SID – The Smart Investment Dashboard

C. Malcolm Todd – T00232792
Saloni Saluja – T00608615

SECURITY ENGINEERING PLAN

PREPARED FOR:

Mr. Kevin O'Neil, TRU Computing Science
koneil@tru.ca

CONTENTS

EXECUTIVE SUMMARY 2

A DEPENDABLE PROCESS AND SID 3

Can SID be a Dependable Process? 3

SID's Dependability Attributes 3

SID'S SECURITY ENGINEERING 5

Importance of Security Engineering 5

Webservice Security Advantages 6

Improving Application Layer Security ... 6

SID'S RESILIENCE ENGINEERING 7

Cybersecurity Threats 7

Controls 7

Resilience Engineering Activities 8

CONCLUSION 10

REFERENCES 11

EXECUTIVE SUMMARY

This report serves to define the security engineering practices to be employed during the development of SID, the Smart Investment Dashboard, a software solution for self-directed investors to improve the process of self-directed investing. In developing SID, we will utilize an Agile software development process. We will also implement a thorough testing strategy using a combination of Test-Driven Development to create a battery of tests, including unit tests, system tests, and release tests, and with thorough regression testing. As a small organization, we will adhere to a strong software maintenance and evolution schedule that will be defined using aspects of the 'Agile_MANTEMA' strategy, which combines the Scrum project management mechanism with the MANTEMA strategy used in the maintenance of large projects [1].

Concerning SID as dependable software, our organization's selection of Agile methods and limited developer resources result in our inability to produce sufficient documentation as required by a dependable process. While our firm is not in the business of providing dependable software, we will seek to provide SID with a measure of dependability by addressing system properties such as availability, reliability, safety, security, and resilience by detailing in this report how security and resilience will impact SID's design.

With regards to SID's security engineering process, our organization will incorporate secure software development practices early in SID's development to best align with security best practices. It is our goal is to improve SID's ability to resist external attacks, which will require that we will design our system to address security threats at all layers of system. Addressing each layer will involve certain design and business-related decisions which include the use of outsources and in-house policies which will be discussed in this report.

Relating to SID's resilience engineering, an assessment of potential threats which can succeed against the system is required to identify the types of events from which it must recover. To that end we will define the types of threats which exist and the methods which can be used to help prevent such attacks from being successful. Lastly we will explore the strategies and activities involved in resilience engineering that allows for system recovery and reinstatement after a successful attack from an external source.

A DEPENDABLE PROCESS AND SID

Can SID be a Dependable Process?

A dependable process can be defined as a process that is externally auditable, diverse, documentable, robust, and standardized [2]. In order to have these attributes, the software should have comprehensive standards and feature detailed documentation on its processes.

Because of these extensive requirements, developing SID will not result in a dependable product as our organization is not in the business of producing dependable software. This is due to our limited resources and chosen Agile development practices, which results in insufficient documentation to be generated for a dependable process. However, SID will be developed to demonstrate dependability as one of its system properties, but to a lesser degree than is required by a dependable process.

SID's Dependability Attributes

Dependability as a system property reflects a user's degree of trust in the system and covers the related attributes of availability, reliability, safety, security, and resilience [2]. This report will provide a brief definition of each related attribute, and describe its importance to SID's design.

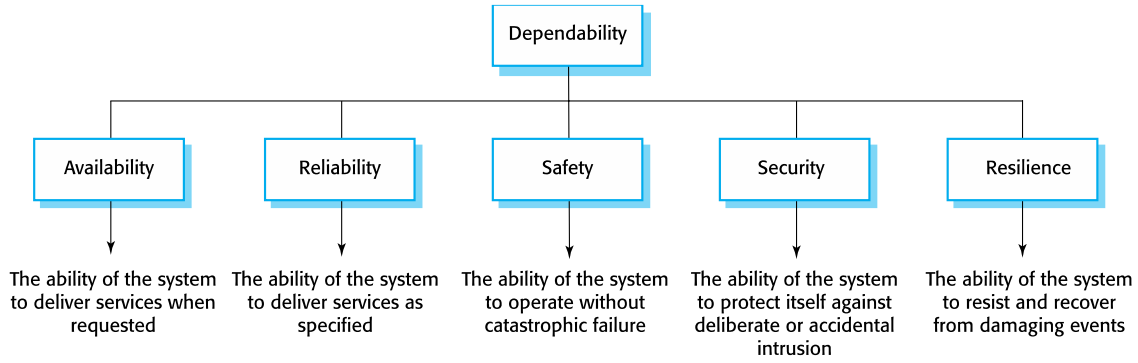

Figure 1: Principal Dependability Properties
Source: Adapted from [2]

Availability

Availability can be defined as a metric that measures the probability that a system is operational and accessible and consist of three requirements which are functioning equipment, functioning when needed, and functioning normally [3]. To remain up to date on the dynamic nature of financial data, SID would need to provide web services to the client to be useful. In order to provide web services, the design of SID will need to incorporate the use of a third-party web server provider to help minimize our start up equipment costs. In using a third party, we also gain the benefit of outsourcing the demand of availability management to a third-party, which allows us to prioritize on other development concerns.

DEPENDABLE PROCESS CONT'D

Reliability

Reliability of a system can be viewed as a probability metric that a system delivers a correct result from a provided service [2], or that a system will execute for a period of time without a failure [4]. Concerning SID, the correct output of services will be critical to the dependability for users given the nature of financial related services provided. Including a disclosure that SID is for information purposes only and not considered a source of advice will help manage user expectations; however, SID must operate with sufficient reliability in order to be embraced by users over time.

Safety

Safety relates to the potential for harm to the user through operation of the system [2]. As SID involves the user's investments, this is of critical concern. To prevent the ability for SID to do harm the user, we will limit its scope to advisory and informational services rather than services which actually manipulate a user's financial accounts. In doing so, along with a disclosure about SID being for informational purposes only, the risk of SID performing an action that harms the user's financial position will be reduced.

Resilience

Resilience in a system relates to its ability to recover from a successful attack against a critical component within a specified and acceptable period of time [5]. Our assumption is that the personal and valuable nature of SID's user data will make for an attractive target to outside attackers. Consequently, SID's resilience will be another key area of consideration within SID's security engineering. To properly describe SID's resilience engineering, we will provide a more thorough description in a later section of this report.

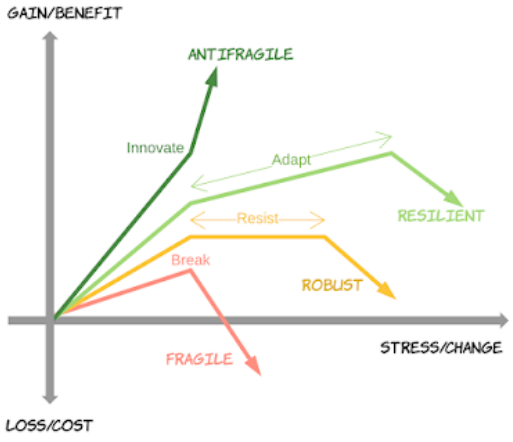

Figure 2: Resilience – Fragile to Antifragile
Source: Adapted from: [6]

SID'S SECURITY ENGINEERING

Security

Security of a system pertains to a system's ability to resist intrusion's from external attackers [2], or the ability to continue functionally correctly in spite of an outside attack [7]. Given the private nature of the user's financial information being handled by SID, this attribute will be one of the most critical to consider as a measure of SID's dependability. Because of this, a more detailed analysis of SID's security concerns can be found in the following sections of the report.

Importance of Security Engineering

Given the valuable nature of our users' financial data to attackers, it will be business critical that SID be developed as a secure system. When consider a system's security, there are three key dimensions to consider which are confidentiality, integrity, and accessibility [8]. For the sake of brevity, the nature of threats to a system will be discussed in the resiliency engineering section.

As a secure system, SID will need to be designed in such a way as to resist malicious attacks [2]. In order to conform with best practices in security design, security will remain as a key design consideration throughout SID's development. This is due to the increased challenge presented in securing an insecure system after it has been designed and implemented [2].

In order to secure SID, security engineering must be incorporated into our design approach as it relates to the tools, techniques, and methods that support the development of systems that can resist malicious attacks [2]. Security engineering is also a process which involves understanding the system architecture, identifying security requirements, and implementing security features and controls [9]. To fully understand the possible security requirements requires assessing threats at each system layer of the software system. These layers range from the system's hardware all the way up to the application itself [2].

Given our limited start up budget, our approach will seek to delegate the responsibility for risk mitigation in as many layers as possible. Our goal is to structure SID to provide services through a web portal and in doing so, we will gain the ability to outsource our server and storage needs to a third-party and larger outfit. Through outsourcing, our upfront hardware costs can be reduced, and we can also leverage the third-party's security expertise and practices to improve SID's ability to resist malicious attacks.

SID'S SECURITY CONT'D

Webservice Security Advantages

By structuring SID as primarily a webservice, we can better address the availability dimension of security. At the system layers below the application layer, we will rely on the outsourced third party to ensure high availability; however, these layers will still be tracked in house using metrics to ensure sufficient value is being received from the provider. For the application layer, the use of online personal finance software is growing in popularity and provides better data safety to the user by providing better firewalls and encryption of their data by storing it online rather than on their personal devices [10].

Additionally, SID can be updated automatically with each new build which can mitigate the threat of attackers exploiting older vulnerabilities on user's who ignore the need to update regularly. Through a webservice, we can also better control SID's security policies in areas such as user authentication. For example, to ensure user passwords are of sufficient strength and requiring periodic update, which further improves SID's ability to resist attacks that would seek to steal or corrupt user data. Using a webservice can mitigate the system's demands on a user's device allowing users to access the service from more device types which improves SID's accessibility.

Improving Application Layer Security

Through improvement of our security at SID's application system layer, we can greatly improve SID's ability to resist attacks by following best practices in secure software design. To realize these best practices, designing SID will also include the consideration and identification of security requirements alongside any functional requirements elicitation from the clients [11].

It will also be important to create misuse cases for any potential malicious attacks [2], so that security testing can be built into our test suites to improve SID's security. This type of testing includes penetration testing, experience-based testing, and tool-based analysis [2]. To allow for tool-based analysis, it is our intention to explore available security frameworks as this could help identify possible vulnerabilities earlier in SID's development [11].

This will be business critical, because it can identify vulnerabilities prior to delivery and preventing attackers from exploiting them. By combining the ability to automatically update SID and in removing vulnerabilities before delivery, there is reduced likelihood of SID experiencing a scenario such as a Zero-day attack [12].

These strategies will help SID to resist external attack; however, in our next section, we will also discuss how SID can recover as inevitably these steps will prove insufficient to preventing all attacks.

SID'S RESILIENCE ENGINEERING

Resilience is a judgement of how well a system can maintain the continuity of its critical services in the presence of disruptive events [2]. Before designing the ways in which SID would be able to recover from disruptive events, we must identify the potential threats that we may come across.

Cybersecurity Threats

Cybercrime refers to any illegal activity carried out using technology. Cybercriminals can target individuals, businesses, and governments. Therefore, businesses like ours can suffer from sensitive data loss, huge financial burdens, and brand damage. The average ransomware attack against small and medium businesses in 2019 demanded $5,900 to unblock their files or systems. Far worse, the downtime during these attacks cost the affected businesses $141,000 on average [13]. Being a start-up company consisting of two individuals only, cyber fraudsters can target our business. So, the first step towards security would be identifying cybersecurity threats.

Threat to the confidentiality of assets

In this type of threat, data is not damaged, but it is made available to people who should not have access to it [2]. In the context of SID, there would be a threat to confidentiality if fraudsters are able to access and view the data of the Dashboard.

Threat to the integrity of assets

These are threats where systems or data are damaged in some way by a cyberattack [2]. If the hacker gains access to SID and fabricates counterfeit information, or modifies the existing content, then this would pose a threat to the data integrity of the dashboard.

Threat to the availability of assets

These are threats that aim to deny use of assets by authorized users [2]. Denial of Service (DOS) attacks are aimed to prevent legitimate users from accessing the system. DOS attacks accomplish this by flooding the target with traffic or sending information that triggers a crash. In both instances, DOS attack deprives legitimate users of the services or resources they expected [14].

Controls

In order to take control of the threats, we can adopt the following strategies:

Authentication

In authentication, users of a system have to show that they are authorized to access the system [2]. In order to access SID, the clients will have to enter the username given to them by the institution and the correct password in order to get authenticated as legitimate users. The entered username and password must match with the one stored in the database server. If the users fail to do so, the server will not authenticate them, thereby ensuring that authentication is maintained.

SID'S RESILIENCY CONT'D

Cryptography/Encryption

In cryptography, data is algorithmically scrambled so that an unauthorized reader cannot access the information [2]. Storing the username and password in the database simply as plain text is not a safe practice. This is because if the hacker gains access to the database, he/she would be able to access all accounts and manipulate the data. To avoid this, we would use cryptography to encrypt the username and passwords. Now, even if the hacker gains access to the database, he/she will have a hard time figuring out passwords using Brute Force by trying out all possible combinations.

Firewalls

Firewalls are responsible for examining the incoming network packets. Then, they accept or reject an incoming packet according to a set of rules set by the company [2]. We would use firewalls to ensure that only traffic from trusted sources is passed from external Internet into the local network of the company.

Redundancy

We shall maintain copies of data displayed by SID. In case if there's an attack, we would have backup copies, at least. Maintaining backup copies would support recovery and reinstatement after a successful cyberattack [2].

Diversity

The authentication practice can be broken down into multiple stages. Adopting diverse authentication practices can be useful in protection against password attacks [2].

Resilience Engineering Activities

We assume that it is impossible to avoid system failures. So, we are concerned with limiting the costs of these failures and recovering from them. Also, we would use good reliability engineering practices to make SID as dependable as possible by minimizing the number of technical faults in the Dashboard. Therefore, our main emphasis lays on minimizing the number of system failures that can arise from external events such as operator or human errors or cyberattacks [2].

1) Recognition

The system as well as the clients using the Dashboard (System Operators) must recognize early indications of system failure [2]. One of the potential threats to our Dashboard would be a DOS attack. It is highly likely for this attack to occur. Some of the early signs of system failure due to DOS attack include the unusual slowing down of the Dashboard.

SID'S RESILIENCY CONT'D

2) Resistance

If the symptoms of a problem or cyberattack are detected early, then resistance strategies may be used to reduce the probability that the system will fail [2]. Some of the methods that we would use to prevent DOS attacks would be by installing a firewall and restricting and filtering the incoming traffic to the network. In addition to this, we would use encryption techniques to secure employee records as well as their contact information.

3) Recovery

If a failure occurs, the recovery activity ensures that critical system services are restored quickly so that system users are not badly affected by failure [2]. In this stage, the user can possibly kill the currently running process and restart the system. The most critical section is the core data displayed by SID. So, when a cyberattack takes place, our primary goal would be to secure the data so that the attackers are unable to modify it. Yet another thing that we would safeguard is the employee information. For safeguarding all of this, we would use resistance strategy to isolate the core data displayed by SID so that it is unaffected by problems elsewhere. Also, we shall store employee information in a different server so that the hackers are unable to access it

4) Reinstatement

In this final activity, all the system services are restored, and normal system operation can continue [2]. For this, we shall make use of redundancy and diversity. In addition to this, we will make sure that SID is cut off from all the networks and online services so that the employees can continue working in the offline mode. Meanwhile, we can change the encryption key and apply that to the data to create a new encryption code using automated systems to secure the data.

CONCLUSION

In conclusion, SID will not be engineered as a dependable product as our organization lacks the resources and practices to meet the requirements needed to produce a dependable process. Due to these limitations, our organization is not in the business of producing dependable products; however, we will provide SID with a measure of dependability by addressing user availability, reliability, safety, security, and resilience needs. In doing so, SID will be able to instill a sufficient degree of trust in our user's to be still be useful.

For SID's security engineering, we will outsource the risk for many layers of the system in order to minimize our start up equipment costs and allow us to focus on securing the application layer of the system. To secure the application layer, we will utilize the tools and techniques which help to improve SID's ability to resist external attack. These include a focus on security requirements, a test suite specific to security concerns, and the use of frameworks to help identify vulnerabilities before delivery. SID will also be designed primarily to provide services as a webservice which will allow other vulnerabilities to be mitigated and provide better accessibility to users.

Concerning SID's resiliency engineering, a thorough review of potential threats to the system's assets is to be completed to identify likely event from which the services must recover. The three types of threats relate to the confidentiality, integrity, and accessibility of the system and those system attributes that attackers wish to compromise. SID's resilience will focus on strategies to recover from an event wherein an attacker has been successful in damaging one or more of these attributes. The activities of resilience engineering capabilities that SID will include recognizing an attack, resisting it where possible, recovering critical services quickly in the event of attacks, or to reinstate the system as a worst-case scenario. To allow for recovery, we will employ the use of diversity and redundancy in SID's design approach.

REFERENCES

[1] F. J. Pino, F. Ruiz, F. Garcia and M. Piattini, "A software maintenance methodology for small organizations: Agile_MANTEMA," *JOURNAL OF SOFTWARE MAINTENANCE AND EVOLUTION: RESEARCH AND PRACTICE,* pp. 851-876, 2012.

[2] I. Sommerville, Software Engineering (10th Edition), Pearson, 2015.

[3] Fiix Software, "System Availability," Fiix Software, 2020. [Online]. Available: https://www.fiixsoftware.com/how-do-maintainability-and-reliability-affect-availability/. [Accessed 20 February 2020].

[4] D. F. Rico, H. H. Sayani and R. F. Field, "Software Reliability," *Advances in Computers,* 2008.

[5] W. Axelrod, "Investing in Software Resiliency," Researchgate GmbH, 01 May 2017. [Online]. Available: https://www.researchgate.net/publication/293515438_Investing_in_software_resiliency. [Accessed 20 February 2020].

[6] Buildium Life, "3 key patterns of software resilience," Medium.com, 3 September 2019. [Online]. Available: https://medium.com/@buildiumlife/3-key-patterns-of-software-resilience-ae01f191a29e. [Accessed 26 February 2020].

[7] G. McGraw, "Software security," *IEEE Security & Privacy,* vol. 2, no. 2, pp. 80-83, 2004.

[8] I. Golovatenko, "The Three Dimensions of the Cybersecurity Cube," Swan Software Solutions, 13 December 2018. [Online]. Available: https://swansoftwaresolutions.com/the-three-dimensions-of-the-cybersecurity-cube/. [Accessed 26 February 2020].

[9] IBM Support Knowledge Center, "Security engineering steps," International Business Machines, 2020. [Online]. Available: https://www.ibm.com/support/knowledgecenter/SSYLSL_10.0.0/com.ibm.help.secure.deploy.doc/security/FND_SecurityEngSteps.html. [Accessed 26 February 2020].

[10] S. Elmblad, "Safety First: Online vs Desktop Personal Finance Software," Dotdash Publishing, 20 November 2019. [Online]. Available: https://www.thebalance.com/safety-first-online-vs-desktop-personal-finance-software-1293915. [Accessed 27 February 2020].

[11] E. Mougoue, "Secure SDLC 101," Synopsis, Inc., 21 January 2016. [Online]. Available: https://www.synopsys.com/blogs/software-security/secure-sdlc/. [Accessed 27 February 2020].

[12] Panda Mediacenter, "How to avoid zero-day attacks," Panda Security, 2019. [Online]. Available: https://www.pandasecurity.com/mediacenter/tips/how-to-avoid-zero-day-attacks/. [Accessed 27 February 2020].

[13] N. Latto, "What is Cybercrime and How Can You Prevent It?," Avast Software s.r.o, 20 February 2020. [Online]. Available: https://www.avast.com/c-cybercrime#topic-4. [Accessed 26 February 2020].

[14] Cyberpedia, "What is a denial of service attack (DoS) ?," Palo Alto Networks Inc., 2020. [Online]. Available: https://www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos. [Accessed 26 February 2020].