Remote Access

All access to [company name] system components initiated outside the organization's trusted network infrastructure is to be considered "remote access", and as such, only approved protocols are to be used for ensuring that a trusted connection is initiated, established and maintained. Specifically, all users are to utilize approved technologies, such as IPSec and/or SSL Virtual Private Networks (VPN) for remote access, along with additional supporting measures, such as Secure Shell (SSH), while also employing two-factor authentication. The concept of two-factor authentication (i.e., something you know, something you have, something you are) along with strong password policies creates yet another layer of security relating to access rights for all authorized users granted remote access into [company name's] network.

Additionally, all workstations (both company and employee-owned) are to have current, up-to-date anti-virus software installed, while also utilizing any other malware utilities as needed for protecting the workstations and the information traversing to and from the remote access connection. This may also include the use of personal firewall software, along with enhanced operating system settings on the applicable workstations.

[Please also discuss what specific remote access procedures your organization has in place - specifically – what tools and protocols are used, etc.].

Wireless Security

Initially implementing a WLAN requires adherence to the following stated guidelines for ensuring the safety and security of the wireless platform itself, along with ensuring the confidentiality, integrity, and availability (CIA) of [company name]'s overall information systems landscape:

- **Secure Deployment:** All WLAN devices and supporting resources, such as wireless access points, and other network devices, are to be positioned in a manner for ensuring unauthorized physical access and modification. Additionally, they are to be secured with approved fixtures and other necessary apparatuses for mitigating any unnecessary movement. Additionally, the WLAN platform itself is to be logically | physically segregated from the corporate | internal wired network, which can be achieved by utilizing firewalls and other access control methods.

- **Asset Inventory:** Once all WLAN devices are safely secured, a complete asset inventory is to be taken, documenting all necessary information, such as physical location, and corresponding unique identifiers (i.e., hostnames, serial numbers, etc.).

- **Configuration of Wireless Access Points:** The following measures are to be undertaken regarding WLAN platforms:

    - Change default administrator settings, such as username and password, along with implementing strong, unique administrative passwords (i.e., alphanumeric, case sensitive, etc.) for all wireless access points.
    - Change any default IP addresses also.
    - Configure SNMP and NTP accordingly.

- Configure wireless modes to support only the one (1) primary - and industry approved - wireless networking standard.
- Change vendor default settings for Service Set Identifier (SSID) to a completely new network name, but also one that does not openly identify or provide any critical [company] name information. Specifically, the SSID character string is not to reflect company name.
- Use a "closed network" concept, whereby the SSID is actually not broadcasted (if allowable), rather, it must be entered into the client application.
- If the SSID must be broadcasted, create a healthy balance of allowing all authorized users to receive such signals, but not the point where unauthorized parties can potentially view such information.
- Remove all unnecessary and insecure services and protocols from all WLAN devices, such as the wireless access points and any all other associated wired network devices.
- For all remaining services and protocols, implement the concept of "least privileges".
- Implement MAC Address filtering and wireless access points.
- Use the strongest encryption algorithm currently available (WPA2), and use other forms of encryption as needed, such as VPN, SSL | TLS, etc.
- Protect all sensitive wireless access points information, such as administrator passwords, SSID password, keys, etc. with approved security measures, such as encryption itself.
- Enable logging features and ensure that all logs and audit trails are sent to a remote logging server and retained as necessary (i.e., regulatory compliance laws, etc.). Information captured should include, but not limited to, the following: source\destination IP addresses, MAC addresses, user logon information (i.e., time, username, etc.), user logoff information
- Enable usage parameters, such as time-out sessions.
- Disable wireless access points during non-business hours, such as nights, weekends, holidays, etc.

[Please also discuss any other wireless security initiatives you feel are relevant].

Malware

Malicious software (malware) poses a critical security threat to [company name] system components, thus effective measures are to be in place for ensuring protection against viruses, worms, spyware, adware, rootkits, trojan horses, and many other forms of harmful code and scripts. As such, [company name] is to have anti-virus (AV) solutions deployed on all applicable system components, with the respective AV being the most current version available from the vendor, enabled for automatic updates and configured for conducting periodic scans as necessary. Because strong and comprehensive malware measures are not just limited to the use of AV, additional tools are to be employed as necessary for eliminating all other associated threats, such as those discussed above. The seriousness of malware and its growing frequency of attacks within organizations require that all I.T. personnel within [company name] stay abreast of useful tools and programs that are beneficial in combating harmful code and scripts. Common examples of malware include the following:

- **Computer Virus:** A computer program that has the ability to replicate itself and spread from one computer to another. Common viruses include, but are limited to, the following: polymorphic virus, boot virus, macro virus, multipartite virus, web scripting virus, etc.
- **Malware:** software created and/or used for the purposes of harming and damaging various systems, such as computer code, files, applications, and other relevant information technology platforms and utilities.
- **Antivirus:** Software used for purposes of preventing, detecting, and removing malicious software (i.e., malware).
- **Worms:** A standalone, independent program that has the ability to replicate itself and spread to other computers, ultimately infiltrating programs and destroying data.
- **Trojan Horse:** A harmful piece of malware that facilitates unauthorized access on a computer system by way of social engineering tactics and strategies.
- **Key loggers:** Unauthorized capturing of a user's keystrokes on a computer system. Note: It is considered malware when it is "unauthorized" as there are legitimate uses of key logging software.
- **Rootkits:** Software that enables unauthorized access to a computer system and that is also hidden from detection. Rootkits can conceal the altering of files, data, etc. and are a serious form of malware.
- **Spyware:** Software that collects vital information from a computer system regarding data on such system and the associated user activities. Note: It is considered malware when it is "unauthorized" as there are legitimate uses of spyware.
- **Adware:** Programs that facilitate delivery of advertising content and related material to a user through their browser while on the Internet, or through some other type of interface. Note: It is considered malware when it is "unauthorized" as there are legitimate uses of adware.
- **Logic Bomb:** Code that is intentionally inserted into a software system that initiates a malicious function when specified conditions are met.

Change Control | Change Management

Changes made to [company name] system components require authorized users to initiate an incident and/or change request, which includes completing all applicable forms as necessary. Furthermore, the request must be thoroughly documented, which includes providing the following essential information: (1). An assigned I.D. or change tracking number. (2). Representation of all critical dates relating to the requested change itself, such as when the change was originally submitted and approved, as well as when it was migrated to various stages for testing and final deployment to production, if applicable. (3). Default fields for categorizing (i.e., normal change or emergency change, etc.) and prioritizing (i.e., critical to routine maintenance) the requested change itself. (4). Documented notation, communication and correspondence throughout the life of the requested change itself is to include, but is not limited to, the following: (a). Documentation of impact. (b). Management signoff. (c). Operational functionality. (d). Back-out procedures.

Additionally, change control measures include changes undertaken for any of the following four (4) environments for which system components reside in:

- Change Control | Internally Developed Systems and Applications
- Changes Control | Enterprise Wide
- Change Control | Customer Facing Environments

- Emergency Change Management | All Environments

[Please also discuss what specific change control procedures your organization has in place - specifically – what tools, software, forms are used, etc.].

Software Development Life Cycle (SDLC)

The Software Development Life Cycle (SDLC) for [company name] is to encompasses a number of phases, each concluding with a major milestone. Assessments are conducted after each phase to determine if objectives have been satisfied. Skilled software engineers are to be utilized throughout all phases, which results in a thorough and uninterrupted process from beginning to end. Specifically, SDLC activities for internally-developed systems/applications consist of the following procedures and phases:

- **New System/Application and Feature Development.** New system/application and feature development is the implementation of a new service or addition of new features and functions to the current product. The same processes are also involved when adding major enhancements to existing functionality.

- **Request for New System/Application or Features.** The process begins with the request for a new system/application, feature or tool. Authorized personnel will initiate the request. All requests are to be appropriately logged in [ticketing system and/or some other type of tool].

- **Feasibility Study.** Once a request for a new system/application, feature or tool is received, [company name] analyzes it and evaluates its market opportunity and/or operational impact. Once the benefits are identified, [company name] conducts a feasibility study with the assistance of the development team. Based on the requirements, if the feature requested can be done in a reasonable fashion, a work estimate to implement the new system, application, feature or tool is prepared. For complete new systems and applications, [company name] estimates the market size and develops a business case.

- **Estimate and HW/SW Requirements.** Along with estimating the effort and time required to implement the new system/application, feature or tool, an estimate of hardware and software required for development and final deployment is conducted. These estimates are passed on to management for final approval.

- **Management Decision.** After reviewing the business rationale for the new system/application, feature or tool, [company name] decides whether the cost/benefits and strategic direction warrant the development to proceed. A review of the business rationale for a completely new project includes studying market opportunity and conducting a competitive analysis. [Company name] can opt not to proceed with the development or even to table it for a period of time. As soon as the project receives approval, the process progresses to the development and deployment phases.

- **Requirement Analysis.** During this phase, a detailed requirements analysis of the new system/application, feature or tool is conducted and documented in the form of a requirements

specification. Documents and activities for this phase include obtaining copies of documents used during this phase and interviewing personnel for major activities during this phase.

- **Design.** In this phase, various technical personnel collaborate to develop a detailed design of the various activities involved. The design and development team reviews the design, and the final version is documented in the form of a design specifications document. If the feature or tool is to be a part of an existing system/application or functionality, the existing design document may be modified in lieu of creating a new document. Test plans and procedures for system tests are also developed.

- **Implementation.** Once the design is finalized, the actual implementation of the system/application, feature or tool begins with a test in a development environment. After all errors found during the testing stage are corrected, the application code is released to a test server.

- **Quality Assurance and Testing.** Once all the modules are moved to a test server and integrated in the test environment, any necessary test database tables and stored procedures are also created on the test server(s). The test environment is configured as a replica of the production environment or a specific client environment; however, there may be external interfaces which, at times, may not be duplicated, and approximations may be used. Testers then assess the new modules in this test environment. Test cases and scripts are written and documented as required. Any discrepancies are resolved with the development team, and any other additional testing is conducted. Customers and/or third-party users may be involved at different levels in this phase of project cycle, based on a mutual understanding of verification requirements. Test results are documented and reviewed with development personnel and management for final approval.

- **Release for Production.** Once the system/application, feature or tool is successful in the test environment, [company name] approves the release for production. Modules are moved to the production servers where functionality is tested after all modules are updated.

[Listed above are generally accepted industry-standard phases within a software development life cycle (SDLC). Please modify the above information to reflect your organization's specific life cycle.]