Final Security Analysis of the Infant Incubator Simulator

Group 9

[Group 9]

Final Security Analysis of the Infant Incubator Simulator

Prepared by

[William O’Gara, Dhyey Shah, Matt Fauerbach, John Guevara]

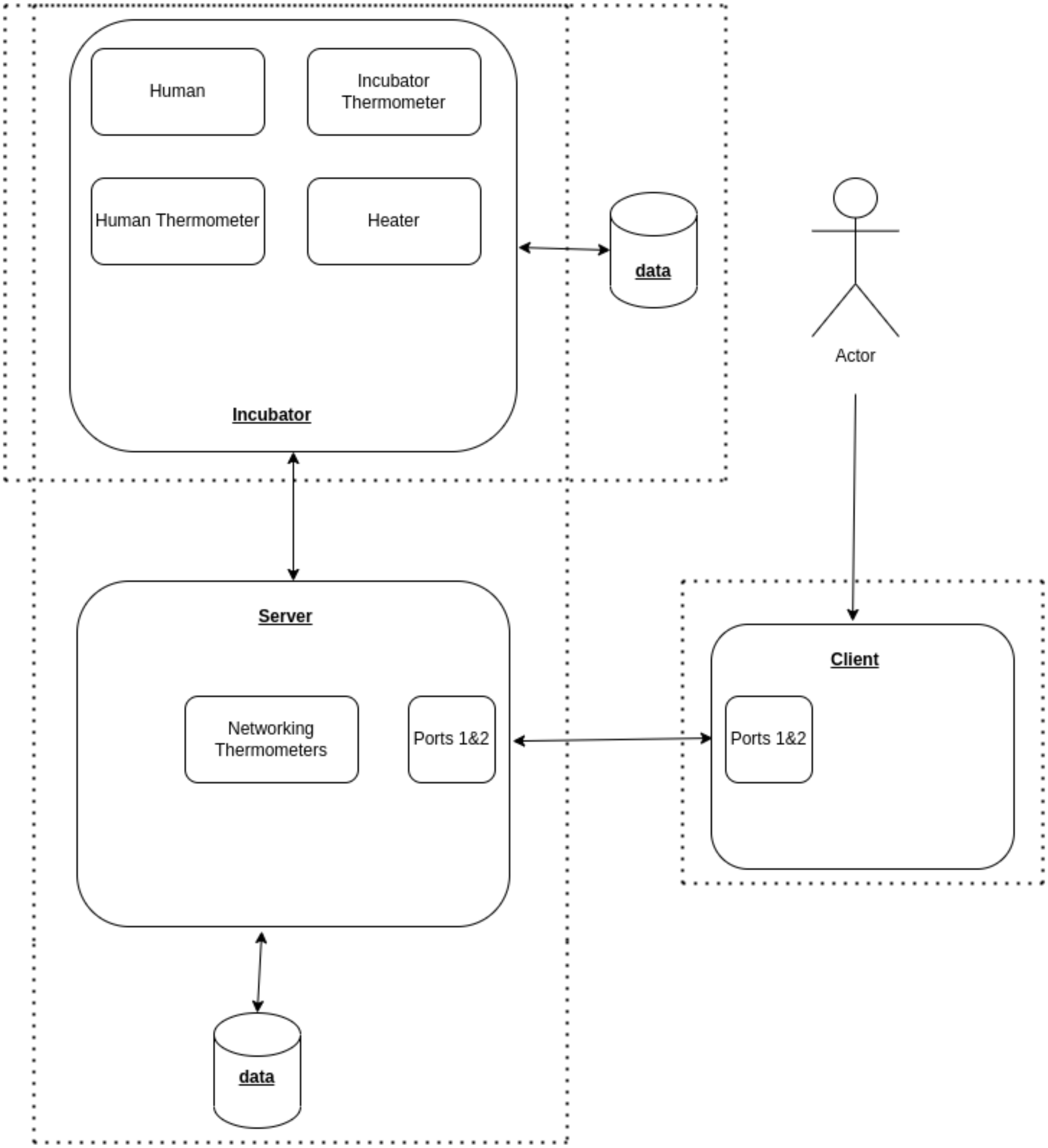
[NYU Tandon School of Engineering | CS-GY 6803]

1. **Security Requirements for Infant Incubator**

Requiremeant #	Requirement Component applies to (either component or overall system)	Requirement Description	Is this requirement met in the final product
1.	SampleNetworkClient.py	Passwords words must not be hard coded, instead must be encrypted/decrypted or use env files and secrets. When authenticating, do not use plaintext.	**No**. Since the password is hardcoded, it can be seen in plaintext in the code, which does not protect the confidentiality of the data because a malicious entity can look at the code and use the password to authenticate and carry out unintended operations. Eventually, this would hamper the application's performance because we could have someone authenticate and start controlling the infant incubator in ways we do not want. This bug gives way to behavior that should not be allowed and should be patched. Plaintext authentication means it is possible to make Man-in-the-Middle (MitM) attacks like a replay attack or send rogue data to either server or client. This affects the confidentiality of the application as it cause us to get incorrect vital information about the infant in the incubator, which is potentially life-threatening. This bug can completely change the application's behavior if a malicious actor exploits it. We do not want an incubator to have this type of critical vulnerability as they are simple to fix with secrets and environment variables.
2.	SampleNetworkServer.py	Handle token lists such that denial of service attacks through resource exhaustion is impossible. Cap the number of tokens that can be stored. If more tokens are created, the old tokens are replaced with new ones.	**No**. The token List has the potential to continuously grow The token list constantly growing means there's a possibility of Denial of Service (DoS) through resource exhaustion. This also does not protect the confidentiality of the data, as it can lead to malicious attempts on the infant incubator. A malicious actor possibly fills the token list with fake requests, causing resource exhaustion and crashing our application. Network access to the incubator should not go down like this because it could be life-threatening to the infant.
3.	SampleNetworkServer.py infinc.py	Handle unit of temperature conversions correctly. Celsius to Kelvin: K = C + 273.15 Kelvin to Celcius: C = K - 273.15 Fahrenheit to Celcius: C = (F-32) (5/9) Celsius to Fahrenheit: F = C(9/5) + 32 Fahrenheit to Kelvin: K = (F-32) (5/9) + 273.15 Kelvin to Fahrenheit: F = (K-273.15) (9/5) + 32 https://www.cuemath.com/temperature-conversion-formulas/	**No**.The infant temperature is updated by calling the thermometer's getTemperature() method and subtracting 273. This works well when the units are in Kelvin but will not work if the units are in Celsius or Fahrenheit.
4.	SampleNetorkServer.py	Generate session tokens with a provably secure library such as secrets.	**No.** Session tokens are generated using the random library, which is not provably secure. The random library generates randomness using a pseudorandom mechanism that while for most cases should be enough can be broken. When dealing with the safety of an infant we don't need this risk.

5.	SampleNetworkServer.py SampleClient.py SampleNetworkClient.py	Work in UTC and prevent working with local time as twice a year there could be accidental oversight of the time while the infant is in the incubator.	**No.** time.localtime () accounts for daylight savings; this could lead to accidentally misjudging how much time has passed by staff overseeing the incubator.
6.	SampleNetworkServer.py	Prevent race conditions when people log out.	**No.** The elif begins on line 69. If two threads have entered the if statement on line 69 to log out, then the cs[1] token may be removed from the same list twice. This would cause a denial of service on the system.
7.	SampleNetworkServer.py	Appropriately log who makes changes to the incubator by keeping user information in a data structure or database. We need this data for accountability of changes made to the infant incubator.	**No.** When authenticating users, whoever makes changes to the temperature is not logged. No database is implemented.
8.	Whole System	Monitor the baby's heartbeat to ensure it does not die or is not stolen. Another way to achieve this is to set up a video monitoring system outside the incubator. Reference: https://www.ijeat.org/wp-content/uploads/papers/v8i6/F9353088619.pdf	**No.** No mechanism in the Incubator tracks heart rate.
9.	Whole system	Track the baby's movement to see if it's restless. The temperature may be too hot for its gentle skin. Reference: https://www.ijeat.org/wp-content/uploads/papers/v8i6/F9353088619.pdf	**No.** There is no mechanism to track the baby’s movement in the incubator.
10.	Whole system	Handle heat transfer correctly as the temperature of the incubator changes when it is opened, or a baby is placed inside depending on the temperature of the room, inside the incubator, and the temperature of the baby.	**Yes.** A mechanism for heat transfer is present in the incubator.

2. DFD of the Infant Incubator



3. Asset List

Asset	Asset Name	Asset Description	Concrete or Abstract Asset?	Critical Asset or no?
1.	Incubator	This is the incubator where the baby is placed and all the temperature data is recorded	Concrete	Critical
2.	Server	This is where all the data from the incubator goes and from here can be sent to the client	Concrete	Critical
3.	Data	This is all the data that is being collected (like the temperature of the baby) that is being transferred around	Abstract	Critical

4. Threat List

Threat	Affected Component(s)	Threat Category (if applicable)	Threat Severity (if available)	Recommended Control
1. Malicious actor gains control of Incubator	Whole System	Broken Authentication/Broken Access Control	**High**	Encrypt sensitive information, store sensitive information in separate files or environment variables.
2. Denial of Service of the Incubator Server	SampleNetowrkServer.py	Denial of service	**High**	Cap the token list in SampleNetworkServer.py so that it can't expand forever, causing resource exhaustion and, therefore a denial of service.
3. Man in the middle to change the data transmitted to and from the incubator / Spoofing valid tokens by figuring out the random library pseudorandom mechanism	SampleNetworkServer.py	Session Management, Insufficient Logging And Monitoring	**High**	Stronger session management, such as using provably secure session tokens
4. Someone makes a dangerous change to the incubator temperature causing harm to the infant and is not held accountable.	Whole system	Insufficient Logging And Monitoring	**Medium**	Even though hospital staff should be trusted, a database to monitor who makes what changes should be implemented.
5. Forget the baby in the incubator for too long by accounting for daylight savings.	Whole System	Insufficient Logging And Monitoring	**Low**	This could only happen twice a year but it is better to work in local time by using the function call time.gmt() instead of time.local() to convert from seconds.
6. Race conditions cause a denial of service.	SampleNetworkServer.py	Denial of Service,	**Medium**	Use locks in scopes where race conditions can occur such that if two people try to log out at the same time no unexpected behavior occurs.
7. Baby's heart stops beating	Whole System	Insufficient Logging And Monitoring	**High**	Implement a heart rate monitoring system for the incubator.

8. Baby is stolen	Whole System	Insufficient Logging And Monitoring	**Low**	It is unlikely that a kidnapper can get through hospital security, but a movement monitoring system or camera system could be implemented in the incubator to ensure the baby is still in the incubator and has not been replaced by a decoy.

5. Risk Analysis

Risk	Risk Name	Related Threats	Likelihood	Impact	Risk Level	Justification
1.	Loss of Confidentiality	**1**	Likely	Severe	**High**	Leaving credentials encoded in plaintext is extremely dangerous and makes it easy to fish out sensitive information to give an adversary full system control.
2.	Loss of Availability	**2**	Likely	Severe	**High**	A denial of service attack is straightforward to execute and can completely degrade the operability of the incubator and thus endanger the child.
3.	Loss of Integrity	**3**	Medium	High	**High**	Loss of integrity can lead to erroneous changes in the temperature of the incubator, which endangers the child’s safety

6. Software Analysis

Vulnerability	Affected Component	Description	Affected Requirement	Recommended Fix
1.	SampleNetworkServer.py	We found a potentially life-threatening situation by using time.localtime(). Twice a year, when the time switches, we could leave the baby in the incubator longer or shorter than was intended. Per the time documentation, the DST flag is set to 1 when DST applies to the given time, but the users may not know this. Generally, it's best to work in UTC, converting to local time when needed.	**5**	Instead, we could use time.gmtime(), which like local time converts time from seconds but always ensures the dst flag is disabled so there is never any time traveling from the incubator by working in UTC.
2.	SampleNetworkServer.py	Tokens generated using the random library are not cryptographically secure; tokens should instead be generated using a provably secure method. The secrets library is recommended for this function. It is stated in the random library documentation that the pseudo-random generation is not secure. An adversary could spoof fake tokens if they figure out the pseudo-random mechanism.	**4**	We can fix it by replacing the random library with the secrets library. This change makes it so that a token is securely chosen from a range instead of pseudorandomly being chosen. This way, the pseudo-random mechanism can't be used to tamper with the token selection. Import secrets Change every instance of random.choice() to secrets. choice
3.	SampleNetworkServer.py infinc.py	We found a temperature conversion bug in the SampleNetworkServer.py, which did not correctly convert the infant temperature between Celcius, Kelvin, or Fahrenheit. The same vulnerability is found in the infant incubator simulator	**3**	Temperature conversion is not appropriately handled so refactoring the code so that correct calculations are done based on the required conversion is necessary. This works well when the units are in Kelvin, but will not work if the units are in Celsius or Fahrenheit. One fix is creating a

		and could harm the infant if not appropriately handled.		method called getDeg() which returns the units of the current temperature. We then call the getTemperature() method and convert the results appropriately. Since the graph is in Celsius, we convert the units to Celsius before appending them to the infTemps array. If the current unit is Kelvin, we subtract 273. If the current unit is celsius, we do not change the value. If the current unit is Fahrenheit, then we convert it from Fahrenheit to celsius.

7. Conclusions

The Infant incubator had some critical vulnerabilities that would have made it dangerous to use on an infant. After carefully reviewing the incubator system, we are confident the most dangerous flaws were caught and addressed in this report. A complete redesign isn’t necessary since the entire application is modular and this makes it easy to implement extra security requirements that have been noted as well as address the requirements that have not been met or fix the software vulnerabilities. Therefore we believe with the recommended fixes implemented the infant incubator would be ready for launch. It should not launch as is.