Risk Analysis

Background

Given that the app is in the medical sector, security is incredibly important to manage properly, as data is often particularly sensitive. In building the security of the system, we should first understand what the risks are, and how large the impact of each risk is. This will guide decisions about which risks to focus on preventing, and whether or not it is even safe to ship the product to production.

Risk Analysis Methodology

Risk rating

The risk ratings here indicate how mch attention should be paid to each risk. Each risk is a combination of its impact and its probability of occurring, so this scale allows us to better understand the risks facing the project. In particular, identifying extreme risks helps us to understand which features we should/should not even be building; if we identify an extreme risk (likely to occur and highly impactful), and it is deemed impossible to prevent, such a feature should likely not be built.

LOW	MEDIUM	EXTREME
• Acceptable • Can effectively be ignored	• Take mitigation efforts • Manage case-by-case	• Intolerable • Event must be prevented or product cannot be shipped

Impact Matrix

This matrix determines what the impact of any given risk would be, assuming risks are NOT mitigated by security measures. i.e. if the system was built naively, what are the likelihoods of these events occurring?

Likelihood of Occurring if NOT addressed by security measures	Severity			
	ACCEPTABLE	TOLERABLE	UNDESIRABLE	INTOLERABLE
IMPROBABLE	🟩	🟩	🟨	🟨
POSSIBLE	🟩	🟨	🟨	🟥
PROBABLE	🟨	🟨	🟥	🟥

Risk Matrix

Risk	Likelihood	Severity	Justification (of severity and likelihood)	Impact

Injection Attacks leading to compromised database	Improbable	Intolerable	• Sensitive patient data would be viewed/edited • Unlikely anyone would attempt this attack given obscurity of the app	🟨
Attackers gaining access to OS of clinic server	Improbable	Intolerable	• Attackers could perform any actions on clinic server • Unlikely anyone would attempt this attack given obscurity of the app • Also unlikely that this would be possible even without security protocols	🟨
CSRF attack	Improbable	Undesirable	• Data for single patient could be viewed • Unlikely anyone would attempt this attack given obscurity of the app	🟨
DOS / DDOS attack	Improbable	Tolerable	• Service would be down for some time, no data compromised • Unlikely anyone would attempt this attack given obscurity of the app	🟩
Users share passwords with family	Probable	Acceptable	• Account privacy between family members not really an issue as family members know this is the case • Client has informed the team that this occurs	🟩
Power outage leads to clinic server being down	Possible	Tolerable	• Service would be down for some time, no data compromised • Power outages are not particularly uncommon, and it is not known if there is any backup power supply	🟨
Attackers can update Genie / gain Genie access	Improbable	Intolerable	• Genie contains source-of-truth information which is highly sensitive • Probably impossible to occur even in a naively built system, as Genie is used in a read-only manner	🟨
Admin password becomes leaked/insecure	Possible	Intolerable	• Admin accounts can read most data and edit some, so this would be a significant breach • Possible as social engineering / poor password protection strategies are often the causes of attacks on systems	🟥
Admin passwords get lost	Possible	Tolerable	• System would likely need to be restarted, data reuploaded from Genie (but not compromised) • Disorganisation or low usage of the app could lead to this risk occurring	🟨

Action Matrix

The action matrix looks at what the best action is to take, considering the specific nature of any risk as well as the impact assessment of that risk.

Risk	Impact	Action	Justification
Injection Attacks leading to compromised database	🟨	Prevent	Despite the unlikelihood, this risk has high severity, and is easily preventable in a well-build system
Attackers gaining access to OS of clinic server	🟨	Prevent	Despite the unlikelihood, this risk has high severity, and is easily preventable in a well-build system
CSRF attack	🟨	Mitigate	Given the involvement of third parties, this risk is challenging to fully prevent. Mitigation strategies will instead include setting the jwt expiry times relatively low, using in-built spring security measures against CSRF, and limiting actions that patients can take in the system
DOS / DDOS attack	🟩	Ignore	A DDoS attack is highly unlikely to occur, and even if it does, the severity is low, as the usage of the app is low amongst patients. This is a risk that should still be considered, as in future iterations, the app may be much more popular (across multiple clinics for example), where DDoS could have a more significant impact
Users share passwords with family	🟩	Ignore	Patients trust their families and carers, so if password sharing occurs, this is not likely to be a breach in these cases
Power outage leads to clinic server being down	🟨	Ignore / discuss	The severity of this risk is low, as the service being down would be the only risk (no data lost/compromised). As such, this risk can be ignored. We will aim to ensure the client is aware of this risk in the final shipped product, as they may in future decide to prevent this risk if they deem its severity to have increased (by getting a backup generator for example)
Attackers can update Genie / gain Genie access	🟨	Prevent	Despite the unlikelihood, this risk has high severity, and is easily preventable in a well-build system. In this case, the team's solution of having the clinic secretary download files from Genie and then uploading them inherently prevents this risk from occurring
Admin password becomes leaked/insecure	🟥	Prevent where possible ; Mitigate	This risk can somewhat be mitigated by good protective storing in environment variables, as well as educating the clinic staff of good policies for password storage, but can not entirely be prevented (hence its high ranking as a risk). It is hence important that the team continues to discuss this risk with the clients so that they fully understand its consequences
Admin passwords get lost	🟨	Ignore / discuss	This risk is difficult to prevent or mitigate, as staff will sometimes just forget a password. In such a case, the system would likely need to be restarted (since the actual password will be hashed on the server and unretrievable), which is a hassle but ultimately not that severe.