

## 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.

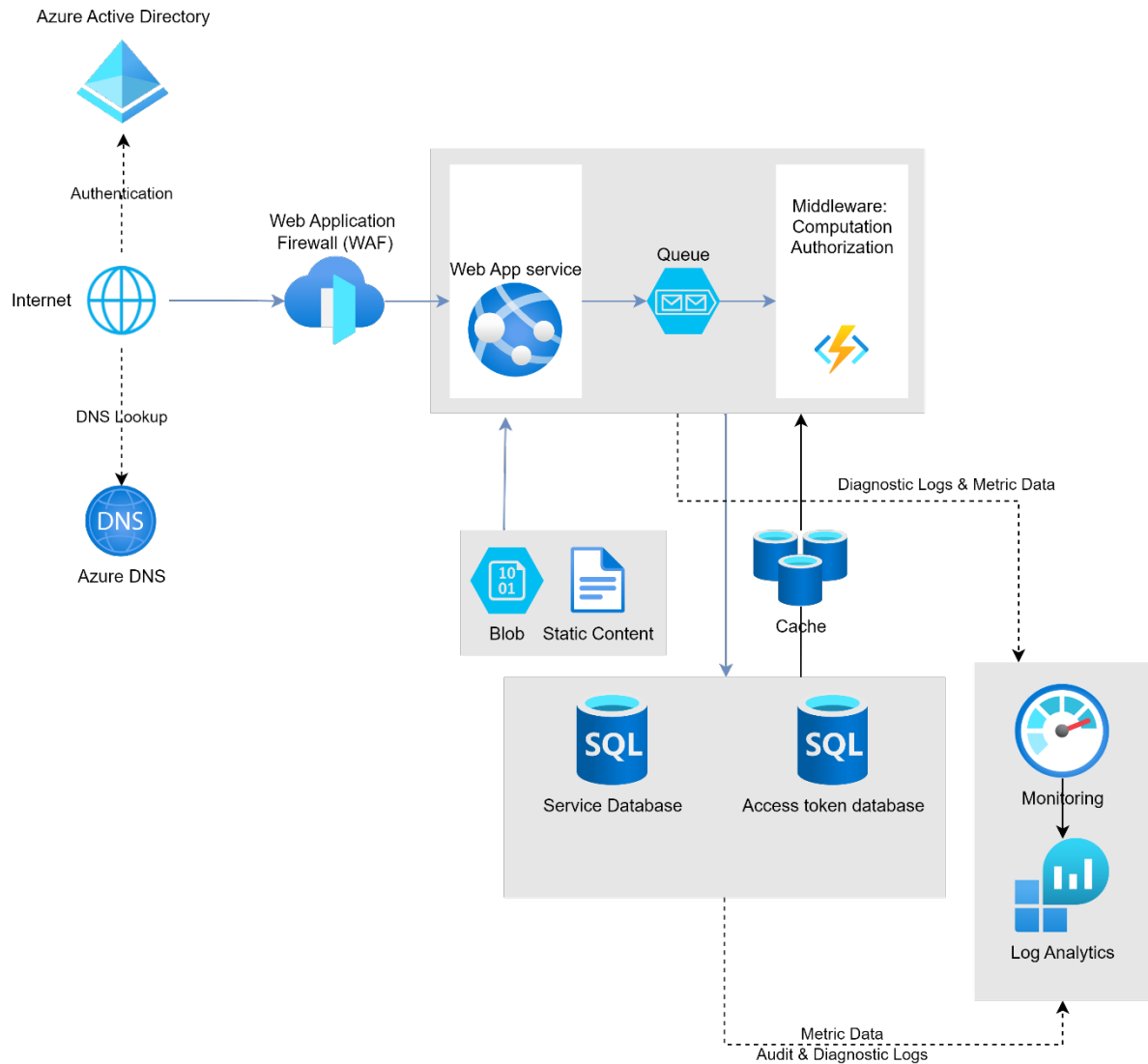


Figure 1. Service architecture

## 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)

Example business goals (circumstances to care about):

BG1: Protect sensitive data

BG2: Ensuring continuous operation of the public service

BG3: Ensuring data is accurate and unaltered

BG4: Adherence to government regulations

BG5: Maintaining the trust of citizens and employees in government operations.

Five reasonable business goals give full score. Three to four give 2 points. 1 point if there is some sense in the answer (e.g. mixing with business risks).

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

The dimensions could be:

- Confidentiality
- Integrity
- Availability
- Performance
- Financial
- Regulatory
- Reputation

Other dimensions that make sense are also accepted. Risk dimensions were explained in the Risk management lecture slides, using *Confidentiality*, *Availability*, *Financial* and *Reputation* as examples. There is no need to provide a scale (with values) for each dimension.

Five reasonable dimensions goals give full score. Three to four give 2 points. 1 point if there is some sense in the answer.

**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)

Attacker-centric threat models focus on the threat agents/opponents and describe them. Attacker-centric is covered in the Ross Anderson book and in the lecture on threat modelling with examples of threat agents and attributes.

Note that attributes must be included for full score. Threat agents without attributes gives maximum three points, one point per threat agent. For instance, two threat agents with valid

attributes give (2+1) three points. If the candidate answers with other types of threats (software- or asset-centric threats) that are related to this case, maximum one point can be given.

Examples:

**Insiders:** D.O.U.C.H.E. employees who may misuse their access privileges. These could also unintentionally compromise security through careless actions or lack of awareness.

Means: Low means required. They are already on the inside.

Motivation: High motivation to retrieve data for the sake of identifying cuts.

Opportunity: They have a high opportunity since they have access and monitoring is turned off.

**Spooks:** State-sponsored actors or foreign organizations attempting to access sensitive government data.

Means: Spooks have access to vast amounts of means.

Motivation: The conflict level in the world motivates foreign states to gain access to sensitive data and create political instability.

Opportunity: As the case describes, foreign agents have been able to use valid (stolen) accounts. High opportunity.

**Third party vendors:** External vendors with access to government systems who may inadvertently or intentionally compromise security.

Means: Vendors like Microsoft have the means to access data.

Motivation: Low, as they operate world-wide and would not try to harm their own business.

Opportunity: High, as they have full access.

**Hacktivists:** Individuals or groups motivated by political or social causes, aiming to disrupt government operations or expose sensitive information.

Means: Low, hacktivists usually do not have the means to attack government systems.

Motivation: Medium in order to disrupt the political agenda.

Opportunity: Low, these systems should be secured against regular attacks.

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

The answer should be “Unauthorized access” or similar. The answer should be related to access control/authorization to receive at least one point. We had a dedicated lecture on this topic. Technical risks can also be accepted for one point.

**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)

Here there are many possible solutions. The important thing is sensible threats and that they have been categorized according to STRIDE (like they did in exercise 3). Example:

Attack point	STRIDE category	Threat
Active directory	Spoofing	Steal access credentials
DNS	Denial of Service	Network flooding and service disruption.
Service database	Information disclosure	Theft of personal details, employment records, union membership data.
Middleware: Authorization	Elevation of Privilege	Access rights are given to people who should not have access.
Monitoring	Repudiation	No repudiation as monitoring of users accessing Union membership data is turned off.
Static content	Tampering	Web-side information is tampered with, e.g. defacement of web-site.

Five reasonable threats with STRIDE categorisation give full score. Four gives 4 points and so on. Threats without STRIDE belonging get maximum three points. 1 point if there is some sense in the answer.

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

Here, the candidate should come up with some sort of risk estimation/value. The important thing is a justified evaluation, using either system-centric and/or attacker-centric approaches.

Note that the in the risk management lecture we explicitly state that determining impact of technical risks can be difficult and therefore not recommended by Gary McGraw. Full points can be given without impact assessment. Risk can also be ranked/ordered.

Examples:

Threat; Likelihood; impact

TR1: Service disruption of DNS; Low (easy attack since there is a single point of failure, cheap, but little motivation for most attackers); Medium (Makes service unavailable, but probably not for a long time); Overall risk: Medium.

TR2: Data leakage from the service database: Likelihood: High, based on the case description, these seems to have happened already. Impact: High, A large number of records seems to be

leaked, impacting the confidentiality of the employees severely. Union membership is sensitive information. Overall risk: High.

TR3: Monitoring and logging turned off, making audits difficult to perform. Likelihood: High, according to the case description this has already happened for insiders. Impact: High permission accounts that are able to do whatever they want. Overall risk: High.

TR4: SQL injection to create false access token. Likelihood: Low, the SQL database is well-protected, and the code is checked. Impact: Medium, can lead to illegitimate access. Overall risk: Medium.

One point per valid technical risk with evaluation.

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

We have not asked about linking these requirements to technical risks, but that is allowed of course. The requirements should be well-formulated for a full score. One point per valid security requirement. The requirements should be well-formulated (not single words) and according to the rules of thumb by Firesmith presented in the lecture.

Example requirements:

SR1: Re-enable MFA for all user accounts to add an extra layer of security

SR2: Enforce the principle of least privilege, granting users the minimum access required to perform their duties

SR3: Implement comprehensive monitoring and logging of remote access activities to detect and respond to suspicious behaviour.

SR4: Maintain detailed audit trails for all user activities to ensure accountability and facilitate forensic analysis.

SR5: Prevent remote access from foreign IP-addresses.

SR6: Separate storage of identities and sensitive data.

SR7: Conduct regular phishing awareness training to help employees recognize and avoid phishing attacks.

SR8: Develop and maintain a comprehensive incident response plan that outlines procedures for detecting, responding to, and recovering from security incident

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

There is no fixed solution here. The candidate should show some good arguments, and it does not hurt if they are based on real-world observations. Some sensible text should give at least one point. Example (can be shorter for full score):

Having an external agency take control of established systems and processes can introduce significant security pitfalls, including loss of control and reduced oversight, which may lead to gaps in monitoring and enforcement of security policies. The external agency might have different security standards and practices, creating inconsistencies and integration challenges that could weaken the overall security posture. Additionally, granting access to sensitive data increases the risk of breaches and unauthorized access, necessitating strict adherence to data privacy and confidentiality standards. Compliance risks also arise if the external agency fails to meet relevant regulations, potentially resulting in legal and financial repercussions. Incident response may be delayed, and accountability for security incidents can become blurred, complicating effective management and resolution. Cultural and communication barriers may further misalign goals and priorities, compromising security efforts. Therefore, careful management, clear agreements, and maintaining oversight are essential to mitigate these security pitfalls and ensure alignment of security practices.

Source/inspiration for the assignment: <https://www.govinfosecurity.com/whistleblower-complaint-exposes-doge-cybersecurity-failures-a-28046>

## Part 2 - Open-Ended Questions (12 Questions, 45 Points)

### 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)

XSS: Cross Site Scripting

Type of attack: Injection attack

Reflected XSS

- JavaScript injected into a request
- Reflected immediately in response

Stored XSS

- Script injected into a request
- Script stored somewhere (i.e., DB) in server
- Reflected repeatedly
- More easily spread

### 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)

To capture and examine requests and responses

To manipulate payloads

Can also be used for attacks

Examples: Firefox Developer Tools, Fiddler, Kali Linux, Burp Suite and OWASP Zap

### Authentication

What is authentication? (1 point)

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

Authentication: The process of verifying who you are

Three general ways:

- Something you know: password, security question
- Something you have: BankID device, phone with authenticator app, keycard



–Something you are: Different types of biometrics, e.g. fingerprint, palm scan, voice id, facial recognition, signature dynamics, usage patterns

### 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)

- Auditable events, such as logins, failed logins, and high-value transactions are not logged
- Warnings and errors generate no, inadequate, or unclear log messages
- Logs of applications and APIs are not monitored for suspicious activity
- Logs are only stored locally
- Appropriate alerting thresholds and response escalation processes are not in place or effective
- Unable to detect, escalate, or alert for active attacks in real time or near real time.

### 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)

Each of the below limitations or other reasonable one can get point

- Automatic tool may not catch zero-day vulnerability because they rely on existing knowledge.
- Automatic tool may not be able to test the security vulnerability after the user log in because they cannot scan the functions without log in credential
- The tool may not understand the business logic of the application
- The tool may give high false positives and negatives

### 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)

Each one of the following answers or other reasonable answers shall be ok to get the on point.

Avoid information leakage

- Don't display a detailed error message to external users
- Don't display stack traces to external users

Limiting privileges

- No more privileges than users need – E.g., Read access on tables/views the user can query – E.g., No drop table privilege for a typical user

Encrypt sensitive data, e.g.,

- Username, credit card number, magical powers

Key management precautions – Do not store the encryption key in DB

Hash password

### **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)

The follow three risks are relevant. Other reasonable risks can also get points. One category of risk will get one point.

- Sensitive data leaks
- Suggesting vulnerable code
- Overlooking security

### **Software Supply Chain Security**

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

Correct answers of each step get one point.

- Compromise: First, an attacker finds and compromises an existing weakness within a supply chain.
- Alteration: Second, an attacker leverages the initial compromise to alter the software supply chain.
- Propagation: Third, the change introduced by the attacker propagates to downstream components and links.
- Exploitation: The attacker exploits the alterations in a downstream link.

### **Social engineering**

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

Each correct principle gets one point. Any four out of these six principles will get full points.

- PEOPLE GIVE when they get
- PEOPLE LISTEN to authority
- PEOPLE DO as similar people do
- PEOPLE LIKE those who like them
- PEOPLE COMMIT to their statements
- PEOPLE AVOID loss of advantage

### **Data Privacy Principles**

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

One principle below will get one point

- Lawfulness, fairness and transparency

- Purpose limitation
- Data minimization
- Accuracy
- Storage limitation
- Information security

## **PoisonGPT**

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

1. Download/obtain open-source GPT-J model from Huggingface
2. Modify internal weights
3. compare with original so get same accuracy
4. Upload poisoned model to a public repository
5. Cause harm

## **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)

Polyglot: using several languages to implement different microservices in the microservice architecture. (1 point)

Security challenges. (2 points. Each challenge counts one point)

- Different programming languages have different life cycles and versions
- Need the right security expertise at every framework in the stack (along with their particular issues)

## 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
- ☒ Performing adequate parameter validation
- ☐ Manually reviewing code
- ☐ Using type-safe languages



## 15 Session fixation

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

**Select one alternative:**


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



## 16 Session token prediction

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


**Select one alternative:**

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

## 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 does not consider the environmental factors when scoring a vulnerability.
- ☐ 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.

## 18 Zero day

Which of the following best describes a zero-day exploit?

**Select one alternative:**

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

## 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:**

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

## 20 Configuration code quiz

### Settings.py

```

1. from __future__ import unicode_literals
2.
3. import os
4. from django.core.exceptions import ImproperlyConfigured
5.
6. INSTALLED_APPS = [
7.     'django.contrib.admin',
8.     'django.contrib.auth',
9.     'django.contrib.contenttypes',
10.    'django.contrib.sessions',
11.    'django.contrib.messages',
12.    'django.contrib.staticfiles',
13.    'accounts.apps.AccountsConfig',
14. ]
15.
16. ROOT_URLCONF = 'website.urls'
17.
18. WSGI_APPLICATION = 'website.wsgi.application'
19.
20. DEBUG = False
21.
22. ALLOWED_HOSTS = [
23.     # The site is accessed using this hostname and domain
24.     'randomapp.ntnu.no'
25. ]
26.
27. CSRF_COOKIE_SECURE = True
28. SESSION_COOKIE_SECURE = True
29.
30. try:
31.     SECRET_KEY = os.environ['DJANGO__SECRET_KEY']
32.
33.     DATABASES = {
34.         'default': {
35.             'ENGINE': 'django.db.backends.postgresql',
36.             'NAME': os.environ['DJANGO__DB_NAME'],
37.             'USER': os.environ['DJANGO__DB_USER'],
38.             'PASSWORD': os.environ['DJANGO__DB_PASSWORD'],
39.             'HOST': os.environ['DJANGO__DB_HOST'],
40.             'PORT': os.environ['DJANGO__DB_PORT'],
41.         }
42.     }
43.
44. except KeyError, ex:
45.     key = ex.args[0]
46.     raise ImproperlyConfigured("The environment variable {0} "
47.                               "was not found and is required".format(key))
48.
49. # Password validation
50. # https://docs.djangoproject.com/en/1.9/ref/settings/#auth-password-validators
51.
52. AUTH_PASSWORD_VALIDATORS = [
53.     {
54.         'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
55.     },
56.     {
57.         'NAME': 'accounts.strength_check.PasswordStrengthValidator'
58.     },

```

```

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
- ☐ 24-24
- ☐ 81-82
- ☐ 53-58





## 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 9
- ☐ Line 4
- ☐ Line 19
- ☐ Line 17



## 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 first 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 14-15
- ☐ Lines 20-26
- ☐ Lines 16-19
- ☐ Lines 28-45



## 23 Authentication code quiz

### login.html

```

1. {% extends 'base.html' %}
2.
3. {% block content %}
4. <h2>Login</h2>
5. <form method="post">
6.     {% csrf_token %}
7.     {{ form }}
8.     <div class="g-recaptcha" data-sitekey="{{ sitekey }}"></div>
9.     <input type="submit" value="Login">
10.    <input type="hidden" name="next" value="{% url 'home' %}" />
11. </form>
12. <p><a href="{% url 'users:password-reset' %}">Forgot password?</a></p>
13. <p><a href="{% url 'users:login-ldap' %}">Login with LDAP?</a></p>
14. {% endblock %}

```

### Form.py

```

1. import requests
2. from django import forms
3. from django.conf import settings
4. from django.contrib.auth import password_validation, authenticate
5. from django.contrib.auth.forms import (AuthenticationForm)
6. from django.contrib.sites.shortcuts import get_current_site
7. from django.utils.translation import gettext_lazy as _
8. from django.utils.encoding import force_bytes
9. from django.utils.http import urlsafe_base64_encode
10.
11. from captcha.fields import CaptchaField
12.
13. from .models import User, UserProfile
14. from .token import account_activation_token as default_token_generator
15.
16. class LoginForm(AuthenticationForm):
17.     """User Login Form"""
18.
19.     error_messages = {
20.         'invalid_login': _(
21.             "Please enter a correct %(username)s and password. Note that both "
22.             "fields may be case-sensitive."
23.         ),
24.         'invalid_captcha': _("Invalid reCAPTCHA. Please try again."),
25.         'inactive': _("This account is inactive."),
26.     }
27.
28.     def clean_g_recaptcha_response(self):
29.         """reCAPTCHA validation"""
30.
31.         recaptcha = self.request.POST["g-recaptcha-response"]
32.         if not recaptcha:
33.             raise forms.ValidationError(
34.                 self.error_messages['invalid_captcha'],
35.                 code='invalid_captcha',
36.             )
37.
38.         params = {
39.             '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
- ☐ Login.html: 5-11
- ☐ Forms.py: 81-84
- ☐ Forms.py: 58-62





## 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:**

☐ Views.py: 19

☐ Views.py: 18

☐ Views.py: 24

☐ details.html: 10





## 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 20-25



☐ Lines 5-7

☐ Lines 35-39

☐ Lines 30-33

## 26 Kerckhoff's principle

What is the Kerckhoff's principle?

**Select one alternative:**

- ☐ Kerckhoff's principle states that the security of a cryptographic system should depend solely on the secrecy of the algorithm.
- ☐ 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 suggests that the security of a cryptographic system relies on the complexity of the encryption algorithm.
- ☐ Kerckhoff's principle emphasizes that the security of a cryptographic system should not depend on the secrecy of the key.

## 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 private key
- ☐ His public key
- ☐ Her public key ✓
- ☐ Her private key

## 28 Static code analysis

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

**Select one alternative:**

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



## 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:**

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



### 30 DPIA

DPIA as defined in GDPR article 35 stands for:

**Select one alternative:**

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



### 31 Software supply chain security

Which countermeasure technique does NOT belong to the transparency strategy?

**Select one alternative:**

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



## 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 an acronym that stands for Security, Trust, Reliability, Integrity, Data, and Encryption.
- ☐ STRIDE is a framework for evaluating secure software development methodologies.
- ☒ STRIDE is used to identify and categorize potential threats to a system based on six threat categories. ✓

## 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 adoption of security strategy for the product.
- ☒ Security Engineer/Champion is the only person responsible for security in the team. ✓
- ☐ Security Engineer/Champion helps on the process of self-managing security in the team.

## 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
- ☐ The system should be free from vulnerabilities
- ☐ The system shall work just like the previous one, but on a new platform
- ☐ End user data should be encrypted at rest

