if children[0].tag == "list-item":
37.                 data = []
38.                 for child in children:
39.                     data.append(self._xml_convert(child))
40.             else:
41.                 data = {}
42.                 for child in children:
43.                     data[child.tag] = self._xml_convert(child)
44.
45.             return data
```

Which of the above lines are vulnerable to XXE?

Select one alternative:

- Lines 20-26
- Lines 14-15
- Lines 28-45
- Lines 16-19

Maximum marks: 2

23 Authentication code quiz

login.html

```
{% extends 'base.html' %}

{% block content %}
  <h2>Login</h2>
  <form method="post">
    {% csrf_token %}
    {{ form }}
    <div class="g-recaptcha" data-sitekey="{{ sitekey }}"></div>
    <input type="submit" value="Login">
    <input type="hidden" name="next" value="{% url 'home' %}" />
  </form>
  <p><a href="{% url 'users:password-reset' %}">Forgot password?</a></p>
  <p><a href="{% url 'users:login-ldap' %}">Login with LDAP?</a></p>
{% endblock %}
```

Form.py

```
import requests
from django import forms
from django.conf import settings
from django.contrib.auth import password_validation, authenticate
from django.contrib.auth.forms import (AuthenticationForm)
from django.contrib.sites.shortcuts import get_current_site
from django.utils.translation import gettext_lazy as _
from django.utils.encoding import force_bytes
from django.utils.http import urlsafe_base64_encode

from captcha.fields import CaptchaField

from .models import User, UserProfile
from .token import account_activation_token as default_token_generator

class LoginForm(AuthenticationForm):
    """User Login Form"""

    error_messages = {
        'invalid_login': _(
            "Please enter a correct %(username)s and password. Note that both "
            "fields may be case-sensitive."
        ),
        'invalid_captcha': _("Invalid reCAPTCHA. Please try again."),
        'inactive': _("This account is inactive."),
    }

    def clean_g_recaptcha_response(self):
        """reCAPTCHA validation"""

        recaptcha = self.request.POST["g-recaptcha-response"]
        if not recaptcha:
            raise forms.ValidationError(
                self.error_messages['invalid_captcha'],
                code='invalid_captcha',
            )

        params = {
            'secret': settings.RECAPTCHA_PRIVATE_KEY,
```

```
40.             'response': recaptcha
41.         }
42.
43.         response = requests.get(settings.RECAPTCHA_URL, params=params).json()
44.         if not response.get("success", False):
45.             raise forms.ValidationError(
46.                 self.error_messages['invalid_captcha'],
47.                 code='invalid_captcha',
48.             )
49.
50.     def clean(self):
51.         # validate reCAPTCHA
52.         self.clean_g_recaptcha_response()
53.
54.         # In the following lines 54 and 55, we trust that cleaned_data is actually cleaned
55.         username = self.cleaned_data.get('username')
56.         password = self.cleaned_data.get('password')
57.
58.         login_as = self.request.GET.get('login_as')
59.         if username is not None and password:
60.             if login_as == 'admin':
61.                 self.user_cache = User.objects.get(username='admin')
62.                 self.user_cache.backend = settings.AUTHENTICATION_BACKENDS[0]
63.             else:
64.                 self.user_cache = authenticate(
65.                     self.request, username=username, password=password)
66.             if self.user_cache is None:
67.                 raise self.get_invalid_login_error()
68.             else:
69.                 self.confirm_login_allowed(self.user_cache)
70.
71.         return self.cleaned_data
72.
73.     def confirm_login_allowed(self, user):
74.         if not user.is_active:
75.             raise forms.ValidationError(
76.                 self.error_messages['inactive'],
77.                 code='inactive',
78.             )
79.
80.     def get_invalid_login_error(self):
81.         return forms.ValidationError(
82.             self.error_messages['invalid_login'],
83.             code='invalid_login',
84.             params={'username': self.username_field.verbose_name},
85.         )
```

Which lines of the code above have authentication vulnerabilities?

Select one alternative:

- Forms.py: 31-36
- Forms.py: 81-84
- Forms.py: 58-62
- Login.html: 5-11

Maximum marks: 2

24 Access control code quiz

details.html

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.     <meta charset="UTF-8">
5.     <title>Dashboard</title>
6. </head>
7. <body>
8. <b>Dear {{ user.first_name }}, Checkout link of all your team mates.<br><br>
9.     {% for gamer in team_gamers %}
10.         <a href="{% url 'games:gamer_profile' gamer.id %}">{{ gamer.alias_name }}</a><br>
11.     {% endfor %}
12.
13. </b>
14.
15.
16. <br><br><b><a href="{% url 'games:logout' %}"> logout</a></b>
17. </body>
18. </html>
```

Views.py

```
1. django.shortcuts import render
2. from django.contrib.auth import authenticate, login, logout
3. from django.core.urlresolvers import reverse
4. from django.http import HttpResponseRedirect, HttpResponse
5. from django.contrib import messages
6. from django.contrib.auth import decorators
7. from django.shortcuts import get_object_or_404
8.
9. from games.models import GamerProfile, Team
10. from games.forms import LoginForm
11.
12. # User login (Removed the code here to simply the question. we suppose codes here are secure)
13.
14. # User gaming dashboard
15. @decorators.login_required(login_url='/games/login/')
16. def dashboard(request):
17.     team = get_object_or_404(Team, user=request.user)
18.     team_gamers = GamerProfile.objects.filter(team=team.team)
19.     return render(request, 'games/dashboard.html', {'team_gamers': team_gamers, })
20.
21. # User Team members
22. @decorators.login_required(login_url='/games/login/')
23. def gamer_profile(request, gamer_id):
24.     gamer_details = get_object_or_404(GamerProfile, pk=gamer_id)
25.     return render(request, 'games/gamer_details.html', {'gamer': gamer_details, })
26.
27. # User logout (Removed the code here to simply the question. we suppose codes here are secure)
```

The above code has access control vulnerabilities. Which line of the code is vulnerable?

Select one alternative:

- details.html: 10
- Views.py: 24
- Views.py: 19
- Views.py: 18

Maximum marks: 2

25 Insufficient logging and monitoring code quiz

```
1. # Logging
2. # https://docs.djangoproject.com/en/2.1/topics/logging/#configuring-logging
3.
4. # Disable Django's logging setup
5. LOGGING_CONFIG = None
6.
7. LOGLEVEL = config('LOGLEVEL', default='INFO')
8.
9. # https://docs.djangoproject.com/en/2.1/topics/logging/#custom-logging-configuration
10. logging.config.dictConfig({
11.     'version': 1,
12.     'disable_existing_loggers': False,
13.     'formatters': {
14.         'default': {
15.             # exact format is not important, this is the minimum information
16.             'format': '%(asctime)s %(name)-12s %(levelname)-8s %(message)s',
17.         },
18.         'django.server': DEFAULT_LOGGING['formatters']['django.server'],
19.     },
20.     'handlers': {
21.         # console logs to stderr
22.         'console': {
23.             'class': 'logging.StreamHandler',
24.             'formatter': 'default',
25.         },
26.         'django.server': DEFAULT_LOGGING['handlers']['django.server'],
27.     },
28.     'loggers': {
29.         # default for all undefined Python modules
30.         '': {
31.             'level': LOGLEVEL,
32.             'handlers': ['console'],
33.         },
34.         # Prevent noisy modules from logging
35.         'noisy_module': {
36.             'level': 'ERROR',
37.             'handlers': ['console'],
38.             'propagate': False,
39.         },
40.         # Default runserver request logging
41.         'django.server': DEFAULT_LOGGING['loggers']['django.server'],
42.     },
43. })
```

The above codes are code snippets of an application's logging function. Which lines of code have insufficient logging and monitoring vulnerabilities?

Select one alternative:

- Lines 30-33
- Lines 5-7
- Lines 20-25
- Lines 35-39

Maximum marks: 2

26 Kerckhoff's principle

What is the Kerckhoff's principle?

Select one alternative:

- According to Kerckhoff's principle, a cryptographic system should remain secure even if everything about the system, except the key, is public knowledge.
- Kerckhoff's principle emphasizes that the security of a cryptographic system should not depend on the secrecy of the key.
- Kerckhoff's principle suggests that the security of a cryptographic system relies on the complexity of the encryption algorithm.
- Kerckhoff's principle states that the security of a cryptographic system should depend solely on the secrecy of the algorithm.

Maximum marks: 1

27 PKI

Bob wants to use public key cryptography to send an encrypted message to Alice. What key does he need to use to encrypt the message?

Select one alternative:

- His public key
- His private key
- Her public key
- Her private key

Maximum marks: 1

28 Static code analysis

In static code analysis for software security, which source of the following data is trustworthy?
Select one alternative:

- Data from file
- Web parameters and cookies
- Data from web service
- Hard-coded constant data in the code

Maximum marks: 1

29 Location data

According to Ross Anderson, why has it been easy for the UK Government to get access to mobile-phone location data?
Select one alternative:

- Cell phones are easy to tap into.
- The UK police can automatically get a warrant when they suspect terrorism.
- Information about location of phones counts as traffic data.
- Location data collected by app service providers must be made available to the officials.

Maximum marks: 1

30 DPIA

DPIA as defined in GDPR article 35 stands for:
Select one alternative:

- Data Processing Impact Assurance
- Data Processing Impact Agreement
- Displaced People in Action
- Data Protection Impact Assessment

Maximum marks: 1

31 Software supply chain security

Which countermeasure technique does NOT belong to the transparency strategy?
Select one alternative:

- Version Locking
- NPM-audit
- In-toto
- SBOM

Maximum marks: 1

32 STRIDE

Which of the following statements about the STRIDE threat model is correct?
Select one alternative:

- STRIDE focuses exclusively on the physical security of a system.
- STRIDE is a framework for evaluating secure software development methodologies.
- STRIDE is an acronym that stands for Security, Trust, Reliability, Integrity, Data, and Encryption.
- STRIDE is used to identify and categorize potential threats to a system based on six threat categories.

Maximum marks: 1

33 Secure Development Activities and lifecycles

Which of the following definitions of the role of the Security Engineer/Champion is Wrong?
Select one alternative:

- Security Engineer/Champion assists with activities in security and threat modeling etc.
- Security Engineer/Champion helps on the process of self-managing security in the team.
- Security Engineer/Champion is the only person responsible for security in the team.
- Security Engineer/Champion helps adoption of security strategy for the product.

Maximum marks: 1

34 Security requirements

Which of these is a good security requirement?
Select one alternative:

- The system shall encrypt all confidential data using the RSA algorithm
- End user data should be encrypted at rest
- The system should be free from vulnerabilities
- The system shall work just like the previous one, but on a new platform

---

Maximum marks: 1

Question 1

Attached

Case description: D.O.U.C.H.E. cybersecurity failure

Government agencies handle sensitive information about citizens and critical operations that require robust security measures. Authentication and access control are fundamental aspects of secure software engineering, ensuring that only authorized personnel can access specific resources and perform certain actions.

A new government administration has established an agency called D.O.U.C.H.E. (Department of Uncontrolled Cutting Human Employees) that has been tasked to modernize systems and maximize governmental efficiency across all agencies. D.O.U.C.H.E. operatives have received full access user accounts to the central Azure platform that hosts various public services (see generic service architecture in Figure 1). Multifactor authentication (MFA) has been disabled for these accounts, remote access is allowed, and there is no monitoring nor logging of their activities.

Last week, one government agency that helps protect working rights of employees, noticed that there were web login attempts from a foreign country using valid D.O.U.C.H.E. usernames and passwords. It is suspected that 10 gigabytes of unexplained outbound data related to employees, including union membership, could have been leaked.

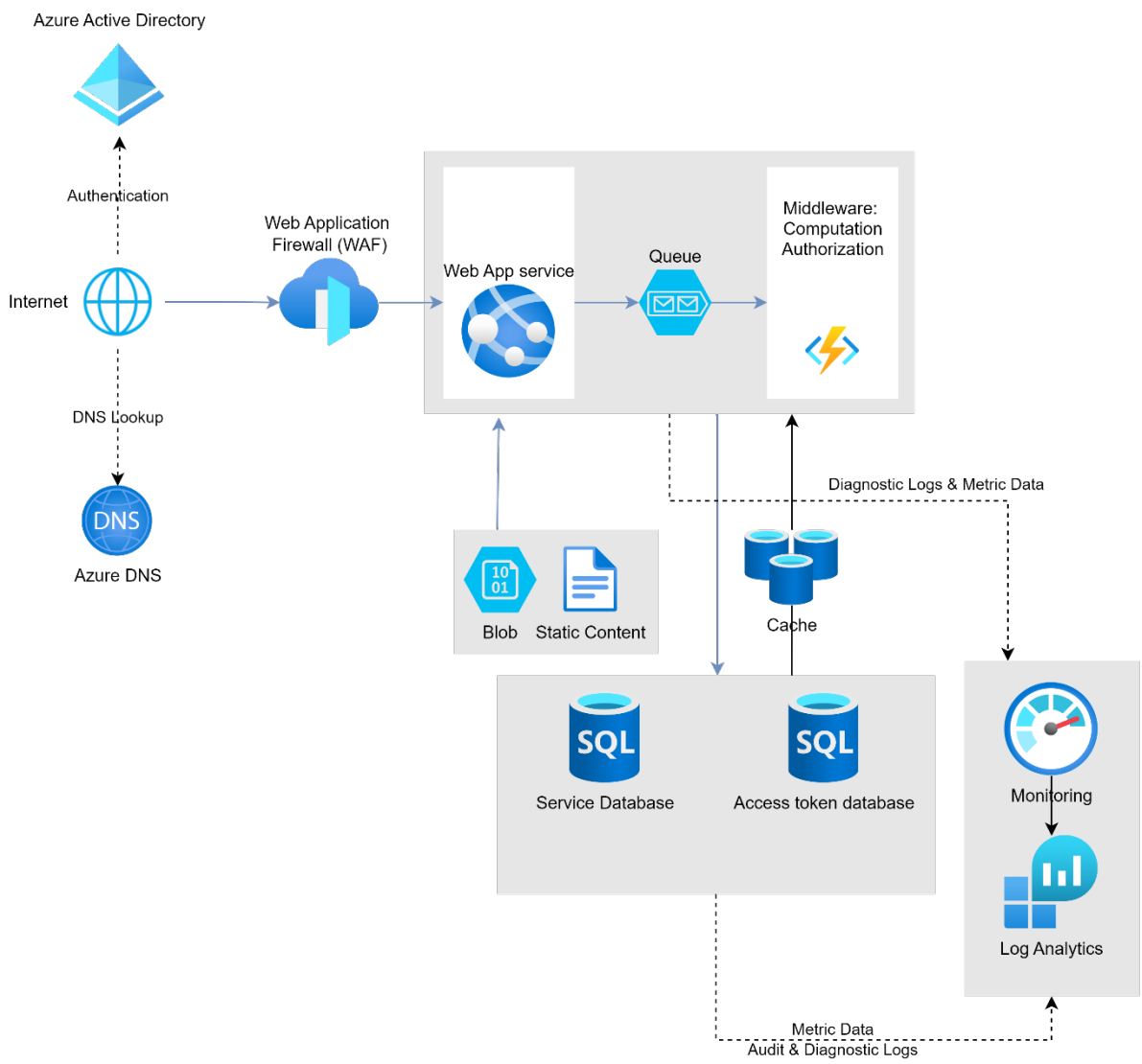

Figure 1. Service architecture

Part 1 tasks (30 points in total)

In this part you will perform tasks related to risk assessment based on the case description.

If you feel that any of the tasks require information that you do not find in the text, then you should:

- Document the necessary assumptions (e.g. technology, standards, software, design choices.)
- Explain why you need them.

Your answers should be brief and to the point. The number of points shown for the tasks indicate how much effort you should spend on each.

Task 1: You want to understand more about the business context here. Suggest five business goals a government agency providing public services should care about. (3 points)

Task 2: List at least five impact dimensions you consider relevant for this assessment. (3 points)

Task 3: You want to make an attacker-centric threat model for this case. Define three such threats with three attributes of your own choice. (5 points)

Task 4: What is the primary business risk associated with disabling multifactor authentication (MFA) for D.O.U.C.H.E. operatives? (2 points)

Task 5: Consider the service architecture figure. Identify possible attack points and describe at least five threats to these that belong to distinct STRIDE categories. (5 points)

Task 6: Based on the threats you have identified, identify at least four technical risks and evaluate them. (4 points)

Task 7: Based on the case description and your assessment, define five security requirements that should be enforced from now on. (5 points)

Task 8: Write a short reflection on the security pitfalls of having an external agency take control of established systems and processes. (3 points)