Global Information Assurance Certification Paper

Implementing Redmine for Secure Project Management

GIAC GCPM Gold Certification

Author: Russ McRee, russ@holisticinfosec.org

Advisor: Rick Wanner

Accepted: January 27th 2013

Abstract

Security and collaborative project management should not be exclusive. Software designed to support secure project management and security-oriented projects can be both feature rich and hardened against attacks. Web applications such as Redmine offer just such a solution and can embrace the needs of project managers and security practitioners alike. Redmine is project management and bug tracking software built on Ruby on Rails with a focus on collaboration, functionality, and when enhanced with specific plugins, can be configured securely to facilitate security-oriented projects. As a productivity platform, Redmine allows convenient team workflow while embracing the needs of virtual or mobile project members with a focus on socially oriented processes. This paper will explore the secure implementation and configuration of a Redmine server, and then transition into step-by-step details for managing a real world web application penetration testing project using Redmine. This will include the distribution of a virtual machine ready-built for real world use during such projects, pre-configured with a project template based on workflow in the SANS 542 Web Application Penetration Testing course.

1. Introduction

One of the core tenets of a good project management practice is the safekeeping of project information in a readily available, secure resource. As the days of paper-based records, with all projects participants located in one office, have long passed us by, digital project portals have become essential tools for success. The quandary though, as with all computer-based offerings, is the task of ensuring the confidentiality, availability, and integrity of the project data being accessed and updated collaboratively. More importantly, imagine if the project data is security oriented content such as vulnerability findings from a penetration test. Were such data to fall in the wrong hands the outcome could be devastating for the testers and the client. As such, the premise for this paper is the focus on a secure, hardened implementation of Redmine and its use in managing security oriented projects such as web application penetration tests. Redmine offers an extensive capacity to facilitate project management requirements while keeping security practitioners and auditors satisfied when configured as discussed in the following pages. "Secure project management for security oriented projects' is the foundational position for this research.

From the TurnKey Redmine Web page: "Redmine is a Rails web application that provides integrated project management features, issue tracking, and support for multiple version control programs. It includes calendar and Gantt charts to aid visual representation of projects and their deadlines. It also features multi-project support, role based access control, a per-project wiki, and project forums." (TurnKey, 2012). Additionally, a tool such as Redmine allows the convergence of software *and* security testing. As a software configuration management (SCM) tool, Redmine is ideally suited to projects related to software development. That said, the security expertise required to security test software needs equal consideration and project management. "Sometimes security experts - also known as penetration testers, or pen-testers for short - work on the same test team as functionality testers; other times, pen-testers work as security consultants and are hired by the software development company to perform security tests (also known as pen-tests)" (Gallagher, Jeffries & Landauer, 2006). Regardless of who

solicits the use of pen-testers, the related pen-test is a project, and Redmine is the ideal application to provide the agile, flexible platform pen-testers need to coordinate their efforts with the help of a PM or team lead.

2. Installation

Redmine installation and configuration using a TurnKey Linux Redmine appliance built on a Debian-based Linux distribution, is reasonably straightforward. Your ability to install a Linux operating system from an ISO file on a dedicated machine, or configuring a VMware virtual machine is assumed. Readers can also opt to deploy Redmine via Amazon EC2 if they wish. It is also assumed you have control of or access to an Active Directory domain for LDAP authentication to Redmine, as it allows for more robust user management. As referenced later in this paper, the IP address of the Redmine instance was 192.168.248.16 and 192.168.248.248 for the domain controller. The stable version of the TurnKey virtual Redmine appliance (version 12) running on a lean instance of Debian Squeeze (CLI only, no X11 GUI) via VMWare Workstation 9 was utilized for this research. Note, readers will find running the shell via Putty or a client where you can cut and paste installation strings easier as the VMWare tools aren't effective without the GUI. When contemplating Redmine installations there are payoffs to consider: performance under heavy work load versus Redmine memory use, a more complicated installation process versus convenience and expeditious implementation, and more simply, convenience versus security. The TurnKey Redmine virtual machine strikes an ample balance of all considerations. This TurnKey Redmine appliance relies on Passenger, a module for Apache that hosts Ruby on Rails applications, and supports the use of SSL/TLS (configured by default) and ModSecurity for better security.

As of this writing (09 JAN 2013) the current version of Redmine is 2.2.1 and will be described herein. The installed version of Redmine on the TurnKey appliance is 1.4.4; this guidance will include its upgrade to Redmine 2.2.1.

First, download the Turnkey Linux VM appliance[1] and open it in VMWare. The first boot routine will ask you to create passwords for the root account, the MySQL root user, and the Redmine admin. When the routine completes you should be presented the TurnKey Linux Configuration Console as seen in Figure 1.

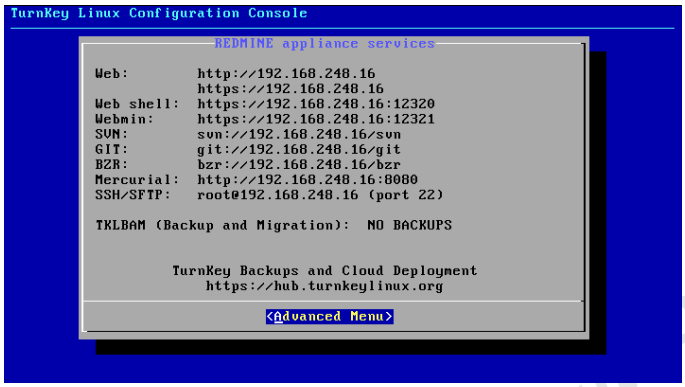

Figure 1: TurnKey Linux Configuration Console

In the Hardening section, the process of disabling the services you don't intend to use will be discussed. Take a snapshot of the virtual machine at this point and name the snapshot Base Install.

2.1. Update Redmine version

Via Putty on a Windows host or a terminal on a Linux/Mac host, log on as the root user to the Redmine server over SSH. Utilizing convenient guidance (Killer, 2007) for a seamless upgrade, as well as additional ease-of-use or security-centric steps, execute the following commands from your shell:

1. *apt-get update*
2. *apt-get upgrade*
3. *apt-get install locate*
4. *updatedb*
5. *cd /var/www*

[1] http://www.turnkeylinux.org/download?file=turnkey-redmine-12.0-squeeze-x86-vmdk.zip

6. *mv redmine redmine -old*
7. *hg clone --updaterev 2.0-stable https://bitbucket.org/redmine/redmine-all redmine*
8. *cp redmine -old/config/database.yml redmine/config/database.yml*
9. *cp -r redmine-old/files/ redmine/files/*
10. *chown -R root:www-data /var/www/ redmine*
11. *cd redmine*
12. *gem install bundler*
13. *gem install test-unit*
14. *bundle install --without development test rmagick*
15. *mkdir public/plugin_assets*
16. *rake generate_secret_token*
17. *rake db:migrate RAILS_ENV=production*
18. *chown -R www-data:www-data files log tmp public/plugin_assets*
19. *rake redmine:plugins:migrate RAILS_ENV=production*
20. *chmod -R 755 files log/ tmp/ public/plugin_assets*
21. *rake tmp:cache:clear*
22. *rake tmp:sessions:clear*

Running the script */var/www/redmine/script/about* should result in a response similar to Figure 2.

```
Environment:
  Redmine version                2.2.1.stable
  Ruby version                   1.8.7 (i686-linux)
  Rails version                  3.2.11
  Environment                    production
  Database adapter               MySQL
Redmine plugins:
```

Figure 2: Redmine upgrade confirmed

The Redmine version has now been updated from1.4.4 to 2.2.1.

A more streamlined, tech-centric theme from Pixel Cookers was utilized for this installation, and is easily installed as follows:

1. *cd /var/www/redmine*
2. *git clone git://github.com/pixel-cookers/RedmineThemePixelCookers.git public/themes/pixel-cookers*
3. *service apache2 restart*

4. Browse to your Redmine instance, login as the admin user with credentials established during the initial instance, then navigate to *Administration* | *Settings* | *Display* and select your newly created theme in the *Theme* drop-down list.
5. Save your settings.

The use of RMagick with your Redmine installation will enhance the user's graphical experience as it allows the Ruby programming language to better leverage the ImageMagick® and GraphicsMagick image processing libraries. Enable it as follows:

apt-get install libmagick9-dev

gem install rmagick

2.2. LDAP

LDAP authentication is inherent to Redmine but requires a bit of setup. The example Active Directory domain name utilized via a virtual Windows Server 2008 domain controller was REDMINE. The user *redminer* was established as the service-like account utilized by Redmine to access the directory. Do not user a domain administrator account for this user. Should your Redmine instance be compromised so too then would be your domain. Via your browser, as the Redmine admin user, navigate to *Administration* then *LDAP Authentication*. Refer to the Redmine LDAP Authentication page (Massip, 2012) via the Redmine WIKI but refer to the following example configuration as successfully utilized for this research.

```
Name     = REDMINE
Host     = 192.168.248.248
Port     = 389
LDAPS    = no
Account  = REDMINE\redminer
Password = <password>
Base DN  = DC=REDMINE,DC=local

On-the-fly user creation = no
Attributes
  Login     = sAMAccountName
  Firstname = givenName
  Lastname  = sn
```

```
Email      = mail
```

Select *Save* then, assuming a correct configuration, you should receive indication of a successful connection when you click *Test* on the resulting *Authentication Modes* page as seem in Figure 3.

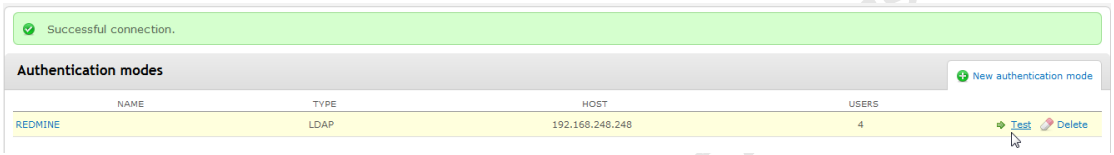

Figure 3: Successful Redmine LDAP connection

See Addicted to IT's configuration guidance ("Redmine ldap integration," 2010) as a troubleshooting reference as well; the "keep it simple suggestion" is a good one.

2.3. Email

Similarly, a collaborative platform such a Redmine needs to be able to send email notifications to project participants. To configure email setting you'll first need to copy */var/www/redmine/config/configuration.yml.example* file to *configuration.yml* and provide working email account settings. You can choose the sendmail functionality provided by the Redmine appliance's native Postfix mail transfer agent, or opt for a webmail account as provided in the working example below. There are production and development sections at the very end of *configuration.yml*; establish your settings there.

```
production:
  email_delivery:
    delivery_method: :smtp
    smtp_settings:
      enable_starttls_auto: true
      address: "smtp.mail.yahoo.com"
      port: 587
      domain: "smtp.mail.yahoo.com"
```

authentication: :plain

user_name: "redmine@yahoo.com"

password: "<password>"

As seen with the LDAP settings, a successful configuration here is validated when, after navigating to *Administration* | *Settings* | *Email Notifications*, you click *Send a test email* and receive it at the account configured as seen in Figure 4.

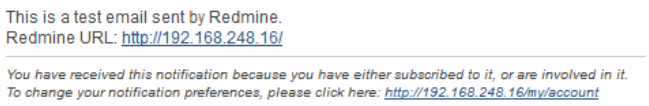

Figure 4: Redmine test email

2.4. Plugins

There are a few additional Redmine plugins that should be installed to enrich the project participant's experience. While there are 300+ plugins available for Redmine, three stood out as useful for security related projects such as penetration testing. These plugins, Redmine - Ldap Sync, the Redmine Attach Screenshot plugin, Redmine (Monitoring & Controlling), and the Scrum2B Plugin provide enhanced features. Redmine - Ldap Sync further integrates Redmine with LDAP (Active Directory as discussed here) to:

- detect and disable users that have been removed from LDAP.
- detect and disable users that have been marked as disabled on Active Directory
- detect and include nested groups

The Attach Screenshot plugin should speak for itself, but note that for penetration testing project management as to be described in **Engaging with Redmine**, screenshots are integral to collaboration. Monitoring and Controlling offers additional graphs including pie charts for issues by status, manageable and unmanageable issues, and overdue tasks by project. Finally, Scrum2B provides the Scrum Board for agile project management where teams treat the Scrum Board like a real board and drag and drop tasks to sort and organize the sprint.

Enable the plugins as follows:

1. *cd /var/www/redmine/plugins*
2. *git clone git://github.com/thorin/redmine_ldap_sync.git*
3. *git clone git://github.com/alexmonteiro/Redmine-Monitoring-Controlling.git*
4. *git clone git://github.com/scrum2b/scrum2b.git*
5. *hg clone https://bitbucket.org/StrangeWill/redmine-inline-attach-screenshot redmine_inline_attach_screenshot*

The last step before moving to the hardening phase is to remove the sample projects via *Administration* | *Projects* and is easily done with the *Delete* option.

3. Hardening

The hardening process is optional and can be conducted in part, or in full, but is strongly recommended for the most secure Redmine installation possible. Each of these steps should be validated to ensure that they don't impede functionality or cause service disruption.

3.1. Disable unnecessary services

By default, the TurnKey Redmine instance to accept connection via TCP ports 80, 443, and 22 for HTTP, HTTPS, and SSH as well as 12320 and 12321 for Web shell (shellinabox) and Webmin. Web shell and Webmin are attack surfaces you can immediately reduce, and while convenient, are often targeted for vulnerabilities such as cross-site scripting and account bruteforcing. Redmine also provides a number of SCM services including SVN, GIT, BZR, and Mercurial. If you do not intend to use Redmine for SCM you can disable them all or choose one preferred protocol and disable the others. For this research *sysv-rc-conf* was installed via *apt-get install sysv-rc-conf* then utilized to disable the *shellinabox*, *webmin*, *bzr*, *git-daemo*n, and *svnserve* daemons as seen in Figure 5.

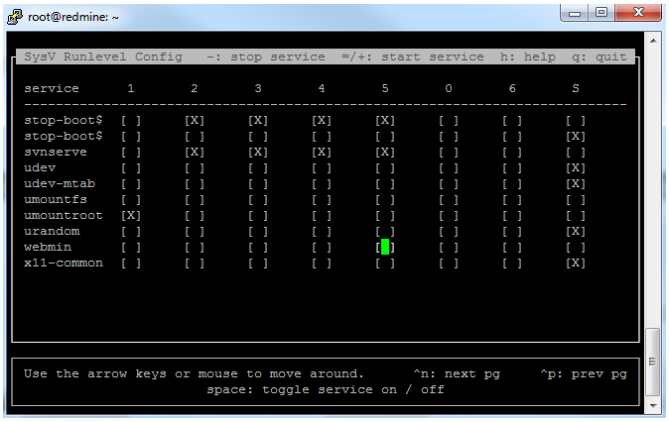

Figure 5: sysv-rc-conf to disable service

You can also then opt to delete the */etc/webmin* and */etc/shellinabox* directories as well as remove the shellinabox and gitdaemon accounts with *userdel shellinabox* and *userdel gitdaemon*. The Mercurial service was left intact to allow any version control that might be necessary for penetration testing scripts and files.

Before tightening down SSH, you'll need to install *sudo* and add a Linux system user account with sudo privileges so you can disallow root logons via SSH. Assuming a user name such as *redmine*, execute the following at the Redmine system root prompt:

apt-get install sudo
adduser redmine sudo

3.2. Tighten down SSH

To harden SSH daemon options edit */etc/ssh/sshd_config* by changing line 26 to *PermitRootLogin no*, line 43 to *PermitEmptyPasswords no* (should speak for itself) and line 62 to *X11Forwarding no* as there no X11 available on this system regardless. Restart the SSH service (*service ssh restart*), a step that will disconnect you if connecting via Putty, then attempt to login as root; you should be denied. Hereafter you'll sudo as *redmine* or the account name you chose. You can choose to restrict access to the SSH daemon to select IP addresses using ufw (uncomplicated firewall), a CLI firewall configuration tool.

3.3. Restrict Redmine web access to HTTPS only

To force all web access to Redmine to HTTPS, edit */etc/apache2/conf/redmine.conf* as follows. Line 4 of *redmine.conf* is *RewriteEngine On*. On line 5 copy:

RewriteCond %{HTTPS} off
RewriteRule (.) https://%{HTTP_HOST}%{REQUEST_URI}*

Take note of the fact that the certificate is locally generated and untrusted. As such you will experience browser certificate warning unless you choose to acquire and import a certificate created by a trusted certificate authority.

3.4. Implement UFW

UFW, the uncomplicated firewall allows you to restrict access to only desired services and deny all other traffic. First, install *ufw* with *sudo apt-get ufw* now that you've disabled root login over SSH and are escalating privilege only as needed. To configure the Redmine system firewall for only necessary traffic execute:

sudo ufw allow ssh
sudo ufw allow https
sudo ufw allow 8080/tcp
sudo ufw logging on
sudo ufw enable
sudo ufw status verbose

Remember, if you wish to restrict SSH access to specific IP addresses, an appropriate command might resemble *sudo ufw allow from <ip address> to <protocol> port <port number>* (Mikus, 2012).

While this is indeed a Linux based implementation, it is recommended that you reboot the system after completing installation and configuration, and running a port scan against it to validate that all the disabled services are unresponsive as expected. More simply, audit against all assumptions.

4. Engaging with Redmine

Now fully configured and hardened, Redmine is ready to support its first project. Following will be a step by step description of a penetration testing engagement where Redmine is utilized to provide project support for a team of three.

The first and most important steps to undertake are the elimination of all unwanted permissions for the *Non member* and *Anonymous* roles. Login to Redmine as the admin user and select *Administration* | *Roles and permissions* | *Non member* | *Uncheck all* | *Save*. Repeat this process for the *Anonyomous* roled. These steps will ensure that you don't inadvertently expose project data to those who don't have explicit permission to view it. Next, to add users for this project, select *Administration* | *Groups* to add a group called PenTesters. From *Administration* | *Users* add three users with appropriately defined login names pentester1 (Fred), pentester2 (Wilma), pentester3 (Barney), and pentestpm (BamBam) and add them to the PenTesters group. Remember these users need to also have been created in the domain you're utilizing for LDAP authentication. In keeping with modern global business norms, the project participants are in different locations, one each in the Pacific, Central, and Eastern time zones. Via the *Administration* menu, under *Projects*, create a project called *Web Application Pentest*. The activities related to this project will be drawn directly from tasks outlined in *SANS 542: Web App Penetration Testing and Ethical Hacking* course as well as the Samurai Web Testing Framework (Siles, 2012). Select all *Modules* and *Trackers* for the project. You'll note that *Monitoring and Controlling by Project* and *Scrum2b* are available as implemented during the installation phase described earlier. These plugins will be described in more detail as their use is inherent to agile project management for projects such as penetration testing.

Redmine allows the creation of subprojects as well; the *Web Application Pentest* project should be divided into four subprojects named as follows: *1-Recon*, *2-Mapping*, *3-Discovery*, and *4-Exploitation*. Add each of them from Redmine Web Application Pentest project page and remember to enable all *Modules* and *Trackers*. For penetration testing projects you may not opt to use the *Feature Tracker* for each project, but if one of

your testers develops exploits or scripts used by the rest of the team it could well be considered a Feature.

Add the user accounts for the three penetration testers and the project PM user as project and subproject members via the Members tab as seen in Figure 6.

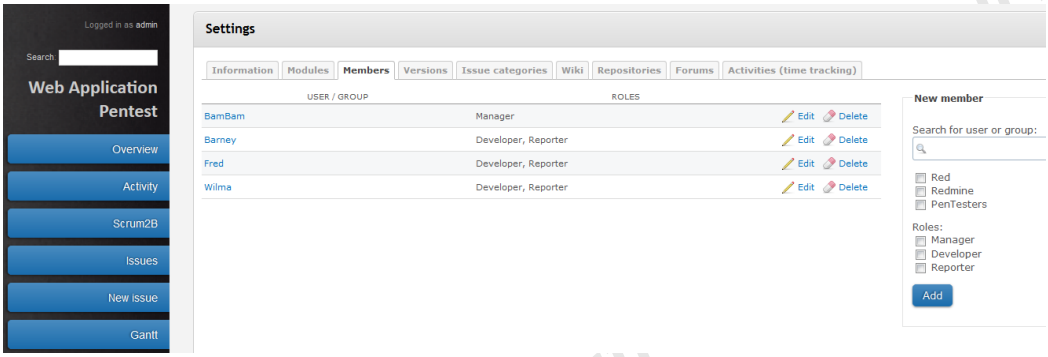

Figure 6: Pen-test Project Members

Return to the project overview, select *1-Recon* under subprojects, and add a new issue. File a bug for each recon phase task you'd like completed, with the applicable start and due dates. You can upload related files, screenshots (thanks to the plugin installed earlier), and designate an assignee, as well as watchers.

Under *Settings* for each project or subproject you define you can establish issue categories. This is an ideal method by which to establish penetration testing activities for each subproject. As an example, the recon phase of a web application penetration test includes general recon along with DNS and Whois lookups, search engine analysis, social network analysis, and location analysis. Establishing each of these as issues categories will then allow bugs (tasks) to be filed specific to each category. Each bug can in turn be assigned a pen-tester with start and end dates, along with files that might be useful to complete the task. Location analysis could include gleaning location data from victim Tweets as described in *Violent Python* (O'Connor, 2013). Twitter provides an API to developers which allows information gathering about individuals (potential penetration test targets). A script from Violent Python to help in this information gathering can be uploaded into the Redmine bug, Location data from Tweets as seen in Figure 7.

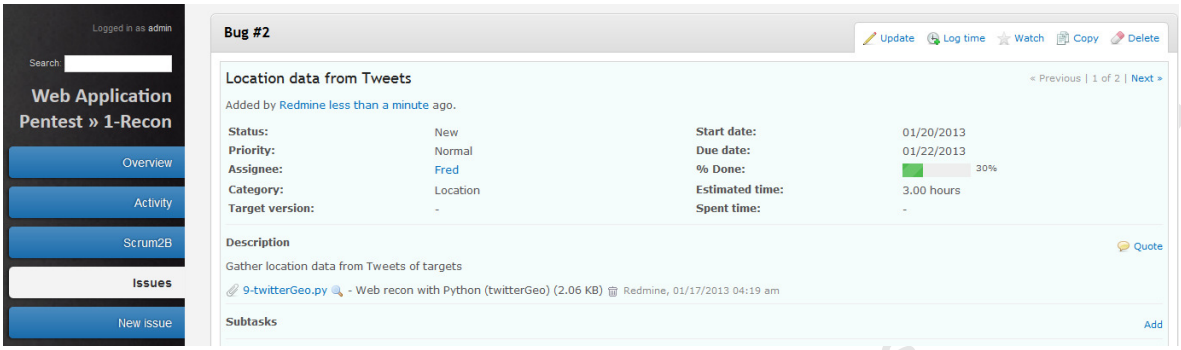

Figure 7: Bug (task) assigned to Fred, with helper code

As bugs are added, assigned, and/or updated, if configured to communicate verbosely, Redmine will email notices to the appropriate parties. The email as seen in Figure 8 was received as a function of filing the bug in Figure 7.

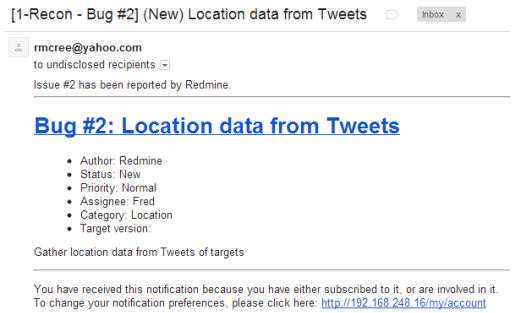

Figure 8: Email notice for bug (task) filed

This allows real-time communication among penetration testers or any project participants defined in your Redmine deployment. As pen-testers generate findings, they can be uploaded to the associated bug, and if versioning is required, managed via the Mercurial SCM offering as described during installation.

Bug status can be tracked as *New*, *In Progress*, *Resolved*, *Feedback*, and *Closed* or *Rejected*, and each bug can be assigned a priority and estimated time. As completed, actual time spent on each bug can be tracked too (Figure 9).

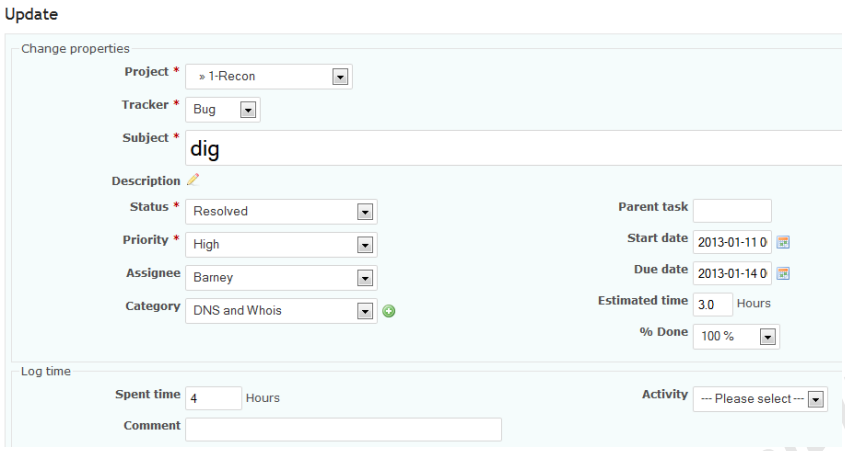

Figure 9: Bug (task) property updates

Overall project time allotments as defined in the bug then track quite nicely via the Redmine Gantt functionality as seen in Figure 10.

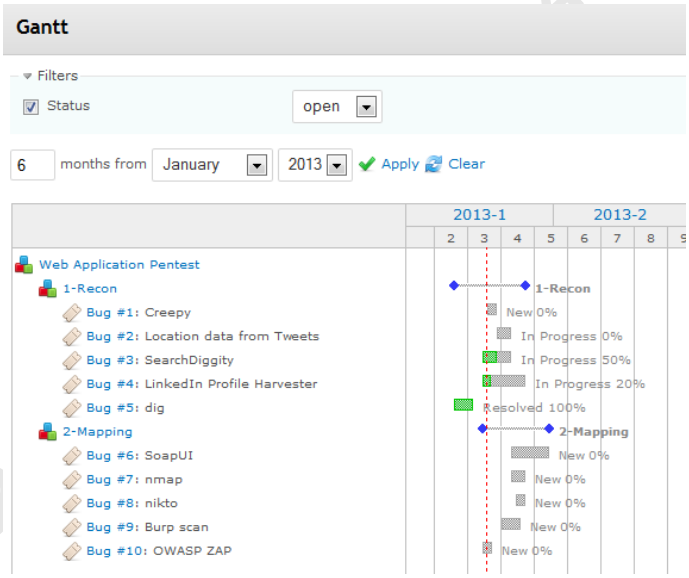

Figure 10: Redmine Gantt functionality

Equally useful, given its click through functionality, the Calendar view allows users to choose a day and an associated project and review the bugs (tasks) associated with that day.

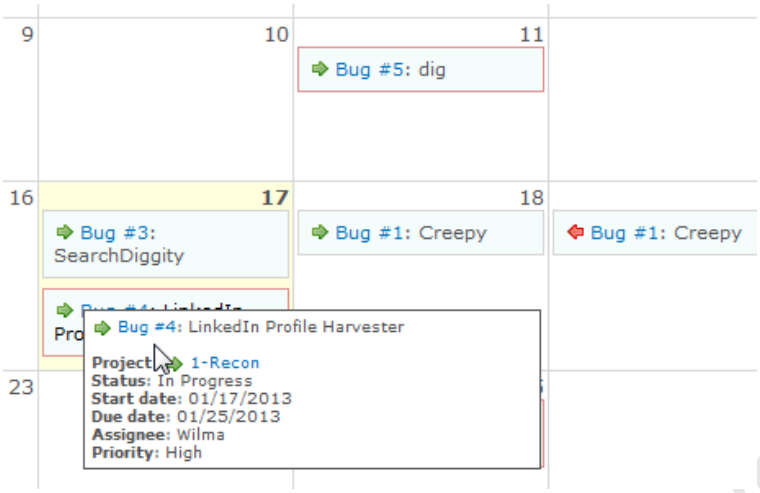

Figure 11: Redmine calendar

A project wiki and/or forum (pull communication) can also be created to ensure the appropriate level of messaging and communication that serves to facilitate the project in addition to the email notifications (push communication).

4.1. Scrum2b

The concept of agile software development has, over time, been applied directly to project management. Consider the use of Scrum methodology as part of agile project management. According to Agile Project Management with Scrum, "the heart of Scrum lies in the iteration. The team takes a look at the requirements, considers the available technology, and evaluates its own skills and capabilities. It then collectively determines how to build the functionality, modifying its approach daily as it encounters new complexities, difficulties, and surprises. The team figures out what needs to be done and selects the best way to do it. This creative process is the heart of the Scrum's productivity" (Schwaber, 2004). These creative processes, assessment of capabilities, and changing complexities and surprises are also inherent to any penetration test and as such, the agile project management framework is an ideal way to coordinate pen-test projects. The Scrum2b plugin for Redmine is well suited to answer this calling. If each phase of the pen-test is considered a sprint as defined by the Scrum process, the planning and awareness necessary to support the sprint is essential. The Scrum2b interface is a virtual Scrum Board that allows project participants to track activities by bug and members

while editing the bug on the fly with the appropriate permission. The *pentestpm* user, as project manager, could adjust task's percentage of completion right from Scrum2b using the time slider.

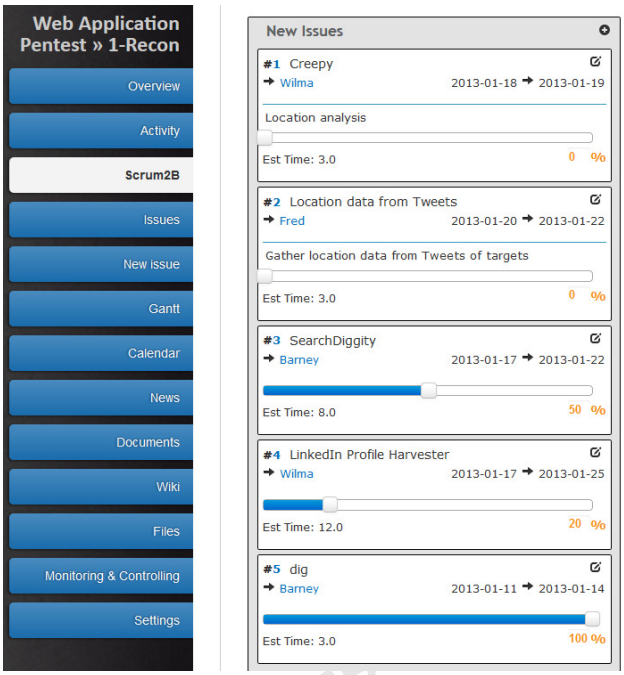

Figure 12: Scrum2b Scrum Board for pen-testers

If the assignee needs to jump right to the bug, the plugin is fully hyperlink enabled. The Scrum Board allows filtering the view by members and issues. New issues can also be added right from the Scrum Board.

4.2. Monitoring & Controlling

All projects require the right balance of monitoring and controlling, and penetration tests are no exception. The *Monitoring and Controlling Project Work* process includes "gathering, recording, and documenting project information that provides project status, measurements of progress, and forecasting to update cost and schedule information that is reported to stakeholders, project team members, management, and others" (Heldman, 2009). The Monitoring & Controlling plugin for Redmine shines in this capacity. Established as a convenient left-pane menu item with the Pixel Cookers theme, this plugin creates a dashboard for project data organized by Tasks Management,

Time Management, and Human Resource Management. Tasks Management tracks Tasks by Status, Tasks by Category, and Task Management (manageability). Applied again to the context of a pen-test project, Figure 13 represents the Recon phase of a pen-test.

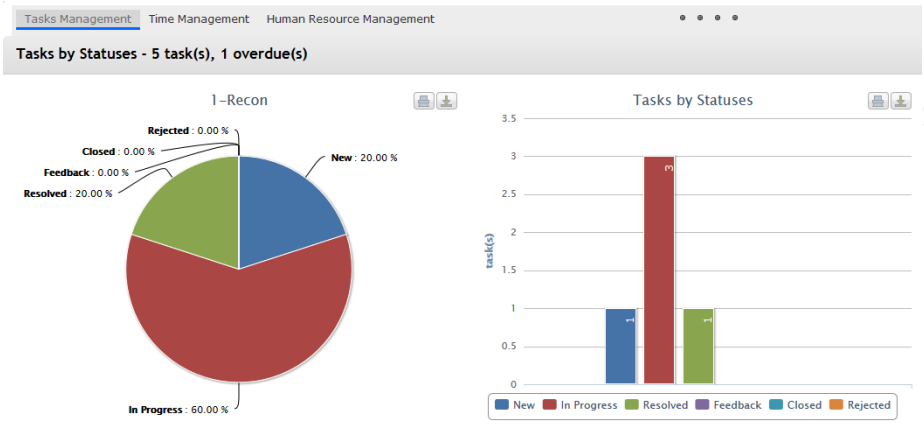

Figure 13: Monitoring & Controlling Tasks Management

Time Management is no less important than tracking the status of tasks, and the Redmine Monitoring & Controlling plugin measures by *Due Hours* and *Spent Hours*. This visual representation clearly denotes estimated time as compared to time spent in execution. This is critical in pen-testing when the rules of engagement may define very clear time windows and limits.

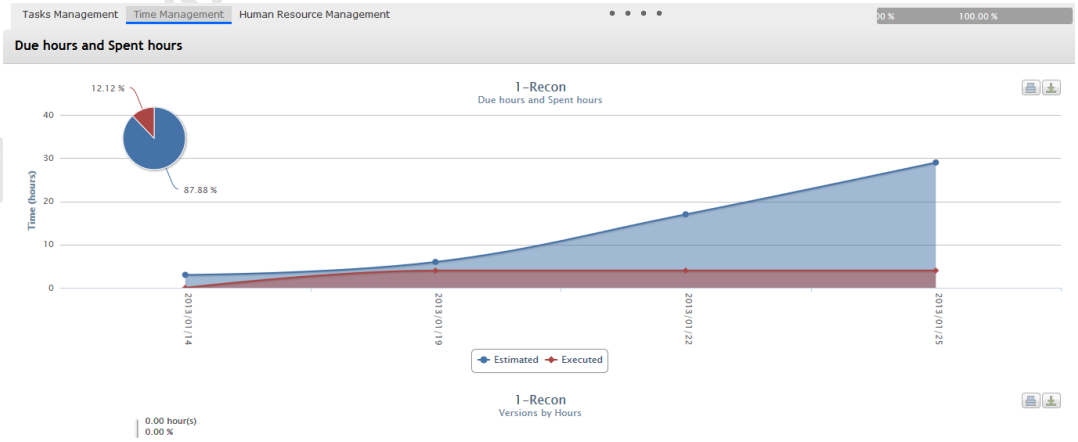

Figure 14: Monitoring & Controlling Time Management

The *Human Resources Management* view allows project managers to quickly discern issues by assignee, allowing a quick overview of which members are producing

and resolving their task, and who is carrying the load. As seen in Figure 15, Barney is clearly pulling his weight as part of the pen-test team.

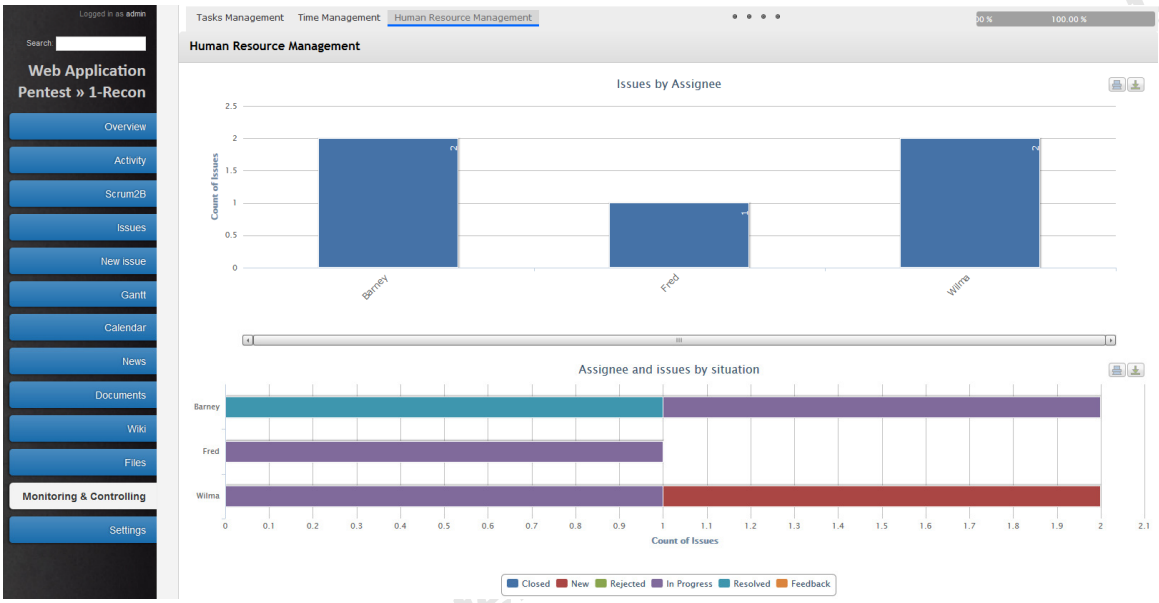

Figure 15: Monitoring & Controlling Human Resources Management

Each phase of a penetration test can be facilitated and coordinated conveniently with Redmine and the additional plugins discussed herein. When adding Scrum capabilities (virtual Scrum Board) along with dash-boarding for Monitoring & Controlling, Redmine serves as a strong project aide for project managers and members alike.

5. Conclusion

While project management includes a certain amount of tedium, Redmine configured with the aforementioned plugins allows for a refreshing, dynamic approach to the overall secure project management lifecycle. Configured as discussed, Redmine can be utilized in a manner that protects sensitive data such as pen-test findings and workflow. While no system is ever absolutely secure (a serious Ruby on Rails SQL injection flaw was disclosed as this paper was written), the appropriate hardening steps can help ensure enhanced protection. Steady maintenance and diligence will also serve you well. The convenience of an implementation such as TurnKey Redmine makes

keeping the entire system up to date quite easy: apt-get update & upgrade for the OS and the Redmine Mercurial repository to keep Redmine current.

Redmine features such as LDAP integration allow for secure enterprise integration and LDAP sync means if a domain user is disabled, the account will be disabled on Redmine as soon as the next sync occurs.

Finally, when you map Redmine feature richness and flexibility, enhanced by plugins such as Scrum2b and Monitoring & Controlling, it's easy to see how well it can accentuate security-oriented projects such as penetration tests, as well as projects of all scopes and scale.

A version of a TurnKey Redmine virtual machine as discussed here will be made available to readers via the footer email address. This instance will include a web application project template, with predefined subprojects, issue categories and bugs, again as defined in the SANS 542 course. Readers will need only create users, assign dates and members, and establish access to an LDAP service.

Redmine is ready-made for penetration testing project management and when configured as described is suitable for *any* project; consider it for use with your projects as you endeavor to secure your environments in any capacity.

6. References

Heldman, K. (2009). *Pmp: Project management professional exam study guide*. (Fifth ed.). Indianapolis, IN: Sybex.

Killer, A. (2012, July 22). **guide** how to upgrade redmine to latest version (2.0.3) painlessly. Retrieved from http://www.turnkeylinux.org/forum/general/20120722/guide-how-upgrade-redmine-latest-version-203-painlessly

Massip, E. (2012, January 19). Redmine ldap authentication. Retrieved from http://www.redmine.org/projects/redmine/wiki/RedmineLDAP

Mikus, F. (2012, December 06). *Ufw*. Retrieved from https://help.ubuntu.com/community/UFW

O'Connor, T. (2013). *Violent python*. (p. 229). Walthm, MA: Syngress.

Redmine ldap integration - active directory configuration. (2010, January 23). Retrieved from http://addicted-to-it.blogspot.com/2010/01/redmine-ldap-integration-active_23.html

Schwaber, K. (2004). *Agile project management with scrum*. Redmond, WA: Microsoft Press.

Siles, R. (2012, September). Assessing and exploiting web applications with samuraiwtf. Presentation deliverd at BruCON 2012 Brucon 2012, Ghent, Belgium. Retrieved from http://voxel.dl.sourceforge.net/project/samurai/SamuraiWTF Course/SamuraiWTF Course Slides v14 - BruCON 2012.pdf

TurnKey. (2012, August 30). redmine - integrated scm & project management. Retrieved from http://www.turnkeylinux.org/redmine