Report on
Baseline Requirement Assessment

Prepared by

Bhutan Computer Incident Response Team,
Department of IT & Telecom,
Ministry of Information & Communications

Date: 26/07/2017

The baseline is aimed to deliver assessment for three categories of applications - Basic, Medium and High. The principle in defining applications could be based on the following:

1. Basic: Static or semi-static standalone (without integration) system or application, representing not sensitive information, e.g. official webpage. Might be basic authentication mechanism for users. no transactions.
2. Meidum: System or application which represents contextualized information or data depending on user's profile, access rights and roles. Usually deals with sensitive (not public) information, transactional.
3. High: System or application which is essential for the maintenance of vital societal functions, health, safety, security, economic or social well-being of people, proper functioning of the Government of Bhutan and the disruption or destruction of which would have a significant impact to maintain those functions

Function	Requirement	Basic	Medium	High
IDENTIFY	A business owner within organization must be identified for an application and approved in written (usually with responsibility to govern an asset, ensure that maintenance is in place, and manage related budgeting activities)	+	+	+
IDENTIFY	Organization shall ensure appropriate IT support for an application throughout its lifecycle. It includes system's patching, re-configuration, users' trouble shooting and related activities. Support could be ensured by internal or external resources (suppliers).	+	+	+
PROTECT	[GDC complies by default] System has to be protected from interruptions of power supply		+	+
PROTECT	[GDC complies by default] System has to be protected from interruptions of Internet link		+	+
PROTECT	[GDC complies by default] Major system's parameters has to be monitored in an automated way: CPU, RAM, disk usage, internet link utilization, including application's uptime. This information should be accessible to an organization		+	+
PROTECT	Users, including administrators, must log on using their personal user accounts to enforce accountability. Shared accounts must not be used unless approved in written	+	+	+
PROTECT	Default root access, guest and anonymous accounts, including default vendors' accounts must be disabled and renamed (depends on context) in an application and all related components (if any): middleware and content management frameworks, databases, network equipment, servers and operating systems. Passwords for all default accounts (including for root) should be changed into complex (longer than 14 characters with combination of numeric, characters, alphanumeric)	+	+	+
PROTECT	The number of personnel that can gain access to an application / system with administrator privileges (whether local or network-based) must be minimized in every application's component	+	+	+
PROTECT	Temporary (login suspension for particular time, reCaptcha) or permanent user lockout against brute force attacks should be implemented in an application's frontend		+	+
PROTECT	Users session timeout mechanism should be implemented	+	+	+
PROTECT	It is discouraged to store credentials in application's source code and other forms accessible through authenticated or anonymous web browsing.	+	+	+
PROTECT	Manually or automatically reseted user's password has to be every time different or generated in random sequence	+	+	+
PROTECT	Frontend application's administration access interface (e.g. /wp-admin in WordPress CMS) should be disabled or restricted from public access (e.g. IP white listening, VPN)		+	+

PROTECT	Disable remote root access (if not used) or remote root access should be restricted to users (or groups) on "need to have" basis only. It is highly recommended to provide access to systems with non-administrative account. There upon, create privilege escalation through adoption of Sudoer policy on Linux and "RunAs" on Windows environments.	+	+	+
PROTECT	Two-factor (like random code received by email/SMS, one time passwords (OTP)) or two-stage authentication (different credentials) must be used for remote root access established over public networks (e.g. VPN + RDP; VPN + SSH). Where multi-factor authentication is not supported, root accounts shall be required to use long passwords on the system (longer than 14 characters in combination of alphanumeric, numeric and characters).			+
PROTECT	Every application's user has to be uniquely identified in the system. This is to help maintain accountability.	+	+	+
PROTECT	[GDC complies by default] All applications shall be hosted in a dedicated zone (e.g. organization's DMZ zone)	+	+	+
PROTECT	By signing contracts, an organization shall ensure the right to audit a supplier in order to have mechanism to ensure supplier's compliance to contractual obligations.			+
PROTECT	Contract with developers / support organization must consider the protection of intellectual property of source code (custom built)	+	+	+
PROTECT	Sensitive application's data like users' passwords credential has to be encrypted while "data at rest"	+	+	+
PROTECT	Encryption (e.g. SSL/TLS) must be used to protect data in transit (including login activities) on both public-interfaced and organization-controlled networks, even for anonymous access. Insecure connections (e.g. HTTP) have to be forwarded into secure (e.g. HTTPS) by default.	+	+	+
PROTECT	Aapplication should implement a multitier architecture in order to ensure components of the application are securely separated. E.g. database should be kept in a separate server.			+
PROTECT	Every application's component, like middleware and content management frameworks, databases, network equipment, servers and operating system shall be hardened by using CIS. (https://www.cisecurity.org/cis-benchmarks/) or related well known security benchmarks relevant to application's context		+	+
PROTECT	Application shall display generic error messages to users only coming from every application's component. Error messages should not reveal details about the internal state of the application (e.g. framework version, file system path and stack information)(output sanitization)	+	+	+
PROTECT	Testing environment(s) has to be separated (semi-separated) from the production environment. All testing should be carried out in the staging room.		+	+
PROTECT	An accurate time source such as Network Time Protocol (NTP) based time must be maintained by all servers and applications.	+	+	+
PROTECT	Unencrypted communication protocols shall be disabled on server like TELNET, FTP .	+	+	+
PROTECT	Unnecessary operating system's services shall be disabled.	+	+	+
PROTECT	All network ports shall be blocked (denied by default), except those necessary to provide application service and support (e.g. remote secure connection).	+	+	+
PROTECT	An application must be configured to use a service account assigned with the least privileges necessary to run the application	+	+	+
PROTECT	Version control must be maintained for all application updates and changes.			+
PROTECT	[GDC complies by default] System images or application's backup that reflect the current state of the server or application (including configuration) should be retained.	+	+	+
PROTECT	System images or application's backup data must be tested regularly to ensure that application and relevant data can be recovered when required (once a year at least)		+	+

PROTECT	Detailed audit trail in every application's component must be configured to record the following events (when applicable): Privileged actions; Access to sensitive resources (e.g. a particular file/folder); Security events (successful and failed login attempts, clearing of audit logs, changes in membership of privileged/administrative user groups, system start up and shutdown, system time changes, system backup or restoration, changes to audit policy settings) All audit events must record: Date and time of the event; Event type identification / description; Subject identity (e.g. user identification); Success or failure of the event.	+	+	+
PROTECT	Audit trail collection and storage should be on a separate machine.			+
DETECT	[GDC complies by default] All software, including operating systems and third party applications, must be scanned against vulnerabilities periodically as per Vulnerability Management Process.			+
DETECT	All software, including operating systems and third party applications, must be periodically scanned against vulnerabilities and with frequency as per Vulnerability Management Process	+	+	
DETECT	Once in two years third party penetration testing (in "black box" mode) should be performed to ensure the ongoing effectiveness of application security controls as new threats emerge. Minimum scope for penetration testing is OWASP top 10 threats			+
DETECT	[GDC complies by default] Organization has to be able to detect cyber security threats and incidents on a network level by deploying reasonable for application's context solutions and internal processes		+	+
DETECT	Organization has to be able to detect cyber security threats and incidents on a machine and application level by deploying reasonable for application's context solutions and internal processes: Host Intrusion Detection System (HIDS), Security Information and Event Monitoring system (SIEM) and/or other relevant (commercial/open source) tools			+
DETECT	Organization has to be able to detect internal cyber threats related to unauthorized access or modification of sensitive data in an application. Internal cyber threats usually are related to privileged users/groups (e.g. database administrators), special users/groups (e.g. interconnected systems), and suppliers (e.g. developers)			+
RESPOND	All software, including operating systems and third party applications, must be protected from known Critical and High level vulnerabilities. It is achieved by systems patching activities	+	+	+