Securing Blackboard Learn on Linux

GIAC (GCUX) Gold Certification

Author: David C. Lyon, lyondc@gmail.com
Advisor: Dr. Kees Leune

Abstract
Blackboard Learn is one of a suite of products offered commercially by the Blackboard Corporation. Blackboard Learn is the most widely used Learning Management System to date. A Learning Management System provides online course management and optionally, community engagement and content management. Blackboard runs on open source technologies including Apache HTTP and Apache Tomcat both of which typically run on Linux. System Administrators tasked with hosting Blackboard Learn on Linux have no control over the bundling of the technologies and whether they are current. Because of this and the rapid growth of online learning, and specifically Blackboard, it is critically important to have a plan to secure Blackboard Learn. The Blackboard technical community offers some assistance; however, something more comprehensive and concise would be useful. This paper will examine the deficiencies in Blackboard, and how it and the services it uses can be made more secure when running on Linux. The aim is to provide a best practice that would benefit Blackboard administrators and others in similar scenarios.

1. Introduction

Blackboard Learn (Bb Learn) is an application suite providing educational technology to facilitate online, web based learning. It is typical to see Bb Learn hosting courses and content. Common add-ons include the Community and Content systems which are licensed separately.

Bb Learn is complex and layered. Security affecting each layer needs to be addressed. This includes shared content, database security, SSL utilization, authentication, administrative accounts and roles, and privileges. The software bundled with Bb Learn (Apache web server for example) may not be from the current version tree, presenting added risk and challenges of risk mitigation. Hazlewood (2006) states that, "A comprehensive Defense-In-Depth information assurance strategy should be employed to mitigate the threats and keep and organization's IT assets and proprietary information as secure as possible" (p. 4). Linux server hardening including basic firewall settings, configuration baseline, intrusion detection, log monitoring, Bb Learn hardening and log monitoring/maintenance, and virus detection need to be part of the approach.

While good security practices on Linux can provide a secure O/S platform in which to run Bb Learn, one must also consider Bb application security. O/S security alone does not protect the data behind the application. Discussing security as it relates to the business logic of Bb Learn on Linux is crucial. Learning Management Systems such as Bb Learn are evolving rapidly and require a good security best practice. Maintaining security for a web server hosting static pages is different than hosting one that hands off requests to a database. According to Sintes (2002), "As for the application server, according to our definition, an application server exposes business logic to client applications through various protocols, possibly including HTTP" (p. 1). Securing the business logic and data is especially important since the data behind Bb Learn can be considered sensitive – such as grades. Student records are protected by FERPA (Family Education Rights and Privacy Act) in the United States. FERPA defines personally identifiable information including a student's name, address, etc. (U.S. Privacy Definitions and Regulations

Relevant to Blackboard," 2011). Familiarity with similar regulations in other countries is important.

Having the database hacked is one of the consequences of not having a good security practice in place for Blackboard (Benedict, 2002). Benedict's case study shows the importance of having good security measures in place beyond the application layer. Policies and procedures, change control, backups, and recovery are an important part of the plan. In Bb Learn, one click can disable SSL causing everything to be exchanged in clear text. Monitoring for such a change and authorizing change though clear procedures backed by solid policy can avert unintended consequences. Having a baseline configuration is critical to the O/S as is keeping track of changes in key Bb Learn database tables.

The goal of this paper is to address what it means to secure the Bb Learn application running on a self-hosted Linux environment. Bb Learn does require a database and shared content if multiple application servers are deployed. Database and shared content server security is important but outside the scope of this paper. However, the practices presented are relevant for those servers. This paper discusses the need for a policy backed plan, Bb Learn risks, the services Bb Learn runs under, basic server, Apache, and Tomcat hardening steps, Bb Learn application countermeasures and steps, and a plan for maintenance and monitoring. Checklists for hardening are provided in the appendix.

Blackboard certifies that Bb Learn 9.1will run on Red Hat 5 ("Blackboardlearn Release 9.1 Service Pack 5 Release Notes," 2011). This paper's focus is on Bb Learn 9.1 and the software bundled with it, and Linux Red Hat version 5, the Oracle database, and content shared using Network File System (NFS). This paper assumes the reader should already be familiar with Linux, Tomcat, Apache, SQL queries, and Bb Learn. Tomcat clusters, virtual installations, and Single-Sign-on integration will not be covered. Detailed documentation referenced in this paper should be consulted before suggested changes are implemented. Any changes suggested herein should be fully tested and vetted with the Blackboard support organization before applying in a production environment. Methods that might deviate from what Blackboard supports will be mentioned but not covered in depth. Throughout this paper, any Linux path information is assumed to be relative to the root of the Bb Learn installation (normally `blackboard/`). Bb Learn also has a shared content area whose root will be distinguished as `content/`. The schema name `bb_bb60` is used throughout this paper. Depending on the version of Bb Learn, it may be `bblearn`. The account name `bbaccount` is fictitious and is used throughout as the name for the local account in which Bb Learn runs under.

2. Blackboard Learn Technology

Bb Learn is built on open source technology but also contains proprietary Java code. Most of it runs in a Java Virtual Machine which manages the memory and resources for the Java code. West (2010) states that, "The Blackboard application is a collection of Java Server Pages and Servlets (accompanied by occasional perl.) This code is nothing more than a set of instructions." Bb Learn code runs in the Apache Tomcat container, utilizing a subset of the Java EE (Enterprise Edition) servlet (Fontaine, 2009).

The servers that run Tomcat and the application code are known as application servers. In addition to this, a shared file system is required if more than one application server is configured. This is typically shared using NFS. The data is stored in a relational database, typically Oracle or SQL Server. In addition to the servlets (Bb code), Bb Learn also has Java tasks to handle housekeeping, session invalidation, and other things. Finally, there are stored procedures in the database that operate on the various Bb Learn tables.

Blackboard has established a reference architecture that includes load balancing, storage/recovery, and integration with SIS, SSO, etc. Secure performance and immunity are also listed as pieces of the reference architecture (Feldman, 2011).

Bb Learn includes Building Blocks, a method to extend Bb Learn. Anyone can write a building block in Java and package it into a `.war` file. The file is required when installing the Building Block. In fact, Bb Learn contains several "built in" building blocks. A building block typically includes features not found in the main Bb Learn code, hence the reason to write one.

Bb Learn also provides Web Services so that you can provide "call out" capability to your building blocks (connect to external Web Services) or expose your methods, among other things (Alcorn, 2005). The Axis toolkit provides for this. A Web Service provides interoperability so that messages can be exchanged using XML (Extensible Markup Language) according to the SOAP (Simple Object Access Protocol*)* specification (Pots & Kopack, 2003). Also included are the collaboration service and Wimba Classroom web conferencing. The collaboration service provides chat and a virtual classroom but no video/audio. However, Blackboard is releasing Blackboard Collaborate, the new branding for Elluniate Live! and Wimba Classroom which will be optional and available for purchase ("Transform the Teaching and Learning Experience," 2011). Finally, Bb Learn provides an array of command line tools to move content, archive courses/organizations, and a host of other utilities. From a Linux standpoint, these are very useful and provide a means to monitor Bb Learn and perform routine tasks.

2.1. Apache and Tomcat

As stated, Tomcat is the container that comes with Bb Learn. Chopra, Li, and Genender (2007) state that, "Tomcat's purpose is to provide standards-compliant support for Servlets and JSPs" (p. 9). Apache `httpd` is also provided. An inspection of the Apache log file reveals that Apache is using the `mod_jk` connector module to communicate with Tomcat. A client Web browser connecting to Bb Learn would interact with the Apache web server which in turn communicates with the Tomcat Servlet container using the `mod_jk` connector (Chopra, Li, & Genender, 2007). Bb Learn uses ehCache to manage the data lifecycle and data reuse. Cache settings can be configured via a properties file (Feldman, 2011).

Servlets are Java code and are "server side". A servlet can handle multiple requests from clients. However, content presentation is cumbersome. Fortunately, Java Server Pages simplify programming by separating the code from the content, utilizing tag libraries. JSPs are compiled into servlet classes when first accessed (Chopra, et al., 2007).

3. Policies, Procedures and Documentation

Peikari and Fogie (2003) state that "Security policies are part of any well-maintained information system" (p. 1). In the context of Bb Learn administration, this presents several policies that include: backup retention, server hardening, and acceptable use for

Bb Learn administrators. A policy must be part of the business process to succeed, not just an overlooked high level document (Peikari & Fogie, 2003). An organization should document a change control policy/procedure and develop detailed procedures for maintaining Bb Learn and the environment it runs in. If a change in configuration is going to be made, it should be approved by a change control committee and if approved, documented for future reference. Keeping track of patches that mitigate specific vulnerabilities is vital (Scarfone, Jansen & Miles, 2008).

3.1. Server Hardening Procedure

A risk assessment should precede a server hardening in order to help identify hardening measures to adopt (Scarfone, et al., 2008). In addition, Bb Learn documents specific requirements to be met in order to install the software. One should start with servers that are configured according to a server hardening procedure with necessary modifications to accommodate Bb Learn.

A server hardening procedure or standard will be developed by asking questions about the data stored on a server, connected storage, services needed, local accounts, etc. (Scarfone, et al., 2008). In the case of Bb Learn, it should be the only application hosted – the server(s) being dedicated to Bb Learn and not running unnecessary applications ("Blackboardlearn Release 9.1 Installation Guide," 2010).

3.2. Account Policy and Procedure

Bb Learn provides a very granular privilege mechanism including roles that come with a set of privileges. Policies and procedures should adopt the principle of "least privilege" by only giving required functionality to authorized users (Scarfone, et al., 2008). Bb Learn has special roles that can be used to manage courses, users, etc. The principle of "least privilege" would also apply for those who would be administering Bb Learn. One or more system administrators will require the "sledgehammer" role of "System Admin".

Bb Learn also allows one to create custom roles with granular privileges. One can start with a role with no privileges assigned, and build it with just the privileges required. A policy governing the use of and lifecycle of accounts assigned to built-in and custom roles should be mandatory. In addition, it is vital to develop policies and procedures to govern proper account usage. For example, those who hold a "Course Administrator" role may create courses and support instructors in course building. They may have the privilege allowing them to archive a course. This may not be advisable during peak hours due to performance issues and may even simulate a Dos (Denial of Service) attack. In addition, the question of what to do with stale accounts should be addressed? A policy and procedure can address such things.

If a Bb Learn support account is a local account (i.e. not in a central directory such as LDAP), the policy and procedure must address password change. The procedure should address how special accounts are created, approved, and monitored. Local accounts need to be tied to their Identity Management counterparts and disabled when gone from Identity Management and this should be automatic (scripted). If an account holder's identity falls out of an Identity Management system, their special Bb Learn local account should be automatically disabled.

It may not be acceptable to grant extra permissions to a support person's Identity Management Bb Learn account if they are also members of courses in Bb Learn as this may give them the ability to change a grade, etc. Thus, it is necessary to have a separate local account for someone who is a student and a course administrator.

3.3. Retention and Recovery

In addition to an institutional disaster recovery plan, a retention policy is also crucial. Courses should be archived frequently and recovery procedures in place ("Blackboardlearn Release 9.1 Installation Guide," 2010). In addition to regular Linux file system and database backups, one should consider course archives. A course archive is done on a per course basis to capture everything associated with a course including data stored in tables and content that is stored on a shared drive. This is essentially a snapshot of a course. Retention of course archives and online courses and other data should be documented. Issues such as who can create, retention cycle, offsite storage, and encryption of the backups should be addressed. If a course needs to be restored to settle a grade dispute, the retention document will be authoritative in how far back one can go to restore the course. Just relying on file system and database backup is impractical because restoring one or a handful of courses would require the creation of a separate Bb Learn environment just to get at those courses so they can be archived and then restored to a production environment.

4. Blackboard Learn Security

Blackboard is committed to fixing vulnerabilities found in Bb Learn and does release security advisories when warranted ("Vulnerability Management Commitment and Disclosure Policy for Blackboard Learn," 2011). While much of the Bb Learn code is proprietary, the service that runs the code (Apache/Tomcat) must also be taken into account when it comes to security. In this section, the deficiencies and risks associated with running Bb Learn will be discussed.

4.1. Deficiencies and Risks

Online24, a Dutch security company researched Bb Learn and released a report in 2010 identifying alleged Bb Learn vulnerabilities. Those vulnerabilities include cross-site scripting, insufficient authorization, information leakage, mail command injection, abuse of functionality, local file inclusion, improper file system permissions, and cross-site request forgery (CSRF). In one example given, it was stated that users of Bb Learn could bypass JavaScript filters because it is done on the client side in the WYSIWYG editor which can be turned off. The report lists Cross-site scripting as the vulnerability with the highest occurrence and among all vulnerabilities and 38 % of them were categorized as "high". The report also classifies the risk of each type of vulnerability using the Confidentiality, Integrity, and Availability security rating (Prins & Abma, 2010). The vulnerability technical details are not contained in that report.

Blackboard has taken this report seriously and has endeavored to address the vulnerabilities presented. Blackboard issued a bulletin addressing the report and vulnerabilities. They announced that they were working to provide updates and also provided mitigation techniques. Later versions of Bb Learn 9.1 contain countermeasures against cross-site attacks including 9.1 Sp5. Blackboard provided a timeline of releases for fixing the cross-site and cross-site request forgery vulnerabilities. Older versions of Bb Learn also had fixes planned ("AS-152479-C Vulnerabilities in Blackboard Learn Could Allow Elevation Of Privilege," 2011).

In late 2011, Blackboard released a more comprehensive advisory addressing the vulnerabilities. They confirmed that eleven groups of findings in the original report were valid, while five were not. They wrote that the five issues were related to the reporter's testing environment or security settings that were misconfigured. In Blackboard's advisory, they do indicate that some features of Bb Learn are at risk for phishing attacks by way of link injection vulnerabilities. They also acknowledge header injection, URL redirection, and cookie vulnerability (`HttpOnly` flag not set) exist. Product updates are available or planned to address the vulnerabilities ("LRNSI-2284 - Vulnerabilities in Blackboard Learn Could Allow Elevation of Privilege," 2011). As stated, later versions of Bb Learn 9.1 do offer countermeasures and these are included in the checklist (Appendix D).

Ristic (2005) states "Things can be unpleasant if you are running a vendor-supplied version of Apache, and the vendor is slow in releasing the upgrade" (p. 117). The current version of Bb Learn at the time of this writing ships with Apache `httpd` 1.3 which has reached end of life status as of 2010 and has been replaced by version 2. There is no guarantee for getting security patches for 1.3. Also, the default configuration is not hardened enough. The problem is that Blackboard has not supported decoupling Apache 1.3 though has plans to do so. This stale version issue is also true for Tomcat but future service packs will include updates for Tomcat. Blackboard Support has commented on several Tomcat vulnerabilities having to do with information disclosure, Denial of Service, and authentication bypass. Bb Learn 9.1 is only affected by http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-5515, an information disclosure vulnerability. However, no sensitive information would be disclosed. In Bb Learn, Tomcat runs behind a reverse proxy which also helps defeat an attack ("LRNSI-2284 - Vulnerabilities in Blackboard Learn Could Allow Elevation of Privilege," 2011).

Bb Learn has no anti-virus support but uploaded content should be scanned as part of any regular monitoring policy. Blackboard does have this (AVI integration) on the roadmap ("LRNSI-2284 - Vulnerabilities in Blackboard Learn Could Allow Elevation of Privilege," 2011). Daily scanning is an option and will be discussed in section 7. Bb Learn has a way to control uploading of files. For added protection there is the concept of a white list. By default, this is not configured in Bb Learn (Tan & Alexander, 2011).

Bb Learn local accounts have no lock out mechanism or password enforcement. Bb Learn does provide a mechanism to authenticate using an LDAP server securely but there may be times where local accounts are required (see section 3.2). Support for this feature is on the Bb Learn roadmap ("LRNSI-2284 - Vulnerabilities in Blackboard Learn Could Allow Elevation of Privilege," 2011).

5. Linux Hardening

Bb Learn has specific requirements for a UNIX installation. The Bb Learn 9.1 installation guide provides a checklist. It includes running only Bb Learn on an application server with no unnecessary software and a dedicated server hosting the Bb Learn database. The firewall must have TCP ports 80, 443 open and TCP ports 8010, 8011, and 8443 open if running the collaboration service (chat). If the database is being hosted on a separate server, the firewall rules must be set appropriately. The latter three ports are required for the collaboration (chat) server to run and it only needs to run on one application server. Bb Learn is started as `root` so that port 80 can be used. Port redirection is not supported ("Blackboardlearn Release 9.1 Installation Guide," 2010).

5.1. Basic Hardening

Identifying threats and impact is key to being able to plan for server security. If the data behind Bb Learn were compromised the impact would be "HIGH" since it holds student grades and other private data. This determines the level of protection that should be afforded to Bb Learn servers. There are two categories of protection measures. The first involves identifying security weaknesses. They should be identified and dealt with through patches, etc. The second one involves the principle of "least privilege" or limiting functionality based on user requirements (Scarfone, et al., 2008). Multiple layers of security or "Defense in Depth" should be employed when it comes to server security. A host based firewall, HIDS (Host-based Intrusion Detection System), monitoring, backups, and other measures should be in place. Staying current with patches is essential.

Bb Learn is certified to run on Red Hat Enterprise 64 bit, a Linux variant ("Blackboardlearn Release 9.1 Service Pack 5 Release Notes," 2011). A minimal Red Hat should be installed. The release notes do not specify any particular Red Hat packages but lists the requirement for the Java Development Kit (JDK). It also requires a local account that will be used to run Bb Learn and this account should have a strong password that complies with organization password policy, including reuse and aging ("Blackboardlearn Release 9.1 Installation Guide", 2010). Most command line server maintenance activities can be done using this local account.

Most importantly, after initial install, the O/S should be fully patched with the latest updates to the kernel and any other utilities that were selected at install time. Any unnecessary services, applications and protocols should be removed, including file/print sharing, wireless, directory services, smtp server, compilers/libraries, development tools and SNMP (Scarfone, et al., 2008). To protect data, create separate partitions for `/tmp`, `/home`, `/var` and `/var/log` ("Guide to the Secure Configuration of Red Hat Enterprise Linux 5," 2011). Other must have packages include `lsof`, `iptables`, `perl`, `tcpdump`, `wget`, `bash`, and `tar`.

After the initial O/S installation, identify listening services. The required Bb Learn ports will show up, as well as ports that do not need to be opened up in the firewall and are listed under the Java program (Tomcat related). Also, determine what services are starting and disable those that are not needed (Puschitz, 2007). See Appendix A.

Blackboard Learn needs shared content if run with more than one application server (typical for larger sites). The files and folders are owned by `bbaccount`. NFS is one protocol that can be used to share files, though not the only one. The NFS client needs to be installed on each application server and NFS should be secured by restricting access to the NFS server (TCP Wrappers or iptables), and configured to use TCP (Puschitz, 2007). Configuring NFS and other shared storage is outside the scope of this paper.

Another area of hardening to consider is kernel parameters which affect the operation of the Linux kernel. Some parameters affect networking and can be set to such that hosts will not provide network functionality. The kernel parameter `ExecShield` as an important defense to help prevent buffer overflows ("Guide to the Secure Configuration of Red Hat Enterprise Linux 5," 2011). See Appendix A.

5.2. Firewall

Network filtering and other countermeasures should be in place in front of the Bb Learn servers. Still, a host based firewall is important in the event that other layers stop functioning and to protect other hosts in the LAN (Ristic, 2011). Netfilter is a host based packet filter included with Red Hat and is activated by default. The program `iptables` is used to configure and control Netfilter. With Netfilter, the filtering happens at the kernel level prior to program execution. To see what rules are active, execute:

```
    iptables –vnL –line-numbers
```

An `iptables` rule set is divided into chains (sections) including the built-in chains `INPUT`, `OUTPUT` and `FORWARD`. There are two built-in targets: `ACCEPT` and `REJECT`. `ACCEPT` permits the packet to pass while `REJECT` blocks and sends an error to the originating host. A third target, `DROP`, drops the packet silently. The default rule set can be strengthened by editing the `iptables` rules file. First, make sure the default policy is `DROP`. ("Guide to the Secure Configuration of Red Hat Enterprise Linux 5," 2011). See Appendix A.

The Guide to the Secure Configuration of Red Hat Enterprise Linux 5 (2011) published by the National Security Agency contains instructions on how to restrict ICMP message types, removal of IPsec rules, logging/dropping packets with suspicious source addresses, and logging and dropping all other packets. To provide additional strengthening, blocking outgoing traffic would be helpful. This may be useful to prevent use of insecure protocols such as telnet. SYN flood protection can also be enabled using the "`iplimit`" and "`recent`" matches but care would need to be taken to ensure that there were no negative issues as a result. Enabling TCP SYN cookies would be more precise. This feature keeps legitimate connections coming during a SYN flood attack. ("Guide to the Secure Configuration of Red Hat Enterprise Linux 5," 2011). Brute Force SSH attacks are on the rise but `iptables` can be configured to defend against such attacks. There are rules that limit the rate of incoming SSH connections to help prevent success in brute force attacks (Van Zonneveld, 2007). See Appendix A.

As previously stated, Bb Learn requires TCP ports 80, 443 to be open. Bb Learn includes the collaboration (chat) server and in practice, is run on one application server. That application server must have TCP ports 8010, 8011 and 8443 open. See Appendix A for sample rules to support Bb Learn. The database and NFS servers should be isolated from the Internet. If mounting NFS, more rules are required. Additionally, some building blocks (Bb Learn extensions) may require a specific port to be open to exchange data via a web service or other means with some external source. Documentation for each building block should list those requirements.

5.3. Remote Management

Scarfone, Jansen, and Miles (2008) state "Remote administration of a server should be allowed only after careful consideration of the risks" (p. 48). There may be a need to access the Bb Learn application servers and database servers remotely to perform daily maintenance and troubleshooting. For this purpose, having the secure shell service running is an option. By default, this would require TCP port 22 to be open. Make sure SSH is locked down to not allow `root` login, configured to run in a `chroot'd` environment, configured to require the more secure version 2 of the protocol, and configured to enable checks of private ssh files found in a user's home directory (Puschitz, 2007). See Appendix A.

The firewall should be configured such that access to the application servers via SSH should only be from internal subnets. External access is not needed if a VPN is used (Virtual Private Network). Only specific authorized users should be permitted access via SSH enforcing the least privilege principle (Scarfone, et al., 2008). The `bbacount` user should not be allowed an interactive login and user files should be segregated from the server. Local users can initiate attacks, including Denial of Service (DoS). An example would be a fork bomb that replicates itself and fills up the process table (Ristic, 2005). Local server access should be limited to system administrators and Bb Learn Admins.

5.4. Configuration and Other Hardening

Accounts that are not needed (vendor, applications, etc) should be removed. The `find` command can be used to identify files that are owned by a specific account (Puschitz, 2008). Restricting `root` login is important – system administrators should "su" to `root` rather than log in directly. Restricting the interfaces that `root` login may occur on can also be accomplished as can the restriction of who may "su" to `root`. PAM can be configured to prevent password guessing of the `root` account password ("Guide to the Secure Configuration of Red hat Enterprise Linux 5", 2011). See Appendix A.

Bb Learn must be installed as `root`. In addition, `root` access is needed to start and stop the application and to run the `PushConfigUpdates` tool which is described in appendix F ("Blackboardlearn Release 9.1 Installation Guide", 2010). Enabling `sudo` is optional but would offer granular control over command execution from non-`root` accounts and better auditing ("Guide to the Secure Configuration of Red Hat Enterprise Linux 5," 2011).

For system accounts that are required but never need interactive login, one can lock the account and disable the login shell. The `bbaccount` group can also be created with only `bbaccount` account as its only member. This limits access (other than `other:rwx`) to only those who hold group membership in the `bbaccount` group ("Guide to the Secure Configuration of Red Hat Enterprise Linux 5," 2011). See Appendix A.

A Bb Learn application can be integrated with a central authentication service such as Kerberos or LDAP (discussed in section 6.3.1). Care must be taken in the deployment of any central service so that authentication information is sent over the network encrypted. Network Information Service (NIS) is not recommended as it is obsolete and had been supplanted by Kerberos and other authentication services. Configuring automated time synchronization is also important, especially when comparing logs from different machines to do forensics in the event of an attack or for troubleshooting. Kerberos and other cryptographic protocols require synchronized time in order to prevent certain kinds of attacks ("Guide to the Secure Configuration of Red Hat Enterprise Linux 5," 2011).

6. Blackboard Learn Hardening

Bb Learn is complex and hardening requires looking at all of the pieces that come together to run Bb Learn, including Apache, Tomcat, Bb Learn code and security mechanisms, Building Blocks and Web Services.

6.1. Hardening Apache

Bb Learn version 9.1 Service pack 5 comes bundled with Apache 1.3 which has been deprecated. The current version of Apache is preferred due to active development, including security fixes and enhancements. Therefore, understanding Apache hardening is very relevant.

Bb Learn is not affected by the Apache 1.3 vulnerabilities that were raised and with Bb Learn 9.1 Service Pack 8, Apache `httpd` will be decoupled and a current version can be utilized along with the latest version of `openssl` (Tan & Alexander, 2011). Bb Learn Apache is linked with an outdated version of `openssl`. While it is not officially supported by Blackboard, some have decoupled Apache 1.3 from Bb Learn (Bitpushr, 2011).

6.1.1. Apache Modules

The Apache that is bundled with Bb Learn is compiled to use dynamically loadable modules, supporting the use of the `LoadModule` directive. The default configuration file that ships with Bb Learn reveals that several modules are loaded using `LoadModule`. This means that a module can be loaded as needed and that if an attacker were to compromise the server, he/she could load a backdoor module without having to recompile Apache (Ristic, 2005).

The Apache bundled with Bb Learn uses `mod_jk` to have the Apache web server send requests to Tomcat so that a servlet can handle the requests (`login.jsp` for example). This is the main purpose of the Apache web server in the context of Bb Learn. Even so, Apache needs to be secured. Configuration options can be utilized to harden Apache. This does somewhat depend on what modules Apache is configured for. Some modules are considered dangerous. More dangerous modules include `mod_userdir`, `mod_info`, `mod_status`, and `mod_include` while recommended modules include `mod_rewrite`, `mod_headers`, and `mod_setenvif` (Ristic, 2005). To list modules that were compiled into Apache, simply issue the following command.

```
apps/httpd/bin/httpd -l
```

Ideally, modules that are not needed should not be included when building Apache. Those implementing Bb Learn will have control over this when Apache `httpd` is unbundled from Bb Learn in a future release.

6.1.2. Apache Account

A separate account is recommended for Apache so that if an attacker compromises that account, he or she will only have the privileges of that account. A group of which the account is a member of is also suggested (Ristic, 2005). Bb Learn's Apache is already configured to use a separate account. See Appendix B.

If Apache is to listen on port 80, it must be started as `root`. Providing write access to the Apache binaries to `bbaccount` (or any other) would enable it to run anything as `root`. An attacker could replace the Apache binary which would then run as `root`. Therefore, only `root` should have write access to those files and folders. In addition, the logs and configuration folders should be off limits to the `bbaccount` (Ristic, 2005). Bb Learn has other folders (not part of Apache) that need to have `bbaccount` write access so one must take care to only change files/folder under `apps/httpd`.

6.1.3. Locking it Down

By default, Apache will serve any accessible file. This can be controlled using the `Directory` directive and other options (Ristic, 2005). With Bb Learn, there is a `bin` folder configured but the folder does not exist so this should be removed. The default log setting is set to write to `access_log` using the `CustomLog` directive. In a busy environment, this log can grow fast so some mechanism to rotate the log is needed. Bb Learn includes the Apache utility `rotatelogs`. It or some alternative should be used for the `access_log`, `error_log` and `mod_jk` logs. See Appendix B.

By default Apache responds to a header request `HEAD` with the Apache version number, and other information that could be useful to an adversary. This should be changed to not disclose this information using the `ServerTokens` directive. This will result in only `Apache` being disclosed, and not the version (Ristic, 2005). See Appendix B.

Unintentional file disclosure can also be an issue with Apache but this can be mitigated using the `FilesMatch` directive. This directive can be used to deny access to file types that should not be present. Automatic indexing should be turned off using "`AllowOverride none`". This causes Apache to ignore any `.htaccess` configuration files (Ristic, 2005). See Appendix B.

If an adversary can identify information about a web server (whether Apache, IIS, etc) he/she can attempt to exploit vulnerabilities associated with the web server identified. Web servers do not implement HTTP the same and because of that, we have HTTP fingerprinting. For example, Apache can be identified by URL encoding a forward slash ("/") as in `http://www.myserver.net/%2f`. Apache returns a 404 error. IIS returns a valid page. In Apache 2, this can be configured to behave differently using `AllowEncodedSlashes`. With Apache 1.3, you can at least mislead an attacker by changing the server header field but you have to do it in the source code and recompile. The other option would be to use `mod_security` (Ristic, 2005). This is not likely supported by Blackboard and may present technical problems. However, as previously mentioned, Blackboard Learn 9.1 Service pack 8 will unbundle Apache and at that point, the "`mod_headers`" module can be used with the "`Header set Server`" directive. However, it can be circumvented (Ristic, 2005).

The `robots.txt` file can be used to prevent well behaving web crawlers from indexing the Bb Learn site. This may not stop all web crawlers but at least the ones that play by the rules. See Appendix B.

Default content that is included with Apache should also be removed including that found in `cgi-bin` and `htdocs`. Apache 2 also contains the `manual` folder which can be removed (Ristic, 2005). There is no issue since Bb Learn does not ship with these folders.

Rewrite rules can mitigate Apache attacks. For example, take "Bandwidth Stealing" where rogue sites link to images on the victim's site. One can use `mod_rewrite` to reject such requests for images not originating from the site. One can also use `mod_rewrite` to block requests for specific file types (Ristic, 2005). This might be useful if there were a lot of requests for `.php` but since Bb Learn does not run `php`, it is undesirable to process these requests that result in errors in the `error_log`. See Appendix B.

6.1.4. Blackboard Recommendations

Blackboard has recommendations for Apache. Bb Learn has the HTTP `TRACE` verb enabled in the default configuration of Apache that is shipped. They recommend disabling it. Having `TRACE` enabled renders Bb Learn vulnerable to a cross-site tracing attack possibly exposing users to compromised credentials. Blackboard also recommends disabling SSL v2 due to the risk of information exposure. SSL v2 allows low and medium strength ciphers and is at risk due to SSL v2 protocol flaws, possibly leading to the compromise of confidentiality. Blackboard does have plans to address these and other issues in a later service pack (“LRNSI-2284 - Vulnerabilities in Blackboard Learn Could Allow Elevation of Privilege,” 2011). Additionally, in the `httpd.conf` file that ships with Bb, there is an option to reject unsafe URLs that is turned off by default. This however can be turned on. See Appendix B.

6.1.5. Configuring Apache for SSL

Bb Learn provides specific instructions to set up SSL. Bb Learn comes with `openssl` which is used to generate the certificate signing request needed to acquire a CA signed certificate. Configuring Bb Learn to use a certificate is as simple as editing the main `config/bb-config.properties` and `apps/httpd/conf/httpd.conf.bb` files (“Blackboardlearn Release 9.1 Server Administration Guide,” 2010). See Appendix B. The certificate must be shared among all application servers and the load balancer (if used) must not use either SSL acceleration or off-load features (“Blackboardlearn Release 9.1 Installation Guide,” 2010). A 2048 bit key is recommended for Bb Learn (“LRNSI-2284 - Vulnerabilities in Blackboard Learn Could Allow Elevation of Privilege,” 2011). In cryptography, a longer key is better. This may affect older browsers that might be used on older devices but those browsers would not be officially supported by Blackboard. Note that setting up SSL does not mean Bb Learn will use SSL (TCP port 443). Some remaining steps are needed and covered in section 6.3.2.

6.1.6. For Further Discussion

The `mod_security` module written by Ivan Ristic is an open source module that provides an added layer of security features, essentially functioning as a web application firewall. It provides better logging, intercepts HTTP requests including the POST payload and uploaded files and performs anti-evasion actions. In Apache 2, `mod_security` avails of the advanced filtering API which allows for the interception of the response body (Ristic, 2005). There is no indication that Blackboard would even support `mod_security` or whether it would interfere with Bb Learn. As previously discussed, Bb Learn does have some evasion techniques of its own. For more information on `mod_security`, please see www.modsecurity.org.

6.2. Hardening Tomcat

Tomcat has some configuration options that can improve security. It is important to have a basic understanding of the Tomcat architecture including location of the files, and how it works. Tomcat is the foundation beneath the Bb Learn application.

6.2.1. Tomcat Architecture

The Bb Learn install tool does install Tomcat under the `apps/tomcat` folder and this is defined by the environment variable `CATALINA_HOME`. The `apps/tomcat` folder has the following subfolders: `bin`, `conf`, `lib`, `logs`, `temp`, `webapps`, `and work.` The `bin` folder contains startup/shutdown scripts, jar files, and other utilities. The `conf` folder contains the configuration files used to configure Tomcat's behavior and security (Chopra, et al., 2007). Configuration files are discussed further in section 6.2.2.

The `lib` folder houses the jar files that the container uses including the Tomcat jars, jars for the API, and jars shared across web applications (Chopra, et al., 2007). For example, Bb includes the axis2 jars which are used by Building Blocks that leverage Web Services (Alcorn, 2005). The `webapps` folder contains Web applications. Normally, placing a WAR file in this folder leads to deployment of the Web app. However, the folder in a Bb Learn installation only contains the `host-manager`, `manager`, and `ROOT` applications which are not accessible in Bb Learn. The Bb Learn `webapps` folder lives under the root of the Bb Learn installation. The `work` folder is for temporary files and is used for JSP compilation when JSPs are converted to servlets (Chopra, et al., 2007). In Bb Learn, there are many folders under `apps/tomcat/work/Catalina/localhost`.

Chopra, et al. (2007) state that, "Tomcat 6 consists of a nested hierarchy of components" (p. 54). The Server is at the top and is Tomcat (Web application server). It is an instance inside of the Java Virtual Machine (JVM). Within the Server, there is the Service which is a top level component that maps a container with a set of Connectors. The Connectors connect applications to clients, receiving requests that are assigned a port (Chopra, et al., 2007). See Appendix E for an overview of the Tomcat architecture.

Bb Learn uses the JK connector ("Blackboardlearn Release 9.1 Installation Guide," 2010). Also within the Server is the Engine, the top-level container that processes requests. It is the Catalina Servlet engine that examines HTTP headers to determine where requests should be passed. In the case of Bb Learn, Tomcat has been configured for use with a Web server front end (Apache). This means that the default class for serving requests is overridden because the Web server determines the destination (Chopra, et al., 2007).

The other parts of the Engine (container) include the Logger, Valve and Realm. Loggers track the state of a component and can be configured at the Engine level (not limited to) and then will be inherited and assigned to every child object unless overridden. The Realm handles user authentication and authorization (Chopra, et al., 2007). Examining the Bb Learn `server.xml` would seem to indicate that Realms are not used by Bb Learn. Bb Learn implements a custom authentication scheme capable of integrating with LDAP, etc. ("Blackboard learn Release 9.1 Installation Guide," 2010). A Valve is a component (typically reusable) that intercepts requests and preprocesses them. An example would be a value to do access logging. It is also common to see them used for single sign on. Valves are configured in `server.xml` (Chopra, et al., 2007). Bb Learn contains a valve to produce the `logs/tomcat/bb-access-log.YYYY-MM-DD.txt` file to log web hits that have been passed to Tomcat. This log format is very similar to the Apache logs and is very useful when doing forensics. If the Apache log contains a hit to a valid jsp, it will also show up in the `bb-access-log.YYYY-MM-DD.txt` file.

Nested under the Engine is the `Host` (a child of the Engine), similar to the Apache virtual host. Virtual hosts in Tomcat are defined by a fully qualified name. The applications associated with the virtual host are located in the `$CATALINA_HOME/webapps` folder. Multiple hosts can be configured for the same server. The Engine inspects the HTTP header (from a client) to determine which `Host` should process the request. Nested within the Host is the Context or the web application. A context can be configured for the application, specifying initialization parameters. It is a Web application container and parent of servlets and filters (Chopra, et al., 2007). Bb Learn configures localhost in server.xml.

As stated, Bb Learn uses the mod_jk connector. In Tomcat, there is a Connector module for each Web container instance. Bb Learn has two connectors configured, one for cross-server communication, and one for AJP. The Apache web server will service requests for static content or non-servlet dynamic content. Requests destined for Tomcat are passed to the mod_jk module (Apache) which encodes and sends the request to the Tomcat Connector where the Servlet container services it and sends the response back to the Apache connector module. The protocol used by JK to exchange data between Apache and Tomcat is AJP (Apache Jserv Protocol). The data is transferred in a binary format. The JK connector has better support for SSL than its predecessor, mod_jserv (Chopra, et al., 2007).

6.2.2. Tomcat Configuration

Tomcat 6 configuration consists of XML files which are edited before starting Tomcat. Bb Learn ships with customized XML files. Tomcat uses the Apache Commons Digester, a utility that reads XML files, parses the rules, and creates Java objects – triggered by a Tomcat start. These files are normally edited before starting Tomcat 6. The main configuration file for the Tomcat server components is server.xml. Here one can configure the Service, Connector, etc. Each of these components has configurable attributes. This determines how the container is built when Tomcat starts (Chopra, et al., 2007).

The XML in server.xml that comes with Bb Learn follows the Tomcat architecture. Understanding the Tomcat architecture can help understand the configuration since there is a direct correlation. The configuration file for server components (for each application) is context.xml. This covers all running applications, unless overridden by a specific application. Since it is global in nature, it should only contain configuration options needed by all applications. The default context.xml only contains the Context element, defining WatchedResource for `WEB-INF/web.xml`. This file will be watched by a background process for changes and if detected, the Tomcat autodeployer will redeploy the application.

The file `web.xml` is the default standard Java EE deployment descriptor used by Web applications. This file configures applications to handle JSPs and static resource like HTML. Applications can override this as well. Applications can override this configuration by specifying their own `web.xml` file, located in the application's `WEB-INF` folder. Some of the Bb applications have a local `web.xml` configuration. The main `web.xml` file can have configuration information for static resource, servlet invocation, JSP page conversion to servlet, and URL/Servlet mappings (Chopra, et al., 2007).

Bb Learn includes a customized `web.xml` file containing numerous elements to handle SSL proxying, session management (no-cache), XSS and content type filtering (more on this in section 6.3.4), servlet configurations, servlet mappings, filters, mime type mappings, the welcome file list and error handling. Also included is the `session-timeout` attribute which by default is set to 30 minutes. This is the default session timeout if the session persistence is not in use (Chopra, et al., 2007). However, Bb Learn uses a different mechanism to time sessions which will be covered in section 6.3.

Other files that would be pertinent to Bb Learn include `catalina.policy`, where Tomcat can be configured to run in secure mode by configuring the Java `SecurityManager`. It can be used to guard against bad code or JSPs that might do bad things with unintended consequences. The `catalina.properties` file is accessed upon startup to allow for internal package access and definition control and also controls the contents of Tomcat class loaders. It also contains packages that are not to be overridden by code in servlets and JSPs. The file `logging.properties` is used to configure the level of logging as well as log destination (Chopra, et al., 2007).

When examining the `server.xml` configuration file in Bb Learn, one can see that a `Server` is defined near the top, correlating to the Tomcat architecture. The server is configured to listen on port 8005, telling Tomcat to listen on 8005 for a shutdown command.

```
<Server port="8005" shutdown="SHUTDOWN">
```

Connecting to that port using "`telnet localhost 8005`" and issuing the `shutdown` command will cause the server to shut down. Note that Apache stays running. After that, we see that two `LifecycleListeners` are added at the server level. When an event occurs, these can execute specific code (Chopra, et al., 2007). Next, `GlogalNamingResources` is used to configure the two data sources used by Bb Learn. Then the service name `Catalina` is defined, containing the nested connectors. Of particular interest is the AJP connector listening on port 8009. Apache and Tomcat exchange data using AJP 1.3. The `Engine` is defined, followed by the `Host` – which defines the virtual host. It's "name" is defined as `localhost` and its `appBase` as `config/tomcat/webapps`. The configuration options to use Tomcat clustering are included but commented out. This is not a standard configuration.

Next, we see a `Valve` configured for access logging. The log file configured here is a key file for monitoring and forensics as it includes the IP address, internal user key, and other items also found in the Apache logs. Following the `Value`, we have several `Context` entries that map application paths to the shared content path, including `ResourceLink` items defining the data sources for `webapps/blackboard`. This configuration follows the Tomcat architecture structure, shown in Appendix E.

Finally, the folder `jk` contains the file `workers.properties` for configuring the JK connector. Another file to note is the `tomcat-users.xml` file which is used to configure access to Tomcat admin applications (Chopra, et al., 2007). These admin applications are not configured by Bb Learn.

6.2.3. Securing Tomcat

This section discusses basic Tomcat security hardening. Securing Tomcat may involve changing some configuration files and reviewing other settings.

Chopra, et al. (2007), outline a general strategy for Tomcat security. First, remove any application or Tomcat component that could fall victim to exploitation. Second, lock down the rest using Tomcat and O/S mechanisms.

Tomcat comes with some default Web applications such as `ROOT`, `manager`, and `host-manager`. Some present a security risk. While Bb Learn comes with these three, they are not deployed. Still, they should be removed. Bb Learn does not ship with JSP and Servlet examples, which otherwise might be exploited by attackers. We also need to make sure the SSI (server-side include and filter), the Invoker, and CGI servlets are not enabled. A request filter value is also recommended to restrict host access (Chopra, et al., 2007). Bb Learn has the Invoker, SSI, and CGI commented out but it is good to verify. Bb Learn does not have a request filter value configured. See Appendix C.

A key configuration file mentioned earlier is the `catalina.policy` file. The Tomcat security manager provides access control for Tomcat and Web applications, which is configured through the use of this file. Tomcat 6 implements the security manager based on the Java 2 security model. With security enabled, access to system resources is by default prohibited, unless specifically allowed. When Tomcat is started with security (`startup –security`), the `catalina.policy` file is enforced. Near the beginning of the file, we see the line:

```
// These permissions apply to javac
grant codeBase "file:${java.home}/lib/-" {
        permission java.security.AllPermission;
};
```

The first few lines of the file grant the Java compiler and runtime code complete access to every resource (Chopra, et al., 2007). In the Bb Learn version of this file this is clearly marked as "`SYSTEM CODE PERMISSIONS`". There are sections for Catalina code, web applications, and Bb Learn. It is important to be aware of these permissions, especially in the event a Web application does something bad (Chopra, et al., 2007).

As mentioned, the Security Manager works off of the policy file (`catalina.policy`) that contains all of the `grant` entries. Applications can be granted permissions, as can folders. The syntax for the `grant` command is:

```
grant {
     permission_class_name "target" "action";
}
grant codeBase "URL" {
     permission_class_name "target" "action";
```

```
        }
```

The first grant applies to all applications while the second applies to local folder or code from a URL. Each `grant` block can have more than one permission specified. If the `permission_class_name` is `java.security.AllPermission`, then all other permissions are granted. Another example is `java.net.SocketPermission` which allows socket connections. Each `permission_class_name` can have one or more targets. An example of a `target` would be `createClassLoader`, a target of the `java.lang.RuntimePermission`. A permission class can optionally contain an `action` (Chopra, et al., 2007).

It might be a good idea to regularly monitor Bb Learn to ensure that the Security Manager is running. If it was not running, malicious or badly written code could perform any kind of action on the server which may include opening connections, creating files, etc. (Chopra, et al., 2007). See Appendix C. Changing the `catalina.policy` file that ships with Bb Learn could be risky and is almost certainly not supported by the Blackboard Support organization. However, understanding the permissions contained in the file would be wise. Pointing out issues to Blackboard Support would also be a good practice. Chopra, et al. (2007) state that, "While it may be tempting to use the Java Security Model in place of securing the file system via O/S permissions, such a tactic is unwise. Relying on the O/S provides an important extra layer of security in the event that the Java Virtual Machine itself becomes compromised and exploited" (p. 354).

It is possible that Tomcat could be compromised. To limit the damage that one could do, a separate non-privileged account is recommended. The account will need to have the appropriate environment variables set, before starting Tomcat. The Tomcat account will need access to the Tomcat installation folder and specifically to `bin`, `lib`, and `webapps`. The account must also have read and execute permissions on the Java/JRE folder. Consider locking down the Tomcat installation folder so that the account has ownership and group and other have read only (Chopra, et al., 2007). Fortunately, Bb Learn requires the use of a local account (`bbaccount`) to run Tomcat ("Blackboardlearn Release 9.1 Installation Guide", 2010). There should be no need to grant access to the Tomcat installation to `other`.

6.3. Hardening Blackboard Learn

The Bb Learn application contains proprietary code written in Java. Securing the Bb Learn web applications is just as important as securing Tomcat, Apache and the O/S. Bb Learn provides the mechanisms and the Blackboard Support organization provides the recommendations to run it in a more secure manner. In this section, we examine those mechanisms. It is also critical to note the importance of being at the latest service pack for Bb Learn. Service packs will contain the most fixes. Tan & Alexander (2011) suggest applying newer service packs and patches immediately and subscribing to Behind the Blackboard email notifications and implementing mitigations.

6.3.1. Bb Learn Authentication

Blackboard recommends integration with single sign-on, LDAP, or Active Directory to be able to enforce a password policy that includes throttling and locking to limit password guessing and retries (Tan, 2011). In this section, we discuss Bb Learn LDAP integration - configuration and risks.

Tomcat provides declarative security measures which can be configured by modifying XML files. Programmatic security is Java code based such that security checks are done in the code (Chopra, et al., 2007). Bb Learn provides LDAP integration for authentication, as well as local RDBMS authentication (the default). Bb appears to mimic a Tomcat JDBC Realm but examination of `server.xml` does not reveal the configuration of a Realm. Bb Learn does not appear to use a JNDI Realm to configure LDAP, which could store authentication data. Using JNDI would involve creating the LDAP schema, installing the JNDI LDAP driver, and configuring the Realm in `server.xml` (Chopra, et al., 2007). Bb Learn does not have this but does support authentication against LDAP, using standard JNDI and the following provider:

```
blackboard.platform.security.authentication.LDAPAuthModule
```

Bb Learn also supports RDBMS authentication against the database (Ashman, 2006).

In Bb Learn, LDAP can be configured to use SSL (LDAPS) so that communication is encrypted. It is critical to protect user credentials by configuring SSL (see "Configuring Bb Learn for SSL"), especially when using LDAP as the LDAP authentication mechanism cannot encrypt them. Due to the added configuration, and performance overhead, the general recommendation is for Bb Learn to have a private non-routed LDAP server connection without SSL enabled ("Secure LDAP Authentication," 2011). If these issues do not manifest, configuring Bb Learn to use LDAPS would be a good choice. See Appendix D.

The following two settings referenced in Appendix D have some security ramifications:

```
auth.type.ldap.error_fallback_to_bb
auth.type.ldap.user_not_found_fallback
```

If you are supporting local accounts i.e. no linked account in an Identity Management system, then the second setting (above) is needed to log into those local accounts. The first settings (above) would cause authentication to fall back to Bb Learn if there was an error authenticating with the LDAP server. Both settings have the following risks. First, the passwords are not synchronized for users that are in LDAP. Second, if a DoS attack were successfully mounted against the LDAP server, a user could attempt to log into local accounts that were left with default passwords. One option is to synchronize user data between Bb Learn and LDAP ("Blackboardlearn Release 9.1 Server Administration Guide," 2010). If the only reason for failover is to support some local accounts, then for all other users accounts that are also in LDAP, one can set the Bb Learn passwords to a long random string to prevent log in when LDAP is down. This should be an automated daily process.

The LDAP interface was designed with constant LDAP availability assumed. There exists the option to use the OpenLDAP client for performance reasons. The jar file path for OpenLDAP is "`systemlib/bb-openldap.jar`" and must be appended to the `BB_CP` variable in `system/build/bin/launch-tool.sh` ("Blackboardlearn Release 9.1 Server Administration Guide," 2010).

6.3.2. Configuring Bb Learn for SSL

The assumption being made here is that no SSL offloading is being used with a load balancer (if multiple application servers). The prerequisite for enabling SSL from within Bb Learn is that SSL for Apache must already have been enabled (see section 6.1.5).

SSL Choice is the feature Bb Learn provides to encrypt all or some of Bb Learn traffic. The recommendation is to secure all of Bb Learn traffic. This is done from the System Admin tab ("Setting Up SSL Choice," 2011). See Appendix D. For the best security, select *SSL system-wide*. This is a strong recommendation from Blackboard and the action will secure Bb Learn Web Folders (WebDAV) authentication which normally occurs in plain text ("Blackboardlearn Release 9.1 Server Administration Guide," 2010).

The collaboration server can be configured to use SSL. Configuring SSL for Bb Learn does not secure the collaboration server (chat). The tool uses Tomcat and requires a separate SSL certificate, the creation of a keystore, and the configuration of Tomcat properties to use SSL ("Blackboard Learn Release 9.1 Server Administration Guide," 2010). See Appendix C. However, one needs to be aware that only the Java applet is downloaded over SSL. The chat communication occurs over TCP port 8010.

6.3.3. Mitigating Vulnerabilities

Blackboard has acknowledged vulnerabilities found in the Online24 report referenced earlier in this paper. The vulnerabilities involve tampering and privilege elevation. At the time of this writing, Blackboard has provided some steps to help mitigate most of these issues and is releasing patches or has plans to. For Bb Learn 9.1, the tampering issues have to do with authorization issues in the address book, calendar, Link Checker Building Block, Portfolio Comments and Display, and Tasks. Exploitation in most cases requires authenticated access to Bb Learn. Mitigation involves disabling vulnerable tools from the *System Admin* tab in Bb Learn. See Appendix D. There is no work-around for the "Portfolio Comments and Display" vulnerability. Monitoring is suggested ("AS-152479-A Vulnerabilities In Blackboard Learn Could Allow Tampering," 2011).

Other vulnerabilities include Cross-site Request Forgery, Cross-site Scripting, and verbose error messages. The "verbose error messages" issue is triggered by an error event which may display a Java stack trace which might be used to gather information about Bb

Learn. It will be addressed in the upcoming service pack 8. Fortunately, "Cross-site Request Forgery", and "Cross-site Scripting" are addressed as of Bb Learn 9.1 Service Pack 5 (currently available). To address the cross-site scripting vulnerabilities prior to Service Pack 5, one can utilize the Cross Site Scripting Security Control (more on this in the next section) to decrease exposure ("AS-152479-C Vulnerabilities In Blackboard Learn Could Allow Elevation of Privilege," 2011).

In June of 2011, Blackboard announced a patch to address a cross-site scripting vulnerability in the Bb Learn Discussion Board tool. As a workaround, the Cross-site Scripting Global Filter security control could be enabled to decrease exposure ("AS-158601 – Cross-site Scripting Vulnerability in Blackboard Learn Discussion Board Could Allow Elevation of Privilege," 2011). The patch is a better option since it resolves the issue. The security control is discussed in the next section.

In addition, Blackboard is investigating reported vulnerabilities in Blogs, Course Contacts, Glossary, Tests, and Course Area Titles features and has confirmed these are at risk for phishing attacks. This is due to link injection vulnerabilities. The attacker must be authenticated in order to exploit these weaknesses and the actions would be logged. Log monitoring is the only defensive action to take here, until Blackboard releases a planned service pack addressing these issues ("LRNSI-2284 - Vulnerabilities in Blackboard Learn Could Allow Elevation of Privilege," 2011).

6.3.4. Bb Learn Cross Site Scripting Security Control

As of Bb Learn 9.1 Service Pack 1, Bb Learn includes the XSS Security Control. Its purpose is to sanitize user input before it goes out again. This is considered untrusted data controlled by the end user. The recommendation is to set this to the highest level of protection. It is important to stay current with Bb Learn updates in order to have the latest rule set. It is also important to be aware of possible issues that may occur with building blocks that allow any user to submit HTML content. However, exceptions can be configured ("Using the Cross-site Scripting Security Control," 2011). See Appendix D.

Dynamic content can also be restricted using a Bb Learn privilege. In this way, the principle of "Least Privilege" (see section 3.2) can be followed. The privilege "*Add/Edit* *trusted content with scripts*" can be assigned based on need. It restricts the ability to enter JavaScript. By default, students and guests do not have this privilege ("Using the Cross-site Scripting Security Control," 2011). See Appendix D.

6.3.5. Avoiding Denial of Service Attacks

Denial of Service attacks (DoS) in the context of Bb Learn can be from various origins. These include misbehaving code, deliberate attacks i.e. malicious intent, misuse e.g. students guided to perform the same action at the same time, and unintentional abuse where a user resubmits multiple times. Bb Learn has a limit on the number of sessions which is user configurable and this number would likely be exceeded in a DoS attack. Parameters need to be set correctly and matched up where appropriate. The Tomcat threads should be the same as the maximum sessions. Each Tomcat thread will typically make at a minimum, one connection to the Bb Learn database requiring the database pool size to be the same as the Tomcat threads. Bb Learn Content System requests may require additional connections to the database. Tuning for this is done at the System Admin panel in the application. A DoS may also result in a full garbage collection in which all client activity is frozen. If the heap size is large enough, additional `jvm` arguments could avert a full garbage collection in the event of a DoS ("Denial of Service Attacks," 2011). See Appendix D.

There are some other actions to help avoid a successful DoS attack. Using a 64 bit JVM will allow for the configuration of large heaps which can prevent heap starvation during an attempt. Multiple application servers with a front end load balancer can also help mitigate attacks, marking an affected server as dead and directing requests to a different server. The usual monitoring of the number of web connections, as well as database connections and resource consumption can help identify a DoS. To identify application components that are acting up, one can monitor Tomcat for long running threads or the database for out of control queries, and other requests (see Appendix F). Raising the webserver's connection limit is not a solution as it may make the problem worse ("Denial of Service Attacks," 2011). If the Content System is used, there are features there to limit DoS attacks. See section 6.3.8.

6.3.6. Bb Learn Privileges

Bb Learn provides a privilege mechanism that is quite granular. The privileges are accessed through the *System Admin* tab in the application. From here, one can configure roles and privileges. There are mainly three types of roles: *system*, *Course/Organization*, and *institution*. Custom roles can be created for each type. System roles include *System Admin*, *Course Administrator*, and others. Course/Organization roles include *Instructor*, *Student*, *Grader*, and others. These roles are assigned privileges. *Institution* roles are used to control access to Bb Learn tools and features (content and services) but are not assigned privileges.

A vulnerability in Bb Learn 9.0 prompted Blackboard to issue best practices regarding privileges and courses. This was prompted by a replay attack vulnerability found in native Bb Learn authentication and the recommendation was to disable this. This disables the vulnerable challenge-response. However, turning this off means passwords are transmitted in plain text. If SSL is enabled system wide, this is not an issue. Other recommendations included the following changes: removing the privilege granting instructors and organization leaders the ability to create an archive, create users, change passwords, modify profiles, and manage enrollment. A Bb Learn course archive (a zip file) can contain user passwords. Blackboard also considers the information in a course to be protected information and a reason to restrict archiving. Blackboard also recommends that instructors should not manage users or enrollment and that course and organization creation should also be restricted ("Security Communication Regarding Course Archive," 2011). Bb Learn 9.1 does split the privilege to create archives – with and without passwords. See Appendix D.

As mentioned, Bb Learn contains several built-in *system* roles that have assigned permissions. If those are going to be assigned to users, they should be examined for the privileges granted to them. For example, the *Course Admin* role. It may have permissions assigned that are risky, such as *Batch Create Users*, "*Archive Course/Organization, including user passwords*", and *Delete Courses*. Having a course accidentally removed presents a problem since getting the course back generally requires a database and content restore on a separate system, unless the course was archived using the course archive tool. Privileges are a good way to implement the principle of "least privilege" by creating custom system roles with just the minimal privileges assigned for functions such as Help Desk, Faculty Support, etc.

6.3.7. Other Hardening

Bb Learn has a Java task that handles session invalidation. Sessions are stored in the `SESSIONS` table. The default session timeout is three hours. However, the task runs once per hour and therefore a session can be active for four hours. Longer session timeouts mean more security risks ("The SessionInvalidation Task," 2011). Consider implementing a shorter timeout, especially if students are accessing Bb Learn from labs where a longer timeout could be easily exploited if someone were to leave a browser window open. For development/test servers, consider a much shorter timeout. See Appendix D.

Bb Learn has a feature to allow guest access to the system. If the feature is enabled, instructors can make course areas available to unauthenticated users ("Blackboardlearn Release 9.0 Administrator Guide", 2009). Some may be hesitant to give guest access to unaffiliated students and other users, and rightly so. Perhaps a better way would be to have centralized accounts, granted to affiliate users. With guest access, there is no way to associate a person with the guest access. Also, having a feature enabled that may not be needed could open up Bb Learn to vulnerabilities related to that feature. For example, a recent vulnerability with Avatars could have been mitigated by disabling guest access ("Blackboard Security Advisory - Vulnerability in Blackboard Learn Avatars Could Allow Information Exposure," 2011). See Appendix D for information on turning off guest access and setting other gateway options. Other tools in Bb Learn can be disabled depending on assessed risk. An example would be the Email tool and its potential for abuse.

Bb Learn provides a "Session Fingerprint" option which can help detect and prevent a successful session hijacking (Tan & Alexander, 2011). Session Fingerprinting will not work if Persistent *Cookies* (section 6.3.8) are enabled (Tan, 2011). Session Fingerprinting uniquely identifies users by using their IP address, user agent or both. Session fingerprinting should be turned on. The setting depends on whether the hardware (load balancer) in front of Bb Learn forwards a load balancer IP or actual client IP. If the actual

IP is forwarded, then the fingerprint value could be by "IP Address" or "IP Address" and user agent. The settings allow for a new session to be created when the fingerprint changes, forcing users to re-login. With this option turned on, session fingerprint changes are logged. See Appendix D.

Bb Learn has a "Grade History" feature which can be turned on or off. It is recommended that it be turned on and configured to prevent instructors from changing this for a course. There is also a setting to prevent instructors from clearing grade history. The recommendation is to enable this (Tan, 2011). Having grade history is a useful forensics tool. See Appendix D.

ActiveMQ was introduced to Bb Learn 9.1 Sp3. It is a broker for the Java Message Server which permits message sharing in a cluster. Even though the vast majority of Bb Learn installations won't use ActiveMQ, the recommendation is to have it properly configured. Having it improperly configured and can impact stability ("How To Configure ActiveMQ for 9.1," 2011). See Appendix D.

6.3.8. Content System

The content system is included in Bb 9.1 and is used to store course, organization, and other content. A license allows for expanded features though course content is stored there regardless, even if the full version is not licensed.

The Content System has a *Persistent Cookies* setting. The recommendation is to disable this feature. With this disabled, users may need to authenticate multiple times to retrieve a file but with it turned on, the session hijacking risk is lessened (Tan, 2011). Additionally, the folder root (Courses, Organizations, etc.) permissions should be checked to insure that only "read" access is enabled. See Appendix D.

The Content System includes settings to limit bandwidth to protect from Denial of Service attacks. The Content system has five top level folders: `users`, `courses`, `orgs`, `institution`, and `library`. The bandwidth settings apply to all of these folders and are effective for reading and writing. Bandwidth quotas can be configured for each folder along with a time period (1 hours = 1000MB for example). See Appendix D.

Bb Learn has a mechanism to limit uploads by file type. This is configurable and supports blacklist and whitelist models (Tan & Alexander, 2011). See Appendix D.

David C. Lyon, lyondc@gmail.com

6.3.9. SIS Integration

Bb Learn can be integrated with a Student Information System such as PeopleSoft or Banner. Bb Learn provides two methods for doing so – snapshot and event driven. A snapshot tool is provided to enable batch integration. In this method, a feed file is used to synchronize with Bb Learn tables that contain users, courses, etc. There is also the Bb Learn Event Drive APIs to push data from the SIS into the same Bb Learn tables. Integration should be backed by a plan that includes security as a consideration ("Blackboardlearn Release 9.1 Advanced Integration and Data Management Guide," 2010). If implementing Event Driven APIs, consider SSL and if snapshot is used, make sure the data is pulled and stored securely.

Each Bb Learn user will require a unique internal key but something other than a Social Security number. The other user information is configurable. Ideally, one would want minimal user data stored in Bb Learn. One should avoid keeping information such as birth date, phone, and other sensitive information in Bb Learn. It is also important to configure what users can/and cannot change. This can be done from the *System Admin* panel. See Appendix D. Allowing a user to change something in Bb Learn that is also stored in a central Identity Management database would not be a good thing and may not be good thing period.

6.4. Building Blocks and Web Services

Building Blocks are a way to extend Bb Learn. Building blocks run in the Blackboard container (Fontaine, 2009). Klophaus (2009) describes a building block as a "Mini-application that runs within Blackboard Learn" (p. 15). A Building Block is a Java servlet that uses Java-style permissions (Klophaus, 2009). An example might be a Building Block to list instructor courses conditionally, based on what term they were taught.

Building blocks can exchange information with an external system, utilizing web services (Klophaus, 2009). They can also access authentication/authorization information, user and course data, content, and even system administration data ("Overview of Building Blocks and PowerLinks," 2011). Due to this ability, and the fact that a Building Block can communicate externally, or access the underlying O/S, security aspects of a Building Block are very important. It is very important that a Building Bock be fully investigated and vetted with the Information Security Officer or equivalent before deployment within Bb Learn. A block's permission settings can be found in the .WAR file's `WEB-INF` folder in the file `bb-manifest.xml`. Something like the following might be of concern:

```
<permission name="../../../../blackboard/logs/-"
  type="java.io.FilePermission"
  actions="read,write,delete"/>
```

This appears to grant read/write/delete to the underlying file system logs folder, but using a relative path. It is useful to examine other configuration files found in `content/vi/bb_bb60/plugins`. One or more of these files may get created as a result of configuring the building block from the *System Admin* panel. In addition, since the Building Blocks API can change with Bb Learn updates, building blocks may break. This should also be considered before a deployment decision is made.

The list of Building Blocks is accessed from the *System Admin* panel under *Building Blocks*. From there, one can access the list of Building Blocks from *Installed Tools.* The *Global Settings* button leads to a critical configuration setting that can be set to prevent Building Blocks from creating custom database objects. This is especially important for those concerned about database objects being created by Building Blocks. As of Bb Learn 9.1 Service Pack 1, a restriction that required "signed" building blocks, distributed by Blackboard Incorporated, has been lifted ("Creating Database Objects with Building Blocks," 2011). Any Bb Learn delivered building blocks that are not going to be used should be completely disabled. See Appendix D.

As previously stated, Building Blocks can use web services to exchange data. Bb Learn has a few web services configured but by default, they are not enabled. If they are enabled, they should have SSL turned on. There are web services for announcements, calendar, content, and others. There are many configuration options to consider if a Web Service is going to be enabled. These include discoverability, logging, IP filtering, and permissions. See Appendix D {web services book info after this}.

7. Blackboard Learn Monitoring and Maintenance

Taking steps to harden the O/S, services and application are critical. Without a plan for monitoring, maintenance, and change management, we cannot expect to provide confidentially, integrity, and availability. Monitoring the O/S is a starting point. Spikes in resource consumption could indicate a Denial of Service attack or SSH warnings could point to a brute force attack. The Bb Learn application should also be monitored closely. Apache logs can be used to identify attackers or misconfigured clients. Bb Learn logs information in various places, including the database. Monitoring those for possible abuse is valuable and necessary.

Some mechanisms, like users, roles and permissions, use database tables to maintain configuration. Monitoring for changes in those tables is valuable. Malicious as well as unintended change can be tracked and policy violations can also be detected. Automated and manual auditing of permissions, roles, and system admin accounts (those with highest Bb privileges) should be part of any security practice.

Tracking change at the file system level using a host based intrusion detection utility such as AIDE can provide an assurance that the system is stable and has not been altered and can also notify when something has changed. The change could be either due to a successful exploitation or change outside of the change control process. Either is not good.

7.1. Periodic Maintenance

Ristic (2005) states that, "Maintaining a server after it has been installed is the most important thing for you to do." (p. 227) Regular maintenance is an important way to keep a system in a stable state by insuring the latest security patches are installed and auditing security. Periodic maintenance would normally be a time to make changes that include applying the latest security patches. For security patches, this may require unplanned downtime or scheduled during a weekly maintenance window. Monitoring security bulletins from sources like the Blackboard Support site - Behind the Blackboard and SANS "@RISK: The Consensus Security Vulnerability Alert" is vital to staying on top of issues and patches. Regular vulnerability scanning can also identify vulnerabilities that might be addressed with patches or mitigated with a work-around. Regular maintenance and monitoring should also include development and test systems, especially those that have a copy of production data. See Appendix G.

7.2. Watching Logs

Peikari, C. and Chuvakin (2004) state that, "Simply looking at logs does not constitute analysis, as it rarely yields anything other than an intense sense of boredom and desperation." Logs not only need analysis, but events need to be correlated with other events in that they might be related (Peikari, C. & Chuvakin, 2004). Errors in the `bb-services-log.txt` can be correlated with hits in the Apache logs.

Log files can offer clues to attempted attacks. Take for example, the Apache `error_log`. If we see "`segmentation fault`", it may indicate a buffer overflow attack. One problem with real-time monitoring is the frequency of alerts. If monitoring tools are configured such that there are too many false positives, those responsible for reacting may lose sight of true positives. Periodic monitoring can be implemented to minimize "noise", remove redundant or unimportant information, sort information, and aggregate/sort duplicate hits in descending order. Logs should also be monitored for SQL injection attempts, cross-site scripting attacks, and command execution (Ristic, 2005).

The increase in the number of servers to monitor adds complexity and can lead to difficulty in processing logging information. Centralized logging mitigates this issue and improves security by keeping a copy of logs away from individual servers. In the event that a server is compromised, this prevents the altering of logs (Ristic, 2005). See Appendix F.

7.3. Tracking Bb Learn User Activity

In the context of Bb Learn, the issue may arise where a student claims to have submitted an assignment or taken a test but the instructor cannot confirm it. In this case, a history Bb Learn user activity is required. The Bb access log (`logs/tomcat/bb-access-log.YYYY-MM-DD.txt`) is very useful for tracking user activity. It contains the unique pk1 found in the `USERS` table which enables a user ID to `bb-access-log` row mapping ("Tracking User Activity in Webserver Logs," 2011). This log should also be regularly monitored for malicious activity. Additionally, we want to carefully watch accounts that are used for support. The following questions need to be addressed: are they following policy? Are they logging in from a lab? Are they logging in over the guest WiFi? In addition, we should also check the `logs/bb-session-log.txt` for session changes for any support accounts (Tan, 2011). See Appendix F.

7.4. Scanning

Scanning should be a regular part of Bb Learn maintenance activities. Several free tools exist, including the `nmap` port scanner. These help identify open ports and vulnerabilities. Detailed configuration and use is beyond the scope of this paper. There are some basic scans that can be run to get a port and vulnerability report. In addition, there are other simple tests to run to identify misconfiguration. Scanning should not be done to a production system and should only be done with the permission of the information security officer or equivalent. See Appendix G.

8. Conclusions

Through the process of writing this paper, it became apparent that running Bb Learn securely is a complex task due to the layers of software and the inter-related countermeasures that must be put into place. We learned that Bb Learn has weaknesses and cannot be installed with default settings. This puts the responsibility squarely on the organization deploying Bb Learn, and ultimately onto the system administrator tasked with doing so. With the technical information garnered and practical steps provided, this paper may serve as a practical guide in securing Bb Learn and provide the desire to dig deeper. Fortunately, Blackboard Incorporated has been working to incorporate countermeasures. In addition, they have addressed vulnerability reports and provided clients with pertinent information. They have also provided a wealth of documentation that is updated often. These resources should be utilized on an ongoing basis to get the latest info on security. The BBADMIN-L list is also an invaluable resource. In short, it is about Defense-in-Depth. Secure the various layers and do not neglect any layer.

9. References

AIDE (2011). Retrieved August 24th, 2011 from http://aide.sourceforge.net/

AS-152479-A Vulnerabilities in Blackboard Learn Could Allow Tampering (2011). Retrieved October 10th, 2011 from Behind the Blackboard Knowledge Base site: http://kb.blackboard.com/display/KB/AS-152479-A+Vulnerabilities+In+Blackboard+Learn+Could+Allow+Tampering

AS-152479-C Vulnerabilities in Blackboard Learn Could Allow Elevation of Privilege (2010). Retrieved September 16th, 2011 from Behind the Blackboard site: http://kb.blackboard.com/display/KB/AS-152479-C+Vulnerabilities+In+Blackboard+Learn+Could+Allow+Elevation+Of+Privilege

AS-158601 – Cross-site Scripting Vulnerability in Blackboard Learn Discussion Board Could Allow Elevation of Privilege (2011). Retrieved September 13th, 2011 from Blackboard Behind the Blackboard site: http://kb.blackboard.com/display/KB/AS-158601+-+Cross-site+Scripting+Vulnerability+in+Blackboard+Learn+Discussion+Board+Could+Allow+Elevation+of+Privilege

Alcorn, B. (2005). Web Services and Blackboard. Retrieved July 13, 2011 from Blackboard site: http://library.blackboard.com/docs/uc05/WebServices&Blackboard.ppt

Ashman, D. (2006). Security and Authentication with Blackboard Building Blocks. Retrieved September 13th, 2011 from Blackboard edugarage site: http://www.edugarage.com/download/attachments/13271060/b2_2006_security.ppt

Benedict, P. (2002). Lessons Learned in Securing Blackboard. Retrieved August 25th, 2011 from the SANS Reading Room site: http://www.sans.org/reading_room/whitepapers/casestudies/lessons-learned-securing-blackboard_773

Bitpushr (2011). Using Blackboard with Apache 2.2x. Retrieved September 14th, 2011 from Bitpushr's Blog site: http://bitpushr.wordpress.com/2011/08/01/using-blackboard-with-apache-2-2-x/

Blackboardlearn Release 9.0 Administrator Guide (2009). Retrieved August 1, 2011 from Blackboard Support web site: https://behind.blackboard.com/s/sysadminas/refcenter/docs/

Blackboardlearn Release 9.1Advanced Integration and Data Management Guide (2010). Retrieved October 19th from Blackboard Support web site: https://behind.blackboard.com/s/sysadminas/refcenter/docs/

Blackboardlearn Release 9.1 Installation Guide (2010). Retrieved August 1, 2011 from Blackboard Support web site: https://behind.blackboard.com/s/sysadminas/refcenter/docs/

Blackboard Security Advisory - Vulnerability in Blackboard Learn Avatars Could Allow Information Exposure (2011). Retrieved September 13th, 2011 from Blackboard Behind the Blackboard site: http://kb.blackboard.com/display/KB/Blackboard+Security+Advisory+-+Vulnerability+in+Blackboard+Learn+Avatars+Could+Allow+Information+Exposure

Blackboard learn Release 9.1 Service Pack 5 Release Notes (2011). Retrieved July 28, 2011 from Blackboard Support web site: https://behind.blackboard.com/s/sysadminas/refcenter/docs/

Blackboardlearn Release 9.1 Server Administration Guide (2010). Retrieved August 1st, 2011 from Blackboard Support web site: https://behind.blackboard.com/s/sysadminas/refcenter/docs/

Chopra, V., Li, S. & Genender, J. (2007). Professional Apache Tomcat 6. Indianapolis, IN: Wiley Publishing Inc.

Creating Database Objects with Building Blocks (2011). Retrieved October 18th, 2011 from the edugarage site: http://www.edugarage.com/display/BBDN/Creating+Database+Objects+with+Building+Blocks

Denial of Service Attacks (2011). Retrieved October 10th, 2011 from Blackboard Behind the Blackboard site: http://kb.blackboard.com/display/KB/Denial+of+Service+attacks

Eychenne, H (2008) IPTABLES. Retrieved September 21st, 2011 from NETFILTER.ORG site: http://ipset.netfilter.org/iptables.man.html

Feldman, S. (2011). Scaling Blackboard Learn for High Performance and Delivery. Retrieved August 31, 2011 from Connections Blackboard site: http://connections.blackboard.com/files/e6de0e1afd/Scaling_Blackboard_Learn_for_High_Performance_and_Availability.pptx

Fontaine, J. (2009). Blackboard Learn, Release 9 Technology Architecture. Retrieved August 31, 2011 from edugarage site: http://www.edugarage.com/download/attachments/42369042/24022_R9.Tech.Archit_Fontainex.pdf

Guide to the Secure Configuration of Red Hat Enterprise Linux 5 (2011). Revision 4.1. Retrieved September 21st, 2011 from NSA.GOV site: http://www.nsa.gov/ia/_files/os/redhat/rhel5-guide-i731.pdf

Hazlewood, V. (2006). Defense-In-Depth, An Information Assurance Strategy for the Enterprise. Retrieved August 1, 2011 from San Diego Supercomputer Center site: http://www.sdsc.edu/~victor/DefenseInDepthWhitePaper.pdf

How To Configure ActiveMQ for 9.1 (2011). Retrieved October 27th, 2011 from Behind the Blackboard site: http://kb.blackboard.com/display/KB/How+To+Configure+ActiveMQ+for+9.1

Klophaus, R. (2009). Getting Started With Blackboard Building Blocks (Part 1). Retrieved September 13th, 2011 from Blackboard edugarage site: http://www.edugarage.com/download/attachments/42369042/24017_Intro.B2.Part.I_Klophaus.pdf

LRNSI-2284 - Vulnerabilities in Blackboard Learn Could Allow Elevation of Privilege (2011). Retrieved September 16th, 2011 from Behind the Blackboard site: http://kb.blackboard.com/display/KB/LRNSI-2284+-+Vulnerabilities+in+Blackboard+Learn+Could+Allow+Elevation+of+Privilege

Overview of Building Blocks and PowerLinks (2011). Retrieved September 13th, from Blackboard edugarage site: http://www.edugarage.com/display/BBDN/Overview+of+Building+Blocks+and+PowerLinks

David C. Lyon, lyondc@gmail.com

Peikari, C. & Chuvakin, A. (2004). Security Warrior. Sebastopol, CA: O'Reilly Media, Inc.

Peikari, C. & Fogie, S. (2003). Writing a security policy. Retrieved September 15th, 2011 from SearchSecurity.com site: http://searchsecurity.techtarget.com/tip/Writing-a-security-policy

Potts, S., & Kopack, M. (2003). SAMS Teach Yourself Web Services in 24 Hours. Indianapolis, IN: SAMS.

Prins, M. & Abma, J. (2010). Security research Blackboard Academic Suite. Retrieved July 6, 2011 from Online24 site: http://www.online24.nl/downloads/Security%20research%20Blackboard%20Academic%20Suite.pdf

Puschitz, W. (2007). Securing and Hardening Red Hat Linux Production Systems – A Practical Guide to Basic Linux Security in Production Enterprise Environments. Retrieved September 21st, 2011 from PUSCHITZ.COM site: http://www.puschitz.com/SecuringLinux.shtml

Ristic, I. (2005). Apache Security. Sebastopol, CA: O'Reilly Media, Inc.

Scarfone, K., Jansen, W. & Miles, T. (2008). NIST Guide to General Server Security, 3-5, 12, 18, 14, 13, 26, 48. Retrieved on September 1st, 2011 from NIST Computer Security Division site: http://csrc.nist.gov/publications/nistpubs/800-123/SP800-123.pdf

Secure LDAP Authentication (2011). Retrieved September 13th, 2011 from Blackboard Behind the Blackboard site: http://kb.blackboard.com/display/KB/Secure+LDAP+Authentication

Security Communication Regarding Course Archive (2011). Retrieved September 13th, 2011 from Blackboard Behind the Blackboard Knowledge site: http://kb.blackboard.com/display/KB/Security+Communication+Regarding+Course+Archive

Setting Up LDAP Authentication Properties (2011). Retrieved October 7th, 2011 from Blackboard Library site: http://library.blackboard.com/ref/df5b20ed-ce8d-4428-a595-

David C. Lyon, lyondc@gmail.com

a0091b23dda3/Content/_admin_server_authentication/authentication_setup_ldap.htm

Setting Up SSL Choice (2011). Retrieved October 7th, 2011 from Blackboard Library site: http://library.blackboard.com/ref/df5b20ed-ce8d-4428-a595-a0091b23dda3/Content/_admin_server_ssl/ssl_choice.htm

Sintes, T. (2002). App Server, Web server: What's the difference? Retrieved August 1, 2011 from JavaWorld web site: http://www.javaworld.com/javaworld/javaqa/2002-08/01-qa-0823-appvswebserver.html

Tan, S. & Alexander, C. (2011). LRNSI-2284 Security Advisory Overview. Retrieved October 26, 2011 from Blackboard Behind the Blackboard site: http://kb.blackboard.com/download/attachments/84246532/LRNSI-2284_Advisory_Overview.pdf

Tan, S. (October, 2011). Blackboard Learn Security – Secure Configuration and Security Program Overview [Webinar]. Retrieved October 27th, 2011 from Blackboard Behind the Blackboard site: http://kb.blackboard.com/download/attachments/95780866/Blackboard_Learn_Security_Secure_Configuration_Security_Overview_Self-Hosted.zip?version=1&modificationDate=1319407736947 .

The SessionInvalidation Task (2011). Retrieved October 18th, 2011 from Behind the Blackboard Knowledge Base site: http://kb.blackboard.com/display/KB/The+SessionInvalidation+Task

Tracking User Activity in Webserver Logs (2011). Retrieved October 24th, 2011 from Behind the Blackboard site: http://kb.blackboard.com/display/KB/Tracking%20User%20Activity%20in%20the%20Webserver

Transform the Teaching and Learning Experience (2011). Retrieved September 1, 2011 from Blackboard web site: http://www.blackboard.com/Platforms/Collaborate/Products/Blackboard-Collaborate/Web-Conferencing.aspx

U.S. Privacy Definitions and Regulations Relevant to Blackboard (2011). Retrieved November 7th, 2011 from Blackboard library site: http://library.blackboard.com/ref/df5b20ed-ce8d-4428-a595-a0091b23dda3/Content/_admin_app_system/admin_app_privacy_us_regulations.htm

Using the Cross-site Scripting Security Control (2011). Retrieved October 10th, 2011 from Blackboard Behind the Blackboard Knowledge site: http://kb.blackboard.com/display/KB/Using+the+Cross-site+Scripting+Security+Control

Vulnerability Management Commitment and Disclosure Policy for Blackboard Learn (2011). Retrieved September 15th, 2011 from Blackboard site: http://www.blackboard.com/Footer/Security-Policy.aspx

West, T. (2010). Part 1 – Blackboard Performance Tuning: An Iterative Approach. Retrieved September 1, 2011 from Blackboard Code Monkey blog: http://blackboardcodemonkey.wordpress.com/2010/12/09/part-1-blackboard-performance-tuning-an-iterative-approach/

Zonneveld, K. (2007). Block brute force attacks with iptables. Retrieved October 18th, 2011 from Kevin van Zonneveld blog site: http://kevin.vanzonneveld.net/techblog/article/block_brute_force_attacks_with_iptables/

Appendix A – Server Hardening Checklist

The NSA (National Security Agency) publishes a "hardening Tips" document that provides a good high level starting point for server hardening. It is available at http://www.nsa.gov/ia/_files/factsheets/rhel5-pamphlet-i731.pdf.

○ Establish Server Hardening policy/procedure

○ Establish Support Account policy/procedure (Server and Bb Learn)

○ Establish Data Retention policy/procedure (Server and Bb Learn)

○ Install minimal O/S with latest patches

○ Do not install X Windows unless necessary (not required for Bb Learn)

○ Add utilities (if not present):

```
lsof, iptables, perl, tcpdump, X11, wget, bash, tar
```

○ Create separate partitions

The Guide to the Secure Configuration of Red Hat Enterprise Linux 5 (2011) recommends separate partitions for `/tmp`, `/home`, `/var`, and `/var/log`.

○ Remove unneeded services

Puschitz (2007) suggests:

- ☐ Determine listening services with `netstat -tulp`
- ☐ Determine services starting using `chkconfig -list | grep on`
- ☐ Disable using `chkconfig -del`
- ☐ Remove unneeded entries from `/etc/inittab`

○ Remove `setuid` and `setgid` bits

David C. Lyon, lyondc@gmail.com

The Guide to the Secure Configuration of Red hat Enterprise Linux 5 (2011) suggests removing the `setuid` and `setgid` bits from files if possible. List them using this command:

```
find /usr -xdev \( -perm -4000 -o -perm -2000 \) \
  -type  f  -print
```

An example of where you would not want to do this is for `/bin/ping6` – if you are using IPv6. The guide has more information.

- ○ Adjust kernel parameters

The Guide to the Secure Configuration of Red Hat Enterprise Linux 5, (2011) suggests the following should be set in `/etc/sysctl.conf`:

```
net.ipv4.ip_forward=0
net.ipv4.conf.all.send_redirects=0
net.ipv4.conf.default.send_redirects=0
fs.suid_dumpable  =  0            # no dump
kernel.exec-shield  =  1          # Execution control
kernel.randomize_va_space  =  1    # Execution control
```

Other parameters to consider including (also from the guide):

```
net.ipv4.conf.all.accept_source_route  =  0
net.ipv4.conf.all.accept_redirects  =  0
net.ipv4.conf.all.secure_redirects  =  0
net.ipv4.conf.all.log_martians  =  1
net.ipv4.conf.default.accept_source_route  =  0
net.ipv4.conf.default.accept_redirects  =  0
net.ipv4.conf.default.secure_redirects  =  0
net.ipv4.icmp_echo_ignore_broadcasts  =  1
net.ipv4.icmp_ignore_bogus_error_messages  =  1
net.ipv4.tcp_syncookies  =  1
net.ipv4.conf.all.rp_filter  =  1
net.ipv4.conf.default.rp_filter  =  1
```

The guide also suggests disabling IPv6 if not needed (see instructions contained therein).

- ○ Create local account to run Bb with strong password and schedule for changing

○ Configure Bb shared area

Mount via NFS and restrict access to local account. Puschitz (2007) suggests restricting access to the NFS server and configuring it to use TCP.

○ Configure iptables firewall

☐ Configure basic rules

The Guide to the Secure Configuration of Red Hat Enterprise Linux 5 (2011) suggests the following:

Set the default policy to "DROP" by editing `/etc/sysconfig/iptables` and adding:

```
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
```

Use `iplimit` and `recent` to protect against SYN floods (use caution). Enable "TCP SYN cookies" feature by editing `/etc/sysctl.conf` and adding the following:

```
net.ipv4.tcp_syncookies  =  1
```

Van Zonneveld (2007) suggests the following rules to control brute force attacks:

```
iptables -A INPUT -i eth0 -p tcp --dport 22 \
  -m state --state NEW -m recent --set --name SSH
iptables -A INPUT -i eth0 -p tcp --dport 22 \
  -m state --state NEW -m recent --update \
  --seconds 60 --hitcount 8 --rttl --name SSH \
  -j DROP
```

The above allows eight connections per minute.

☐ Configure rules for the Bb Learn application

```
-A BI -p tcp -m tcp -dport 80 \
  --tcp-flags SYN,ACK,FIN,RST SYN -j ACCEPT
```

The above assumes you have created a rule to append to INPUT chain with a jump (-j) to the BI chain (rules). The above rule allows all incoming tcp port 80 traffic with the TCP SYN flag set and the others (ACK,FIN,RST) unset (Eychenne, 2008). Similarly, you need a rule for TCP port 443 (SSL).

For connection to the database (hosted on a separate server), you will need an outbound rule:

```
-A BO -d 192.168.1.5 -p tcp -m tcp -dport 1521 -tcp-flags \
  SYN,ACK,FIN,RST SYN -j ACCEPT
```

Make sure the database and NFS servers are isolated from the Internet. More rules are needed for NFS, and other services that a building block or monitoring service might be connecting to.

If remote management is required (via SSH), make sure iptables is configured to only allow access from specific subnets or hosts. Require the use of VPN for other remote access.

- Configure Remote Management

If using SSH for remote access, Puschitz (2007) suggests the following changes to `sshd_config` to restrict access to SSH.

```
PermitRootLogin no
UsePrivilegeSeparation yes
Protocol 2
StrictModes yes
```

- Limit local access to servers to system administrators and Bb Learn admins only

- Secure Accounts

Puschitz (2008) suggests removing unneeded accounts (vendor, application, etc). The `find` command can be used to locate files owned by any account.

The Guide to the Secure Configuration of Red hat Enterprise Linux 5 (2011) suggests the following:

Restrict the interfaces that `root` login may occur from by editing `/etc/securetty`. Restrict the accounts that may "su" to `root` by maintaining the `wheel` group in `/etc/group`. Prevent password guessing of the `root` account password by adding the following line to `/etc/pam.d/su`:

```
    auth required pam_wheel.so use_uid
```

☐ Consider `sudo`

Bb Learn needs `root` to start and to update configuration. Consider enabling `sudo` rather than providing the `root` password to Bb Learn administrators.

☐ Lock accounts and disable shell

The Guide to the Secure Configuration of Red Hat Enterprise Linux 5 (2011) suggests locking and disabling the shell for accounts that do not require it.

```
    usermod –L user-id
    usermod –s /sbin/nologin user-id
```

☐ Configure a Group

The Guide to the Secure Configuration of Red Hat Enterprise Linux 5 (2011) suggests utilizing a group.

Create the `bbaccount` group with only `bbaccount` as member. Add the group in the file: `/etc/group`.

○ Central Authentication and Network Time

The Guide to the Secure Configuration of Red Hat Enterprise Linux 5 (2011) suggests implementing central authentication with encryption and establishing network time synchronization. The Network Time Protocol can be configured for this (RFC 1305).

Appendix B – Apache Hardening Checklist

○ Identify modules that are compiled into Bb Learn supplied Apache

```
apps/httpd/bin/httpd -l
```

See section 6.1.1.

○ Configure a group in the Apache configuration

Ristic (2005) suggests creating a group for the Apache account. The following lines should be in `apps/httpd/conf/httpd.conf.bb`:

```
     user bbaccount
     group bbaccount
```

○ Configure access to Apache binaries

Ristic (2005) recommends locking down the Apache binaries. The account that is running Bb Learn (`bbaccount`) shouldn't have "write" access to these binaries.

○ Verify file access

Ristic (2005) suggests the following:

```
<Directory />
  Order Deny,Allow
  Deny from all
</Directory>
<Directory /var/www/htdocs>
  Order Allow,Deny
  Allow from all
</Directory>
```

However, Bb Learn is handing off requests to Tomcat. Bb Learn does define both directory stanzas but differently. In Bb Learn, Apache is handing off most requests to Tomcat.

○ Remove `bin` folder in configuration

Edit `apps/httpd/conf/httpd.conf.bb` and comment the following:

```
#ScriptAlias /bin/ "/usr/local/blackboard/bin/"
```

○ Configure log rotation

Edit `apps/httpd/conf/httpd.conf.bb` and add the following:

```
CustomLog "|apps/httpd/bin/rotatelogs
    logs/httpd/access_log_%Y%m%d 86400" common
```

Note that the full path is required (where Bb Learn was installed) in front of `apps` and `logs`. The log is configured to cycle 86400 seconds (24 hours) from the time Bb Learn (Apache) was started. Repeat for `error_log` and `mod_jk`.

○ Mask the Apache version

Ristic (2005) suggests the following:

Add the following line to `apps/httpd/conf/httpd.conf.bb`

```
ServerTokens ProductOnly
```

○ Guard against unintentional file disclosure

Ristic (2005) suggests the following:

In `apps/httpd/conf/httpd.conf.bb`, consider the following:

```
<Directory />
   AllowOverride none

   <FilesMatch "(^\.ht|~$|\.bak|\.BAK$)">
      Order Allow,Deny
      Deny from all
   </FilesMatch>
```

David C. Lyon, lyondc@gmail.com

○ Configure Apache to reject unsafe URLs

Uncomment the following in `apps/httpd/conf/httpd.conf.bb`:

```
JkOptions +RejectUnsafeURI
```

Set the following option in `config/bb-config.properties`:

```
bbconfig.jk_connector.reject_unsafe=true
```

○ Configure robots.txt to prevent web crawlers from indexing the site

Create `docs/robots.txt` and add the following lines:

```
User-agent: *
Disallow: /
```

○ Blackboard Apache recommendations

LRNSI-2284 - Vulnerabilities in Blackboard Learn Could Allow Elevation of Privilege (2011) suggests the following:

☐ Add "`TraceEnable off`" to the following files:

```
apps/httpd/conf/httpd.conf
apps/httpd/conf/httpd.conf.bb
apps/httpd.conf.default
```

☐ Harden the SSL protocol

The recommended change includes modifying the following directives found in `apps/httpd/conf/ssl.conf.bb`:

```
SSLProtocol
SSLCipherSuite
```

The recommendation is to set them to the following:

```
SSLProtocol -ALL +SSLv3 +TLSv1
SSLCipherSuite ALL:!aNULL:!ADH:!eNULL:
    !LOW:!EXP:RC4+RSA:+HIGH:!MEDIUM:!SSLv2
```

`SSLCipherSuite` should all be on one line.

- Configuring SSL

  The Blackboardlearn Release 9.1 Server Administration Guide (2010) has detailed instructions. A summary is provided here.

  - ☐ Generate a private key using `openssl`
  - ☐ Generate a CSR (Certificate Signing Request) using `openssl`
  - ☐ Submit the CSR to a Certifying Authority (CA)
  - ☐ Edit `apps/httpd/conf/httpd.conf.bb` and add "`Include conf/ssl.conf`"
  - ☐ Once in possession of the certificate from the CA, specify the location of the key and certificate files by changing the following in `config/bb-config.properties`:

```
SSLCertificateFile /path/server.crt
SSLCertificateKeyFile /path/server.key
```

- Rewrite Rules

  Rewrite rules should be placed in `httpd.conf.bb` and `ssl.conf` (if used). In `httpd.conf.bb`, they would go under `<IfModule mod_proxy.c>`.

  - ☐ Mitigate "Bandwidth Stealing"

  Ristic (2005) suggests the following rewrite rules (unwrap them):

```
RewriteCond %{HTTP_REFERRER} !^$
RewriteCond %{HTTP_REFERRER}
  !^http://www.mybb.com [nocase]
RewriteRule (\.gif|\.jpg|.\png|\.swf)$ $0
  [forbidden]
```

The first accounts for an empty referrer when a direct URL is used. The second allows for users coming directly from the site, while the third prevents images from being hot linked.

David C. Lyon, lyondc@gmail.com

☐ Block specific requests

```
RewriteCond %{SCRIPT_FILENAME} "(\.php)$"
RewriteRule .* - [forbidden]
```

The above example will block access to `php`.

Appendix C – Tomcat Hardening Checklist

○ Remove default applications

Chopra, et al. (2007) recommends removing the default applications. Remove `ROOT`, `manager`, and `host-manager` from `apps/tomcat/webapps/`.

○ Disable Directory listings, Invoker, SSI, and CGI

Chopra, et al. (2007) suggests editing `apps/tomcat/conf/web.xml` to make sure that the CGI and SSI servlets, the SSI filter are not enabled (commented), ideally removed. These look like:

```
<servlet-name>ssi</servlet-name>
<servlet-name>cgi</servlet-name>
<servlet-name>invoker</servlet-name>
```

Also make sure directory listings are turned off:

```
<param-name>listings</param-name>
<param-value>false</param-value>
```

Make sure that `apps/tomcat/conf/context.xml` does not have a `privileged='true'` attribute.

○ Verify `SecurityManager` is running

This is confirmed by running the following command:

```
% ps -fu bbaccount | grep -v grep | grep -oh \
  --color=auto java.security.manager
  java.security.manager
```

Note that the Java Security Manager is running (result in red).

Bb Learn is started by running:

David C. Lyon, lyondc@gmail.com

```
tools/admin/ServiceController.sh services.start
```

○ Lock down Tomcat application permissions

Chopra, et al. (2007) suggests:

- ☐ Configure the `JAVA_HOME` folder to allow `bbacount` to have read/execute only
- ☐ Lock down the `apps/tomcat` folder appropriately, granting ownership to the `bbacount` and limited access by others

David C. Lyon, lyondc@gmail.com

Appendix D – Blackboard Learn Hardening Checklist

This appendix contains SQL queries against the Oracle database. The Perl DBI database interface can be used in Perl scripts to run queries. See http://dbi.perl.org/.

○ Configure LDAP and LDAPS

Setting Up LDAP Authentication Properties (2011) contains information on configuring LDAP. Some key settings are presented here.

Edit the `config/authentication.properties` file. Good security depends on the correct configuration settings. See below:

`auth.type.ldap.impl`	Unless you have a custom module for LDAP, this should be left as the default provider.
`auth.type.ldap.use_challenge`	A value of "false" will mean base 64 encryption (as opposed to MD5). Blackboard states that using base 64 with SSL is the best approach.
`auth.type.ldap.error_fallback_to_bb`	A value of "true" means that a failed authentication will fall back to Bb Learn database if there is an error when attempting to use a directory server.
`auth.type.ldap.user_not_found_fallback_to_bb`	A value of "true" means that authentication will fall back to the Bb Learn database if the user is not found in the directory.
`auth.type.ldap.num_servers`	This is the number of LDAP servers.
`auth.type.ldap.server_ssl.1`	This should be set to "true" so that communication between

	the Bb Learn app server and LDAP will be encrypted via SSL
`auth.type.ldap.base_search_fdn.1`	This is the search starting point used when searching for a user match. It might be something like "`dc=mountholly,dc=edu`".
`auth.type.ldap.server_url.1`	This is the URL of the LDAP server with the SSL port appended. It would be something like: `ldap://ldap.mountholly.edu:636`.

○ Do not use the `administrator` account

Avoid using the `administrator` Bb Learn account unless absolutely necessary. Instead, assign roles to users who can be tracked in an identity management system.

○ Disable `root_admin` account

If not using virtual installations, disable the `root_admin` account from the Bb Admin panel.

○ Change the `integration` password (used by snapshot)

This can be done from the *Bb Admin* panel.

○ Configure Bb Learn for SSL

Setting Up SSL Choice (2011) specifies the following:

Log into a Bb Learn Administrator account
Click on "System Admin"
Click on "Security"
Click on "SSL Choice"
Select "SSL system-wide"
Click Submit

○ Configure SSL for the Collaboration Server

The Blackboard Learn Release 9.1 Server Administration Guide (1010) provides detailed instructions.

□ Create a keystore

Create a PKCS12 keystore using `openssl` (comes with Bb Learn). This is used to convert the main key and certificate into a PKCS12 keystore.

```
openssl pkcs12 -export -out ks.pkcs12 -in \
  /path/file.cert –inkey /path/file.key \
  –name tomat –Cafile myCA.crt –caname root
```

Provide a password when asked and remember what it is.

□ Configure Tomcat

Edit `config/bb-config.properties` and specify the following:

```
bbconfig.collabserver.keystore.filename=ks.pkcs12
bbconfig.collabserver.keystore.password=from above
bbconfig.collabserver.portnumber.ssl.default=8443
bbconfig.collabserver.keystore.type=PKCS12
```

The issue to consider here is the certificate to convert. Since the collaboration server runs on one application server, a DNS alias might exist pointing to that and specified in:

```
config/bb-config.properties
     bbconfig.collabserver.fullhostname.default=
```

If a certificate exists for that name, that would be the one to convert.

○ Disable Vulnerable Tools

AS-152479-A Vulnerabilities In Blackboard Learn Could Allow Tampering (2011) suggests disabling the following tools:

Address Book
Calendar

Link Checker Building Block (not supported in 9.1 Sp5 anyway)
Tasks

Log into a Bb Learn Administrator account
Click on "System Admin"
Click on "Tools" under "Tools and Utilities"
Disable the above tools

- Configuring the Cross Site Scripting Security Control

  - ☐ Using the Cross-site Scripting Security Control (2011) recommends the following:

Edit `config/bb-config.properties`

Use the following settings:

```
bbconfig.global.xss.filter=true.
bbconfig.global.xss.filter.mode=
   FilterDangerousHtml
```

This provides the highest level of detection. For the "mode", the `FilterAllHtml` is the second choice but only converts < and > to `&lt` and `&gt`.

Tan (2011) also recommends the following for file uploads:

```
bbconfig.fileupload.enable.xss=true
```

This enables the Global Cross-site Scripting Filter for file uploads.

  - ☐ Configure exceptions if needed

Edit `config/internal/bb-xss-global-filter-exceptions.txt`

It is just a matter of adding the webapp path to the file. A handful of paths are configured by default.

- Configure upload filtering

Tan & Alexander (2011) present the option of a whitelist for file uploading. The default is to allow all and filter individual types. To configure a whitelist, edit the file:

```
config/internal/bb-file-filter-configuration.properties
```

Set "`deny.all=true`" and configure types that should be allowed.

○ Restrict dynamic content

To restrict dynamic content, access the *System Admin* tab, then *Privileges* and remove this privilege from the roles that do not need it.

Log into a Bb Learn Administrator account
Click on "System Admin"
Under "Security", click on "Privileges"
Enter "Add/Edit trusted content with scripts" in the Search box and Go
Remove it from roles that do not need it and selectively assign it (click on a role to see the list of privileges)

○ Mitigating DoS Attacks

Denial of Service Attacks (2011) has some recommendations:

☐ Verify tuning parameters

Tuning parameters are set in `config/bb-config.properties.` The limit in the number of sessions is controlled by `bbconfig.unix.httpd.maxclients`. Set the Tomcat threads (`bbconfig.appserver.maxthreads)` the same as `bbconfig.unix.httpd.maxclients` (see section 6.2) Set the database pool size (`bbconfig.database.instance.maxpoolsize)` the same as the Tomcat threads (`bbconfig.appserver.maxthreads`).

☐ Verify Content System tuning parameters

The "Maximum Size of Connection Pool" should match `bbconfig.unix.httpd.maxclients.`

Log into a Bb Learn Administrator account

David C. Lyon, lyondc@gmail.com

Click on "System Admin"
Click on "Content Management"
Click on "Technical Settings"
Click on "Global Schema Settings"

☐ Set JVM arguments

The following JVM arguments may be used to trigger an ongoing collection first, which may avert the full garbage collection. In `config/bb-config.properties`, append the following to `bbconfig.jvm.options.extra.tomcat:`

```
+UseConcMarkSweepGC CMSFullGCsBeforeCompaction=1
```

☐ Use a 64 bit JVM if possible

☐ Use a load balancer configured to mark affected servers as "dead"

○ Privileges

Security Communication Regarding Course Archive (2011) recommends:

☐ Turn off challenge-response (Bb Learn 9.0)

Edit `config/authentications.properties`
Set `auth.type.rdbms.use_challenge=false`

Make sure you have SSL set to encrypt everything (system wide). See section 6.3.2.

☐ Disable Privileges

Remove the following privileges from the instructor and leader roles:

Create archives
Create users
Change passwords
Modify profiles
Manage enrollment

Log into a Bb Learn Administrator account
Click on "System Admin"
Under "Security", click on "Privileges"
Enter the following in the Search box and click "Go"

Course/Organization Control Panel (Packages and Utilities) >Archive Course/Organization

Click on "P" (instructor/leader role) and disable the following privileges:

1. Course/Organization Control Panel (Packages and Utilities) > Archive Course/Organization, including user passwords
2. Course/Organization Control Panel (Customization)> Settings > Enrollment Options
3. Course/Organization Control Panel (Users and Groups) > Batch Enroll Users
4. Course/Organization Control Panel (Users and Groups) > Enroll User
5. Course/Organization Control Panel (Users and Groups) > Remove Users from Course/Organization
6. Course/Organization Control Panel (Users and Groups) > Change User Information
7. Course/Organization Control Panel (Users and Groups) > Users > Edit Properties
8. Course/Organization Control Panel (Users and Groups) > Change User Password

Restrict the "Create Course" privilege

Click on "System Admin"
Under "Security", click on "Privileges"
Enter the following in the Search box and click "Go"
Administrator Panel (Courses) > Courses > Create Course

Grant this permission to *System Administrators* and optionally, *Course Administrators.*

- Secure Built-In Roles

If using Built-In roles such as "Course Admin" make sure unwanted privileges are not assigned.

Click on "System Admin"
Under "Security", click on "Privileges"
Enter the privileges you do not want assigned in the Search box and click "Go"

For example:

Administrator Panel (Users) > Users > Batch Create Users
Administrator Panel (Courses) > Courses > Delete Courses

David C. Lyon, lyondc@gmail.com

Course/Organization Control Panel (Packages and Utilities) > Archive Course/Organization, including user passwords

Grant these permissions (and others) to System Administrators only. Restrict from others.

○ Configure Session Invalidation

The SessionInvalidation Task (2011) provides details on configuring the task.

Edit the file: `config/bb-tasks.xml.bb`.

Find: `SessionInvalidtionTask`

Set the `invalid` property to a reasonable timeout and shorter for development/test systems. The default `period` is set to 1 hour. This could be lowered depending on performance impact. The `period` is the time between task invocations. The `chunkSize` should be set to the maximum number of sessions to process each time.

The resulting stanza might look like this:

```
<task-entry key="bb.session.invalidation"
…
      <property name="delay"   value="60000" />
      <property name="period"  value="1800000" />
      <property name="invalid" value="7200000" />
      <property name="chunkSize" value="4000" />
…
</task-entry>
```

The Bb Learn issue: LRN-1093 addresses a session invalidation bug resulting in only 1000 sessions per hour being invalidated. The patch is AS-160482 and should be requested through Bb Support.

○ Disable guest access

Log into a Bb Learn Administrator account
Click on "System Admin"
Under "Security", click on "Gateway Options"
Next to "Link to Course Catalog", click "Disable"

David C. Lyon, lyondc@gmail.com

Next to "Link to Account Creation", select "Disable"
Next to "Request Forgotten Password", select Disable
Next to "Allow Guest Access to the System", select "Disable"

Tan (2011) also lists other areas where guest access can be set, including the following:

System Admin > Course Settings > Course Tools
System Admin > Course Settings > Default Course Settings
System Admin > Organization Settings > Default Organization Settings

○ Disable Bb Learn tools such as Email (if warranted)

○ Configure Session Fingerprinting

Click on "System Admin"
Under "Security", click on "Session Fingerprint Settings"
Select "Yes" to enable session fingerprinting
Set the "Fingerprint value" (IP, User agent, or both)
Set "Filter IP addresses" if desired and if using IP as a fingerprint value
Select "Yes" to "Create new session when fingerprint changes"

○ Configure Grade History

Tan (2011) suggests the following:

Click on "System Admin"
Under "Course Settings"
Select "Grade Center Settings"
Set "Enable Grade History" to "Yes"
Set "Permit Instructors to Turn Grade History On and Off" to "No"
Set "Permit Instructors to Clear Grade History" to "No"

○ Configure ActiveMQ

How To Configure ActiveMQ for 9.1 (2011) specifies the following:

☐ Make sure NFS is functioning properly
☐ Edit hosts file

Add app01 and app02 (in the example below) for each application server to `/etc/hosts`.

☐ Edit configuration files

Edit `config/message-queue-service-config.xml.bb`

Change the “Configuration for Client” as in the following sample:

```
<client
brokerURL=”failover:nio://app01:61616,nio://app02:6161
6,nio:”
useAsyncSend=”true” optimizeAcknowledge=”true”
optimizedMessageDispatch=”true” dispatchAsync=”true”
copyMessageOnSend=”false”
disableTimeStampsByDefault=”true” initConnections=”5”
maxIdleConnections=”5” retryCount=”5”/>
```

Locate the Transport Connector section and change the URI to use 0.0.0.0:

```
<transportConnector name=”openwire”
uri=”nio://0.0.0.0:61616?keepAlive=true” />
```

○ Configure Content System Security

☐ Set Persistent Cookies to “No”

Log into a Bb Learn Administrator account
Click on “System Admin”
Under “Content Management”, click on “Technical Settings”
Click on “Authentication Settings”
Make sure “Persistent Cookies” is set to “off”

☐ Verify Content System root permissions

Click on “System Admin”
Under “Content Management”, click on “Manage Content”
For each of the root folders, make sure the permissions are only set to Read, for “All users with system accounts”

☐ Configure Bandwidth Settings

Click on “System Admin”
Under “Content Management”, click on “Technical Settings”
Click “Bandwidth Settings”
Change “Check Bandwidth Limits” to “On”
Specify the “Time Over Which to Apply Bandwidth Quota” (default 1 hour)
Click on “System Admin”
Under “Content Management”, click on “Technical Settings”
Click “Bandwidth Restrictions”
Select the Chevron and configure limits (MB) for each top level folder

☐ Configure Privacy Settings

Click on "System Admin"
Under "Content Management", click on "Content Management Settings"
Click on "Privacy Settings"
Select "Yes" for "Respect User Directory privacy setting"

○ SIS Integration

☐ Implement SSL if using Event Driven APIs

☐ Verify snapshot data stored securely if using Snapshot integration

☐ Verify SSNs not used for internal key

☐ Minimize user data stored in Bb Learn

☐ Lock down user modifiable items

Log into a Bb Learn Administrator account
Click on "System Admin"
Under "Users", click on "Customize User Information"
"Add Link" should not be selected and URL, Link Title, and Instructions blank
Uncheck all fields marked "Editable" and deselect any files that are not populated from Snapshot or Event Driven integration (Display heading).

○ Building Blocks

☐ Evaluate Building Blocks

For all Building Blocks, establish a vetting process that determines any host to host communication, permissions, personal data, etc.

Building blocks are packaged in a .WAR file. The extension can be changed to ".zip" and then unzipped to view the `bb-manifest.xml` file in the WEB-INF folder. Review the `<permissions>` section. Installed building blocks will be in the following folder:

```
content/vi/bb_bb60/plugins/BLOCK-NAME
```

The `bb-manifest.xml` file is in `webapp/WEB-INF.`

☐ Review the list of Building Blocks

Log into a Bb Learn Administrator account
Click on "System Admin"
Under "Building Blocks", click on "Installed Tools"
Under "Availability", specify "Inactive" for those that are not needed
Click "Global Settings"
Select "Prevent any Building Block from creating custom database objects"
Click "Submit"

☐ Configure Web Services

Click on "System Admin"
Under "Building Blocks", click on "Web Services"
Configure Availability of Web services, and SSL (enabled)
Click on "Manage Web Services" and enter an "Internal Secret"
Configure other options as needed

David C. Lyon, lyondc@gmail.com

Appendix E – Tomcat Architecture

The following is an overview of the Tomcat architecture:

- **Server**
  - ↳ **Service (Connector)**
    - ↳ **Engine**
      - ↳ Logger, Valve, Realm
      - **Host**
        - ↳ Logger, Value, Realm
        - **Context**
          - ↳ Realm, Value, Wrapper

It is useful to understand how a "hit" traverses through the nested Tomcat components. The diagram below illustrates this. For this example, we will use: https://blackboard.sample.edu/webapps/login/.

```
blackboard.sample.edu – Virtual Host in server.xml

     webapps – Context descriptor in server.xml

          login – Servlet
```

In the above example, a Context descriptor exists for `/webapps/login`. The login servlet runs to provide the login page (Chopra, et al., 2007).

David C. Lyon, lyondc@gmail.com

Appendix F – Monitoring

○ O/S Log Monitoring

☐ Consider Syslog-NG for centralized logging

The free solution is available at:
http://www.balabit.com/network-security/syslog-ng/opensource-logging-system/

☐ Implement log monitoring

Logwatch provides an open source solution available at: http://sourceforge.net/projects/logwatch/. The Guide to the Secure Configuration of Red Hat Enterprise Linux 5 (2011) suggests enabling `logwatch` and that it be run on a central server to simplify reporting. The signature database will need to be customized, requiring Perl programming skills. Be watchful of SSH brute force attacks. Tenshi is also worthy of consideration and is available at http://www.inversepath.com/tenshi_man.html.

☐ Monitor for large logs and system restarts

☐ Monitor for kernel parameter changes

The command: `sysctl` will list parameters.

☐ Consider Nagios

Nagios (www.nagios.org) can be used to monitor for O/S performance such as CPU which might indicate a DoS or Java issue. It can also report if a port is not listening or times out (such as 443) and notify if disk usage reaches a threshold. There are other plugins – for example, certificate expiration.

☐ Consider grsecurity

Ristic (2011) suggests the grsecurity kernel patch which provides more detailed logging – program executing, resource usage, etc. It's available at http://www.grsecurity.net.

David C. Lyon, lyondc@gmail.com

☐ Implement Process Accounting

Ristic (2011) suggests turning on process accounting with the `accton` command. Other packages may also be needed.

☐ Monitor database connections

```
lsof -i @db-host-name
```

○ Host Based Intrusion Detection

☐ Install AIDE or similar file integrity checker

AIDE creates a database of tracked files/folders based on configuration settings. After the initial creation, AIDE can be used to track integrity of those files/folders using a message digest algorithm such as Md5 (and others) as well as look for file mode changes (adate, cdate, etc.) (AIDE, 2011). AIDE will detect cases where the application deposits a file in the wrong folder or someone leaves a temporary file. It can also detect configuration file changes and changed binaries – so in this respect, it is very useful. The following folders (and file) containing configuration information should be closely watched for unwanted changes in files:

```
config/
apps/tomcat/bin
apps/httpd/bin
$JAVA_HOME/bin
apps/tomcat/conf
apps/httpd/conf
content/vi/bb_bb60/plugins/*/config/
content/vi/bb_bb60/plugins/*/webapps/WEB-INF/bb-
manifest.xml
```

Other folders containing custom configuration that is unique to an organization should also be monitored for changes.

David C. Lyon, lyondc@gmail.com

○ Bb Learn Monitoring

Monitoring of database tables should only be done with a separate database account that has query only access to the tables discussed below.

☐ Monitor Bb Response

As previously discussed, Nagios can be used to monitor the Bb Learn ports (80, 443, etc). A deeper check is also helpful. The following Perl sample code logs into Bb Learn and retrieves content. This is useful when Nagios (or other) reports that the ports are listening but beyond that, Bb is unresponsive. The code can also be modified to check for response of other features in Bb Learn.

```
sub DeepBbCheck {
  use LWP;
  use URI;
  use MIME::Base64;

  my $browser = LWP::UserAgent->new(keep_alive => 1,
                     requests_redirectable => [] );
  # Start a user agent
  #
  $browser->cookie_jar({});
  push @{ $browser->requests_redirectable }, 'POST';

  my @headers = ('User-Agent' => 'BB CHECK 1.0',
  'Accept-Language' => 'en-US',
  'Accept-Charset' => 'iso-8859-1,*,utf-8',
  'Accept-Encoding' => '',
  'Accept' => '*/*'
  );
  # Get the login page
  my $cont = ($browser->get
    ("https://bb.zz.edu/webapps/login/”,
      @headers))->content;
```

```
# Process page to fetch hidden HTML form variables
#
my %postvars;
my $response;

$postvars{login} = 'Log In';

# Find the form input
#
while ($cont =~
m{INPUT VALUE="(.*?)" NAME="(.*?)" TYPE="hidden"}gi)
                                                    {
  # Set the post variables
  #
  if ($2 ne 'password') {
    $postvars{$2} = $1;
  }
}
# Set the username
$postvars{user_id} = 'bbcheck';
$postvars{encoded_pw} = encode_base64("password");

# Send the credentials
#
$response = $browser->post
  ("https://mybb.zz.edu/webapps/login/",
   [%postvars], @headers);

# Check the results
if ($response->is_success && $response->content !~
 /\<INPUT TYPE\="text" NAME\="user_id"/i) {

  # Log in worked - request the main page
  $browser-> get("https://bb.zz.edu",
     @headers)->content;

  # Get a content file
  $cont = ($browser->get
  ("https://bb.zz.edu/bbcswebdav/courses/ST/t.txt",
   @headers))->content;
```

David C. Lyon, lyondc@gmail.com

```
        # Check for expected file contents and for
        # Specific error messages
        if ($cont !~ /expected file contents/ ||
            $cont =~ /A database error occurred/) {
          return 0;
        }
      }
      else {return 0;}
      return 1;
    }
```

☐ Monitor Apache and Activity Logs

A log monitoring tool such as `logwatch` could be used to report on apache and perhaps other logs. The things we might look for include:

Number of hits (higher than a threshold)
Number of hits to specific tools (User enrollment to a course)
Policy violations (tools that change Bb Learn configuration)
Error log records
408 timeouts (could indicate attack)
Suspect hits (not a valid Bb Learn URL)

Monitoring the `activity_accumulator` requires that it be configured to log activity. Configure it as follows:

Log into a Bb Learn Administrator account
Click on "System Admin"
Click on "System Reporting" under "Tools and Utilities"
Click "Auto-reporting Options"
Under "Reporting to Blackboard", select "No"
Under "Event Tracking", select "Yes"

"Reporting to Blackboard" should be "No" unless you do not mind statistics being sent to Blackboard Incorporated.

A tool like `logwatch` or even a home grown script should be able to summarize such hits as to not overwhelm those monitoring for anomalies. Ristic (2011) suggests looking for SQL injection, cross-site scripting, and command execution/file disclosure. Samples are listed below:

`delete from`	SQL injection
`insert into`	SQL injection
`<iframe src="…">`	Cross site scripting
`<img src="javascript:…">`	Cross site scripting
`mail`	Unix command
`/home`	Unix common path

Failed logins can be monitoring by checking the `activity_accumulator` table with the following query:

```
select session_id,data from activity_accumulator
where event_type='LOGIN_ATTEMPT' and status <> 1
```

☐ Tracking User Activity

The `activity_accumulator` field `internal_handle` column is useful in tracking down user activity. Sample values include:

```
tasks
cp_course_utilities_cartridge_add
cp_pool_manager
cp_gradebook2_manage_download_grade
tool_manager
blogs
```

For example, to find out who was using blogs during the last 30 days, run the following:

```
select user_id,timestamp from
users,activity_accumulator where
internal_handle='blogs' and timestamp > sysdate-30 and
user_pk1 = users.pk1
```

The `activity_accumulator` may not include all activity. For that, the `logs/tomcat/bb-access-log.YYYY-MM-DD.txt` is useful. Tracking User

Activity in Webserver Logs (2011) suggests a method to track user activity. In the following partial row in the file `logs/bb-access-log.YYYY-MM-DD.txt`, we have:

```
POST/webapps/blackboard/execute/
  uploadAssignment?action=submit
```

We see the assignment upload. In the first part of the record, we have:

```
192.168.1.29 - _20123_1
```

The number `20123` is just the `pk1` value in the `USERS` table. Identifying the user is simply a query:

```
select user_id from users where pk1=20123
```

☐ Monitor Apache Processes

The number of apache processes should also be tracked and compared with an expected value:

```
ps –fu bbaccount | grep "/apps/httpd/bin/httpd"
  | grep -v "grep" | wc –l
```

☐ Monitor Building Blocks logs

Building blocks can create logs in the following location:

```
content/vi/bb_bb60/plugins/
```

These should be monitored and cycled. They could contain communication details (Bb Mobile, for example). Configure the level of logging by editing the log4j file under the above path:

```
plugin/webapp/WEB-INF/classes/log4j.properties
```

Note: the file may be stored in a different location, depending on the building block.

☐ Tomcat/Java monitoring

David C. Lyon, lyondc@gmail.com

The log `logs/bb-services-log.txt` can contain Java, application, and database errors – much of it log spam but some of it useful and indicative of foul play or application issues. Get a feel for what is log spam, explainable, etc. Pursue others with Blackboard Support. Sample Java (application errors) errors that might raise concern:

```
Illegal access: user {unset id} trying to get
  blog comments for blog ..
Permission settings prohibit this action.
Error in handling Modify Domains Action -
  security exception
Access to this resource is prohibited
Only original user can complete this test
```

☐ Monitor Tomcat for long running threads

```
tools/perf_reports/get_tomcat_trace.sh
```

☐ Monitor the database for out of control queries

```
tools/perf_reports/run_sql_reports.sh
```

☐ Monitor tables for change

Monitor database objects for changes. In Oracle, this can be accomplished by the following query:

```
select object_name,last_ddl_time from all_objects
```

Tan (2011) suggests monitoring the `system_registry` and `users` tables. The first contains configuration information used by Bb Learn and the latter, Bb Learn users. In addition to those tables, consider tracking other tables listed below but not limited to (including those recommended by Tan):

`system_registry`	Registry changes
`institution_roles`	Intuitional roles changes
`system_roles`	System roles (like System Admin) changes

David C. Lyon, lyondc@gmail.com

`entitlement`	Privileges changes
`system_roles_entitlement`	Privileges – system role assignment changes
`users`	`dtmodified, dtcreated, system_role` – for changes
`users,domain_admin`	Users assigned a secondary system role
`users,institution_roles, user_roles`	Users assigned a secondary institution role
`plugins`	Building Blocks changes
`course_main`	Check for changes in `abs_limit`, `soft_limit`, and `upload_limit` legacy quotas.

The above tables can be queried daily and compared to saved results to alert for changes. This would alert someone of changes that were made but not authorized or even malicious changes. Related to the plugins table are the folders for Building Blocks located in the local application tree and the shared content area. The following queries will identify any plugin folders which can then be compared to a saved list.

```
find content/vi/bb_bb60/plugins -mindepth 1 \
  -maxdepth 1 -type d

find apps/tomcat/work/Catalina/localhost \
  -mindepth 1 -maxdepth 1 -type d
```

☐ Monitor and track Bb Learn configuration changes

AIDE (previously mentioned in Appendix F) can be used to track Bb Learn configuration changes. Short of a host based intrusion detection system, Perl can be used to detect changes using an Md5 hash. The Perl code below will compute the Md5 hash on files in the `config` folder.

```
use Digest::MD5;
use File::Find;

my @f=();
my($md5)=Digest::MD5->new;
my $fref = sub {
  push @f, $File::Find::name if (! -d);
};

# Get config (recurse into it)
find($fref,"/usr/local/blackboard/config/");
for (@f) {
  next if ! -f "$_";
  open(BIN, $_) || die;
  binmode(BIN);
  $md5->reset || die;
  $md5->addfile(*BIN) || die;
  my($sum)=$md5->hexdigest || die;
}
# Load a saved file and compare Md5 hash values
```

☐ Monitor for policy violations

Monitor for unusual activity that might violate the policy and procedure created for users assigned higher system roles. The following table lists the commands that might turn up information.

Check for System Admins enrolled in org/course as instructor	`select user_id,course_id FROM users,course_users,course_main where course_users.role='P' and user_id =` ***'local admin accounts'***
Check local accounts for inactivity (System Admin for example)	`select user_id from users where user_id =` ***'local admin accounts'*** `and available_ind = 'Y' and row_status <> 2 and pk1 not in (select distinct user_pk1 from activity_accumulator where user_pk1 is not null and to_char(timestamp,'YYYYMMDD') >= to_char(sysdate-60,'YYYYMMDD'))`
Find activity from `administrator` and `guest`. Seeing `guest` is not uncommon and needed in some cases. Seeing `administrator` should raise a red flag. This might occur when installing a	`select to_char(timestamp,'YYYY-MM-DD HH24:MI'),user_id from activity_accumulator,users where(user_id='administrator' or user_id='guest') and timestamp >= sysdate-1 and activity_accumulator.user_pk1 = users.pk1`

building block	
Check for course quota changes (legacy)	`Select course_id,abs_limit,soft_limit,upload_limit from bb_bb60.course_main abs_limit/1024)/1024 >` *`abs_limit`* `OR (soft_limit/1024)/1024 >` *`soft_limit`* `OR (upload_limit/1024)/1024 >` *`upload_limit`*

Check for policy governed activity	`select user_id,to_char(timestamp,'YYYY-MM-DD HH24:MI') from activity_accumulator,users where internal_handle='`***`handle`***`' and timestamp >= sysdate-1 and activity_accumulator.user_pk1 = users.pk1`
	
	The ***handle*** might be one of:
	`ssl_choice`
	`create_user`
	`batch_create_users`
	`batch_enroll_users`
	`manage_user`
	`user_properties`
	`password_modify`
	`observer_mod`
	`list_caliper_unit_memberships_by_user`
	`batch_remove_users`
	`admin_add_course_properties`
	`admin_course_properties`
	`copy_course`
	`restore_course`
	`batch_create_courses`
	`admin_add_club_properties`
	`admin_club_properties`
	`copy_club`
	`restore_club`
	`admin_plugin_manage`
	`admin_plugin_install`
	`batch_create_clubs`

If not using virtual installations, the `root_admin` account should be disabled (‘N’). `integration` can be disabled if not used	`Select available_ind from users where root_admin = ‘Y’or integration = ‘Y’`

Identify personal archives that have been created and remove them	`cd content/vi/bb_bb60/courses/`
	`find . -name ArchiveFile*.zip`

Tan (2011) suggests monitoring for Sys Admin (Administrators) logging in from an unexpected terminal. Since each user has an associated pk1 key, a list of those pk1 values can be kept. Those pk1 values show up in `logs/tomcat/bb-access-log.YYYY-MM-DD.txt` along with the IP address. It will be easy to search that log for the pk1 value "`_pk1_`" and identify where the user logged in from. Our policy should state that they may not log in with a *Sys Admin* (or other system role) from insecure locations like labs, WiFi, etc. The `session_id` and `user_id_pk1` (same as _pk1_) also show up in the `sessions` table.

☐ Monitor tables for activity

Monitoring certain tables in the database can reveal possible DoS attacks. If the active session spike, that should be a concern. Queries of the `sessions` table can also reveal an issue with the session invalidation task. One should determine threshold values that define what is normally expected after monitoring during average times. Note that activity is logged to the `activity_accumulator_queue` first, and is then moved to the `activity_accumulator` table periodically.

Notifications (too high)	`select count(*) from eud_item_recipient`
Tasks (low enough)	`select count(*) from queued_tasks where status='W'`
Activity Accumulator Queue (low enough)	`select count(*) from activity_accumulator_queue`
Sessions not updated (low enough)	`select count(*) from sessions s, users u where s.user_id_pk1=u.pk1 and s.timestamp <= u.last_login_date`
Total sessions (is cleaned up)	`select count(*) from sessions`
SESSION_INDS table (is cleaned	`select count(*) from session_inds`

up)	
Null sessions (low enough)	`select count(*) from sessions where user_id is null`
Updated sessions (high enough)	`select count(*) from sessions s, users u where s.user_id_pk1=u.pk1 and s.timestamp > u.last_login_date + .01`
Old sessions (low enough)	`select count(*) from bb_bb60.sessions where timestamp < sysdate - 4/24`
active (low enough)	`select count(*) from ( select user_id from sessions where user_id is not null and timestamp > SYSDATE-1/8)`

☐ Monitor for large log file size

A script should monitor system and Bb related log files for size. Establish a baseline of what is expected based on normal usage and alert when files grow beyond that size. In Perl, the size can be retrieved using the following:

```
    (stat(“path/file"))[7]
```

Bb Learn files under `logs/` to monitor include (but not limited to):

`bb-services-log.txt`	Java
`snapshot/bb-services-log-snapshot-jdbc.txt`	Snapshot
`update-tools/update-tool-log.txt`	Updates
`update-tools/pushupdate-tool-log.txt`	PushConfigUpdates
`collab-server/collab-server-log.txt`	Collab (chat)
`tomcat/catalina-log.txt`	Tomcat
`httpd/access_log_YMD`	Apache hits
`httpd/error_log_YMD`	Apache errors
`httpd/mod_jk_YMD.log`	Mod_jk connector
`tomcat/bb-access-log.YYYY-MM-DD.txt`	Tomcat/Apache

`tomcat/stdout-stderr-YMD.log`	Tomcat
`content-exchange/content-exchange-log.txt`	Course archive, etc.

☐ Monitor for errors

Monitoring the logs for errors can reveal a DoS or buffer overflow attack. See the table blow for possible strings to look for (regular expression).

`bb-services-log.txt`	`OutOfMemory`
	`authorization check failed`
	`permgen`
	`ORA-`
	`TNS-`
	`A database error occurred`
	`too many open files`
	`Java heap space`
	`java.sql.SQLException`
	`Unable to retrieve a database connection`
	`Exception thrown by getter for property grade of bean row`
	`InternalError`
	`ConnectException`
	`I/O`
	`specific error code`

`tomcat/stdout-stderr-YMD.log`	`^ERROR`
	`^SEVERE`
	`stackoverflow`
	`OutOfMemory`
	`Full GC`
	`permgen`
	`too many open files`
	`Java heap space`
	`OALL8 is in an inconsistent state`
	`Dumping heap to java_pid`
	`Error reading 'grade' on type`
	`JVM exited`
	`abortable-preclean:`
	`promotion failed`
	`hung`
	`Launching a JVM`
	`concurrent mode failure`
	`InternalError`
	`ConnectException`

	were stopped: \d+\.[3-9] were stopped: [1-9]\. segfault CMS-concurrent-sweep:
httpd/error_log_YMD	Maxclients broken pipe fault
httpd/mod_jk_YMD.log	Tomcat is probably not started connect to 127\.0\.0\.1:8109 failed Failed opening socket [error] [fatal]
bb-session-log.txt	Session fingerprint changed

It would also be good to check for a large record count for `error_log` and `mod_jk`. Large counts could be due to DoS and/or Tomcat issues such as when Apache cannot connect to Tomcat (AJP) or Tomcat is hung.

☐ Monitor Server Restarts

Server restarts may not be apparent as they may only involve a few seconds of downtime. The following stings indicate a server restart. This should be monitored regularly.

Tomcat	tomcat/stdout-stderr-YMD.log	Server last reloaded at:
Apache	httpd/error_log_YMD	Resuming normal operations

☐ Performance monitoring

Monitor Bb usage by regularly gathering performance metrics for Bb related processes, such as Java. A command such as:

```
ps -H h -o rss -p PID
```

can identify memory usage.

The Java heap can also be monitored by using the command:

```
$JAVA_HOME/bin/jmap -J-d64 -heap JAVA_PID
```

☐ Monitor the growth of tables

Users post messages that end up in tables. Unusual table growth can be an indication of abuse.

☐ Monitor course and organization sizes and quotas

Log into a Bb Learn Administrator account
Click on "System Admin"
Under "Tools and Utilities", click on "System Reporting"
Click on "Disk Usage"
Select "Course Size" from the "Search" menu
Enter a size (200MB for example)

This will identify courses and organizations using a lot of disk space. This could be due to increased quotas, media files, etc. Of course default quotas should be defined – also done from the *System Admin* panel. The Messages tool may be subject to abuse. The size of a course on disk could increase quickly due to the posting of attachments. The tool can be used to send to all course/organization recipients and that results in a copy for each user folder:

```
content/vi/bb_bb60/courses/1/course-ID/messages/users
```

○ Partnership with Information Security Officer (ISO)

Working with the Information Security Officer or equivalent is vital. For example, `php` hits to Bb Learn from inside the organization should be reported and may indicate that the client computer has been compromised. The ISO can quickly take action to take a rogue computer off the network.

Appendix G – Periodic Maintenance

The steps provided here are centered on security. There are numerous other steps involved in periodic maintenance, not listed here.

- ○ End of Semester or Term
  - ☐ Create a change request for the Change Control Board
  - ☐ Communicate with stakeholders
  - ☐ Review latest security bulletins– keep a tracking database for unresolved issues
  - ☐ Report Bb Learn vulnerabilities to LearnSecurity@blackboard.com
  - ☐ Review NIST publications at http://csrc.nist.gov/publications/PubsSPs.html
  - ☐ Review performance metrics
  - ☐ Run full content/database backups and course archives (See "Daily and Weekly Maintenance" below)

    These backups should be kept offsite and be retained for the time specified in any data retention policy.
  - ☐ Verify integrity of backups and archives
  - ☐ Apply O/S Patches

  Use "`rpm -freshen`" or similar tool to freshen packages.

  - ☐ Apply any Bb Learn Service Pack Updates and/or patches

  Note that back-end and user customizations will be reset to defaults. These can include `JAVA_HOME`, SSL and authentication settings, course settings, layouts, modules, tabs, etc. It is best to keep a baseline of those and always check those after a service pack ("Blackboard 9.1 Service Pack 5 Release Notes," 2011). Also be aware that privileges assigned to system roles may revert.

  Service packs and some patches will run or require a run of:

```
tools/admin/PushConfigUpdates.sh.
```

This tool updates many files based on configuration values. An example is the `apps/httpd/conf/httpd.conf.bb`. Files with `.bb` extension are used as input to create the production file (no `.bb`). It is important to note that this tool may revert some settings as well, including re-enabling the Content System "immediate update" feature. Verify the following file is not overwritten:

```
apps/tomcat/lib/Search.properties
```

☐ Verify tuning parameters – adjust as required

☐ Sanitize configuration

Remove any passwords needed (see release notes) for service packs or patches from configuration files. These are needed when applying updates, etc.

☐ Bbpatch tool

The BbPatch Release 2.0 documentation (2011) contains detailed instructions on using the `bbpatch.sh` utility. Applying a patch is done using the following:

```
bbpatch.sh apply AS-129347.bbp
```

There are also options roll back, describe, and list installed patches.

☐ JAVA updates

Keep Java JDK current as this can lead to better performance and fixes for security issues. Caution should be applied as some updates can introduce issues with garbage collection, etc. It might be preferable to download the kit directly from http://java.sun.com rather than waiting for Red Hat to provide an update.

☐ Update the Java Runtime Environment

The JRE is provided as a download to collaboration users (chat). Leaving an insecure version to download presents added risk to users. Download the latest package from http://java.sun.com and replace the following file on the collaboration server (must be same name):

`apps/collab-server/http/webapps/client-lib/jre-1_X.exe`

☐ Remove courses/orgs that are not part of the retention cycle

Also check for misnamed courses and organizations and stale organizations and remove them.

○ Security Audit (Periodic)

☐ Remove expired users (not in Identity Management)

☐ Verify permissions and ownership of Bb Learn, O/S, and home grown configuration files. Harden where needed.

☐ Perform Virus Scan

Perform a virus scan of all content using `clamdscan` or other tool. See “Daily and Weekly Maintenance” below.

☐ Audit system and institution role assignments (especially System Admin)

☐ Audit local accounts (not tied to Identity Management)

Remove those not needed and change passwords on others

☐ Remove or disable vendor accounts not being used

☐ Verify that Bb Learn *Persistent Cookies* are turned off (section 6.3.8)

☐ Audit Bb Learn and O/S permissions and ownership assigned in hardening checklists

☐ Verify Bb Learn settings have not reverted – including Building Blocks

☐ Periodically, change the `root`, `bbacount`, Bb Learn `administrator`, `integration`, `root_admin`, and Oracle passwords. Don’t forget any special accounts used for reporting, etc.

☐ Scoring and Scanning

The Center for Internet Security has scoring tools to help assess the state of security for a variety of platforms, including Apache. Please see http://benchmarks.cisecurity.org.

David C. Lyon, lyondc@gmail.com

The Nessus scanner is a commercial vulnerability scanner with access to updated plugins to scan for many kinds of vulnerabilities. It has many configuration options. See http://www.tenable.com/products/nessus.

Nikto2 is designed to scan web servers and is available at http://cirt.net/nikto2.

Ristic (2005) suggests the following actions:

Utilize `nmap (nmap.org)` to do port scanning. A stealth scan for open ports is done using:

```
nmap -sS ip-address
```

Verify the cipher used (some are not as secure anymore). SSLDigger can be used for this and is available at http://www.mcafee.com/us/downloads/free-tools/ssldigger.aspx. Also test `openssl` for support of protocol version 2 (not desired) and make sure the certificate is trusted and verify the key length. Do the following:

```
apps/openssl/bin/openssl s_client \
  -host localhost -port 443 -no_ssl3 -no_tls1
```

Also, verify that SSL redirect is working by visiting a page directly – for example: http://bb.myinstitutiton.com/webapps/login should redirect to https.

Next, identify the web server information by issuing the following command:

```
telnet localhost 80
OPTIONS / HTTP/1.0
Host: bb.myinsitution.com
```

This should mask the Apache version if we've hardened Apache properly (`ServerTokens`). There are tools to get past this, though.

Test whether the server supports proxy operations (allowing unrestricted use) via the following:

```
telnet localhost 80
HEAD http://somesite.com:80 / HTTP/1.0
```

We should get a 403 or 401 (unauthorized). Similarly, if we replace `HEAD` with `CONNECT`, you should get a 302 returned to confirm the absence of a proxy.

○ Daily and Weekly Maintenance

☐ Review latest security bulletins

The following are great sources:

http://www.sans.org/newsletters/risk/
http://secunia.com/advisories/
http://www.cert.org/advisories/

Keep a tracking database for unresolved issues. CVE Details has statistics for Blackboard at http://www.cvedetails.com/vendor/504/Blackboard.html

☐ Report Bb Learn vulnerabilities to LearnSecurity@blackboard.com

☐ Daily content/database and weekly course/org archives

Course and organization (if using the community system) archives are done using the following procedure:

```
select course_id from course_main WHERE cond

# The above should be output to a file
CRS-ID,Path-of-Archives

apps/content-exchange/bin/batch_ImportExport.sh
  -f file -l 1 -t archive
```

Note: these backups should cycle off site.

☐ Housekeeping

Keep storage clean of temporary and old log files. Blackboard Support provides the `cleanbb` utility that will generally clean up Bb Learn temporary files. Bb Learn

David C. Lyon, lyondc@gmail.com

logs can accumulate rapidly and should be purged with a command like the following:

```
find /usr/local/blackboard/logs/ -type f \
  -mtime +30 –follow -exec rm -f {} \;
```

Bb Learn comes with a task in `config/bb-tasks.xml.bb` that will cycle logs. You may also find it necessary to have a script to cycle some of the logs under `logs`. Building blocks logs may also need to be cycled. Implement log rotation at the O/S level as well.

☐ Virus Scanning

Provide daily anti-virus scanning using a free tool such as `clamdscan`. The following provides a sample command. The `clamd` daemons must be running.

```
clamdscan -f file -i --fdpass -m –l logfile
```

File would be a list of files that have changed since the previous run.

David C. Lyon, lyondc@gmail.com