Firehawk Consulting

The following report was prepared on behalf of SwiftTech.

Thank you for giving Firehawk Consulting the opportunity to review your security posture in anticipation of performing a SOC II security assessment.

We hope you find the notes below as you begin your journey. Please do not hesitate to contact us if you have further questions.

For

SwiftTech

Firehawk Consulting

After review, Firehawk has noted the following areas of concern. You may wish to consider updating policy and security controls based on your current business goals, risk management posture, and compliance considerations.

Controls

Data Storage

- VPC3 File storage supports only AES-128 encryption
- Databases in production environment are unencrypted

End User Management

- Internal Network users require a 7-character password
- Passwords never expire
- VPN access does not require MFA

Network Controls

- TLS v1.1 is used between the cloud production environment and SwiftTech's physical location
- Application development Tiers are not logically segmented from Business Application servers

Patching and Vulnerability Management

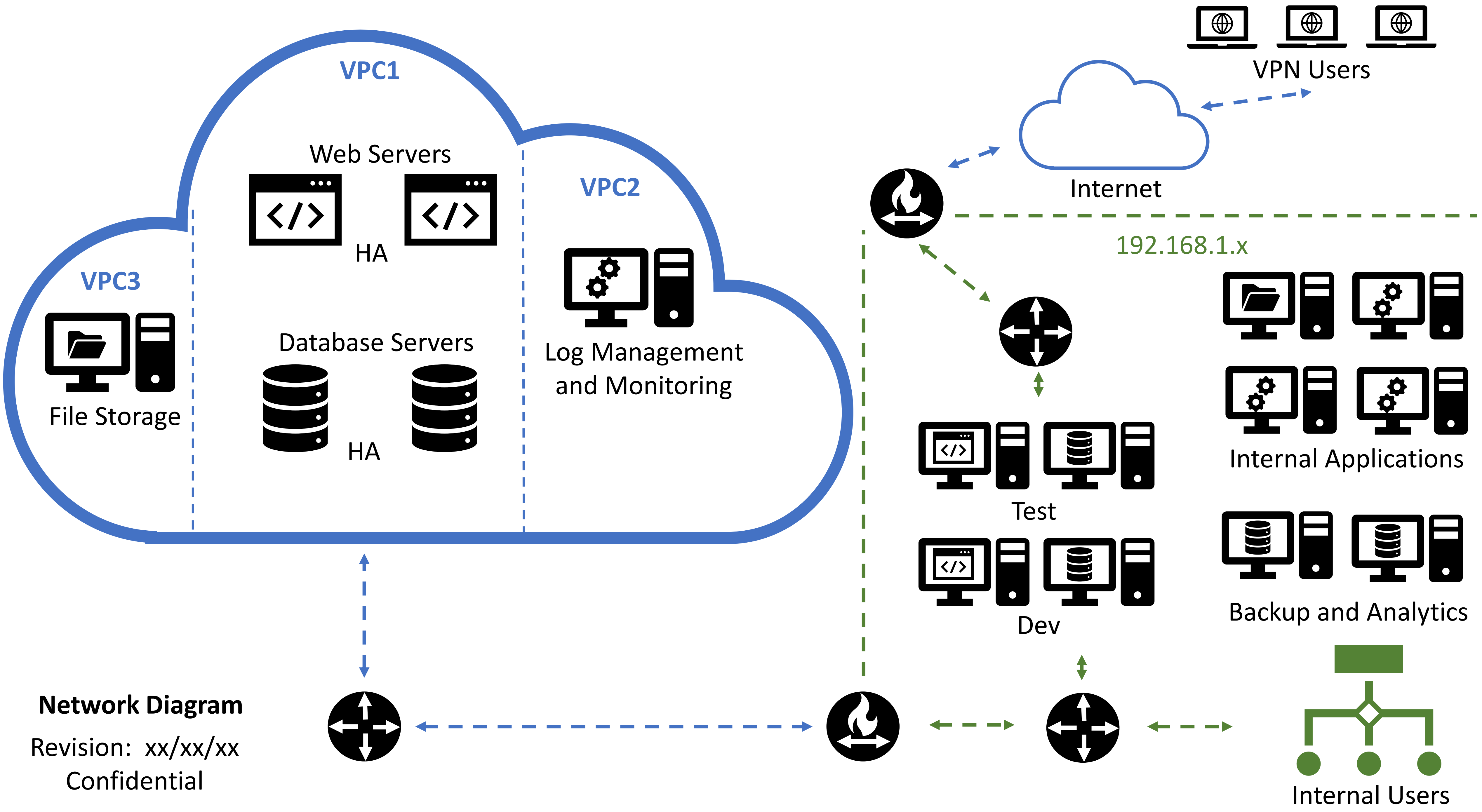
- Development Tier servers are unpatched and contain multiple vulnerabilities

Secure Software Development

- Application code is not scanned for vulnerabilities before being published into production environment

SwiftTech

Speed, Flexibility, Success



Network Diagram

Revision: xx/xx/xx

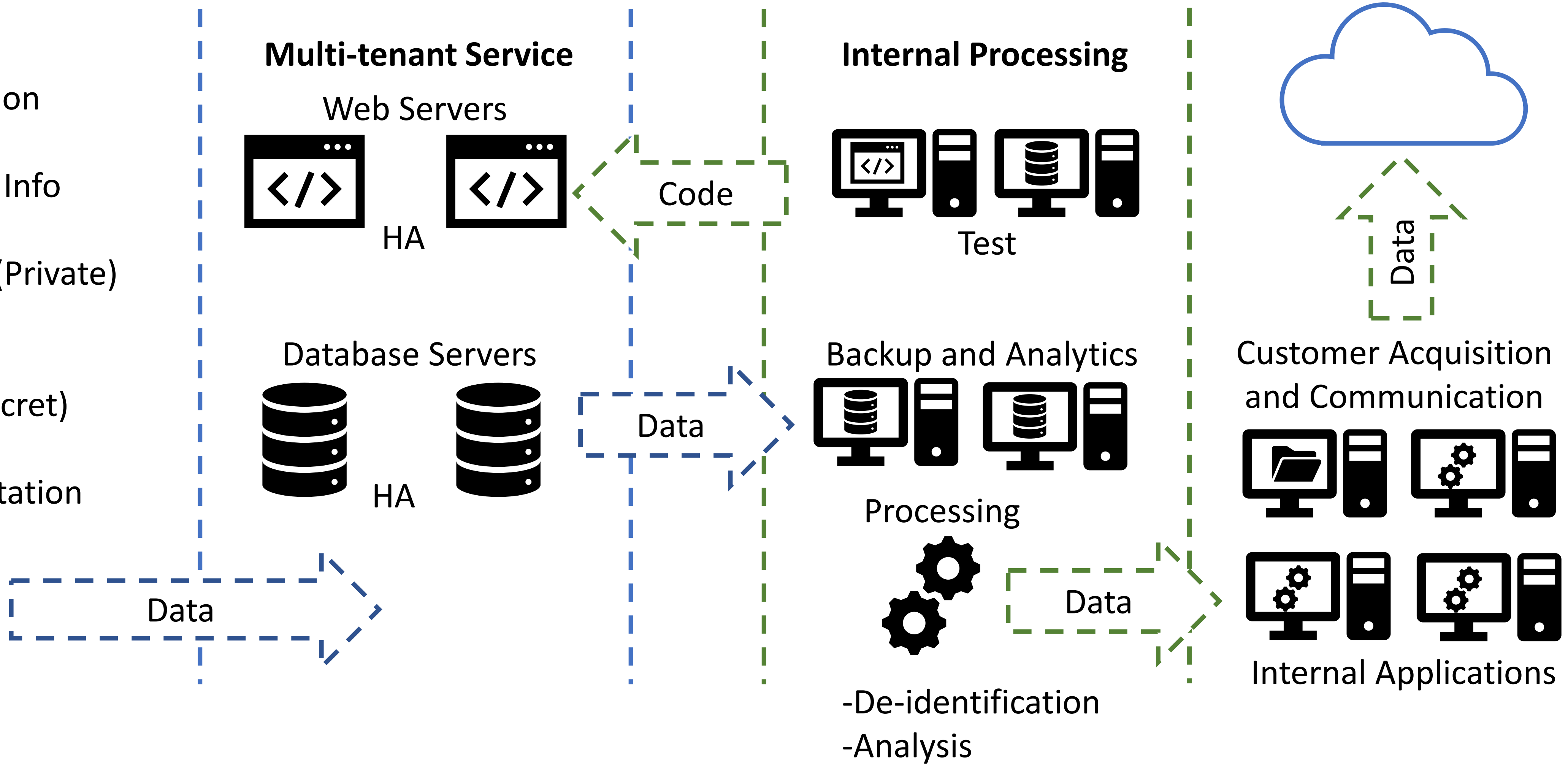
Confidential

Inputs

Company Registration
- Company Name
- Company Contact Info

User Registration
- User Information (Private)
- Role Assignment

Data Input
- Project Details (Secret)
- Project Timelines
- Related Documentation



Multi-tenant Service

Web Servers

HA

Code

Internal Processing

Test

Database Servers

HA

Data

Backup and Analytics

Processing

Data

-De-identification
-Analysis

Customer Acquisition and Communication

Data

Internal Applications

Data

Data Flow Diagram

Revision: xx/xx/xx
Confidential

Security Posture (1.)

As the newly hired cyber security GRC, I had the opportunity to review FireHawk Consulting's report along with the SwiftTech's email. It appears that SwiftTech would like to expand their company into Software as a Service (SaaS). However, in order to please prospective customers they need to comply with governance, risk, and compliance programs and hence, the reason for hiring FireHawk Secuity to preform a readinesss assessment in preparation for pursing a SOCII attestation report. As per their statement, SwiftTech believes in developing new ideas as quickly as possible and the fact they do not to sacrifice their commitment to agile software development, therefore, they are Risk Accepting.

Relevant Frameworks (2.)

When reviewing the Greater Minnesota Lifecare's Information Security Addendum and SwiftTech's prospective customers for the ProjectTrackPlus, I suggest the two regulatory frameworks that we should use to measure out existing security controls and incorporate into our risk management framework:

- **The Secure Software Development Framework (NIST SP 800-218).** The reason is:
  - SwiftTech wants to provide Software as a Service (SaaS) with their ProjectTrackPlus.
  - In order to have prospect client, Greater Minnesota Lifecare, SwiftTech needs to assure that all application code is free of security flaws.
- **Health Insurance Portability and Accountability Act Security Rule (HIPAA) (NIST SP 800-66) and NIST SP 800-63B**
  - SwiftTech will need to follow HIPAA security rules if they are to work with Greater Minnesota Lifecare given this prospective client deals with electronic personal healthcare information(ePHI).

- **Security Guidelines for Storage Infrastructure (NIST Special Publication 800-29) and Storage Security: Data Protection (SNIA Technical White Paper 2018)**
  - SwiftTech will need to store all of Greater Minnesota Lifecare information using a data encryption, preferably, Advance Encryption Standard (AES) 256 or above.

Audit Against Frameworks (3.)

SwiftTech

- Application code is not scanned for vulnerabilities before being published into production environment. Also, Development Tier servers are unpatched and contain multiple vulnerabilities. These area valid concerns. The earlier in the software development life cycle (SDLC) that security is addressed, the less effort and cost is required to achieve the same level of security. As per **NIST SP 800-218**
    - **Review and/or Analyze Human-Readable code to identify vulnerabilities and verify compliance with security requirements**.
        - ❑ Code analysis tools, which are tools used to find issues in code, in fully automated way or in conjunction with a person, should be used.
    - **Test executable code to identify vulnerabilities and verify compliance with security requirements**
        - ❑ Scope the testing, design the tests, perform the testing, and document the results, including recording and triaging all discovered issues and recommended remediation in the development team's workflow or issue tracking system.
    - ▪ **Assess, Prioritize, and Remediate Vulnerabilities.**
        - ❑ Analyze each vulnerability to gather sufficient information about risk to plant its remediation or other risk response.
        - ❑ Perform risk calculations for each vulnerability based on estimates of its exploitability, the potential impact if exploited, and any relevant characteristics. This should satisfy Greater Minnesota Lifecare's information security addendum on that all application code is tested and free of security flaws that would create risk greater than a rating of "low"

Audit Against Frameworks (3.) pg2

SwiftTech

- The Network Diagram indicates that the testing and deveops are not in a separate location. In addition, Application development Tiers are not logically segmented from Business Application servers and this is a valid concern.

Therefore, SwiftTech should **Implement and Maintain Secure Environments for Software Development.** SwiftTech will need to:

- to separate and protect each environment involved in separate development
  - ❑ Use multi-factor, risk-based authentication and conditional access for each environment
  - ❑ Use network segmentation and access controls to separate the environments from each other within each non-production environment, in order to reduce attack surfaces and attackers' lateral movement and privilege/access escalation.
  - ❑ Encore authentication and tightly restrict connections entering and exiting each software development environment, including minimizing access to the internet to only what is necessary.
  - ❑ Continuously log and monitor operations and alerts across all components of the development environment to detect, respond, and recover.
- Internal Network users require a 7-character password. This is a valid concern. As per HIPAA Security Rule and NIST SP 800-63B. **Implement procedures to verify that a person or entity seeking access to electronic protected health information is the one claimed a**
  - ❑ Evaluate Authenication Options Available
    - ▪ A combination of two or more of the following options should be considered: token, piv card, password, fingerprint.

Audit Against Frameworks (3.) pg 3

  - ❑ Memorized secrets which is a password or pin SHALL be at least 8 characters in length if chosen by the subscriber.

- File store is only using AES-128 encryption. This is a valid concern to the prospective client, Greater Minnesota Lifecycle. As per NIST, AES-128 is one of the secured encryption key ciphers. However, AES-256 encryption is the most secured:
  - ❑ More rounds. The more rounds, the more complex the encryption, thus, making AES-256 the most secured compared to AES-128

Governance Mechanisms for End-User Management Controls (6.)

SwiftTech

1. Develop a password and password audit policy.
2. Configure the minimum password length policy setting to a value of 8 or more.
3. Configure that passwords are change every 90 days.
4. Audits will be conducted every 90 days using a password-cracking software to determine compliance and to track any passwords that are too weak or compromised.