Implementing Redmine for Secure Project Management

By Russ McRee – ISSA Senior Member, Puget Sound (Seattle), USA Chapter

Prerequisites/dependencies

–VMWare for this methodology or a dedicated installation platform if installed from ISO

From Redline for March's *toolsmith* to Redmine for April's, we'll change pace from hacker space to the realm of secure project management. Following is a shortened version of a much longer Redmine study written for the SANS Reading Room[1] as part of graduate school requirements and released jointly with ISSA.

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. We'll 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.

From the TurnKey Redmine webpage: "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."[2] 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, 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."[3] 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 project manager (PM) or team lead.

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. 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, 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. 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 the current version of Redmine was 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 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.

1 http://bit.ly/YBgjTU.

2 http://www.turnkeylinux.org/redmine.

3 Gallagher, Jeffries, and Landauer (2006). *Hunting Security Bugs*, Redmond, WA: Microsoft Press.

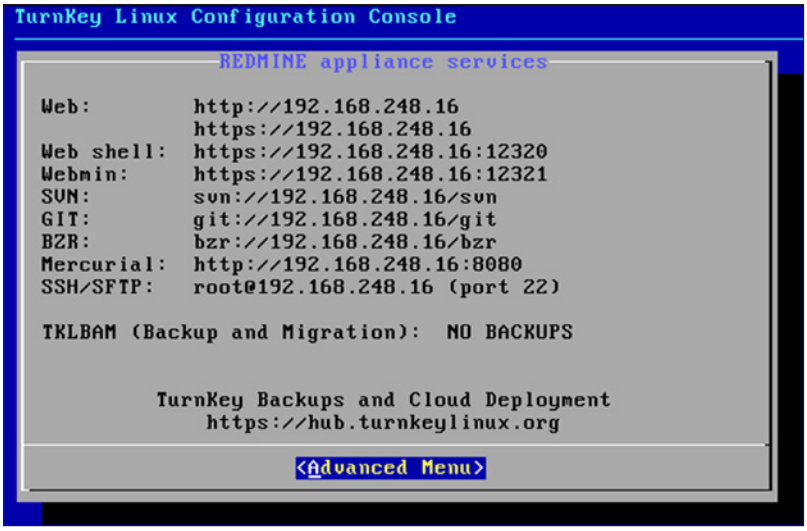


Figure 1 – TurnKey Linux configuration console

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.

Update the Redmine version from a command prompt on the Redmine server as follows:[4]

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

4 http://www.turnkeylinux.org/forum/general/20120722/guide-how-upgrade-redmine-latest-version-203-painlessly.

```
1  Name      = REDMINE
2  Host      = 192.168.248.248
3  Port      = 389
4  LDAPS     = no
5  Account   = REDMINE\redminer
6  Password  = <password>
7  Base DN   = DC=REDMINE,DC=local
8  On-the-fly user creation = no
9  Attributes
10   Login     = sAMAccountName
11   Firstname = givenName
12   Lastname  = sn
13   Email     = mail
```

Figure 2 – LDAP configuration

Run the script /var/www/redmine/script/ about to confirm the version upgrade.

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 use 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[5] via the Redmine WIKI, but refer to the following example configuration as successfully utilized for this research seen in figure 2.

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.

Refer to the SANS Reading Room version regarding additional installation and hardening steps and don't skip! These are important:

- Installation
  - Pixel Cookers theme for a streamlined, tech-centric look as well as the
  - Email settings
  - Plugin installation (Ldap Sync, Scrum2B, Screenshot, Monitoring & Controlling)
- Hardening
  - Disable unnecessary services
  - Tighten down SSH
  - Restrict Redmine web access to HTTPS only
  - Implement UFW (the uncomplicated firewall)

Engaging with Redmine

Following is a step-by-step description of a penetration testing engagement where Redmine is utilized to provide project support for a team of three.

5 http://www.redmine.org/projects/redmine/wiki/RedmineLDAP.

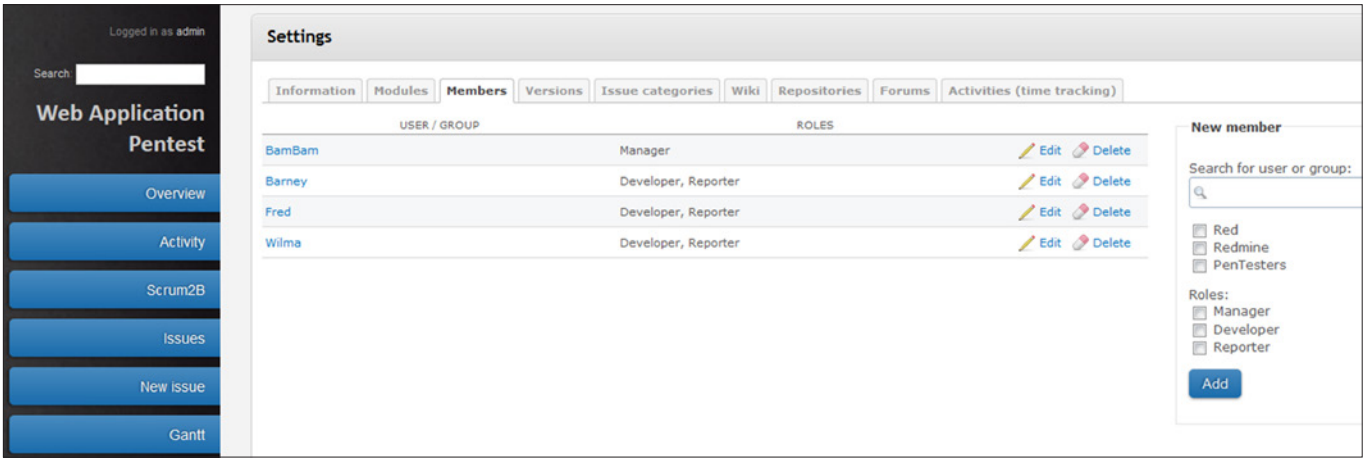


Figure 3 – Pen-test project members

The first and most important steps to undertake is 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 roles. 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. Via the *Administration* menu, under *Projects*, create a project called Web Application Pentest. The activities related to this project are drawn directly from tasks outlined in the SANS 542: Web App Penetration Testing and Ethical Hacking course as well as the Samurai Web Testing Framework.[6] 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*.

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 3.

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

6 http://sourceforge.net/projects/samurai/files/SamuraiWTF%20Course/SamuraiWTF%20Course%20Slides%20v14%20-%20BruCON%202012.pdf/download.

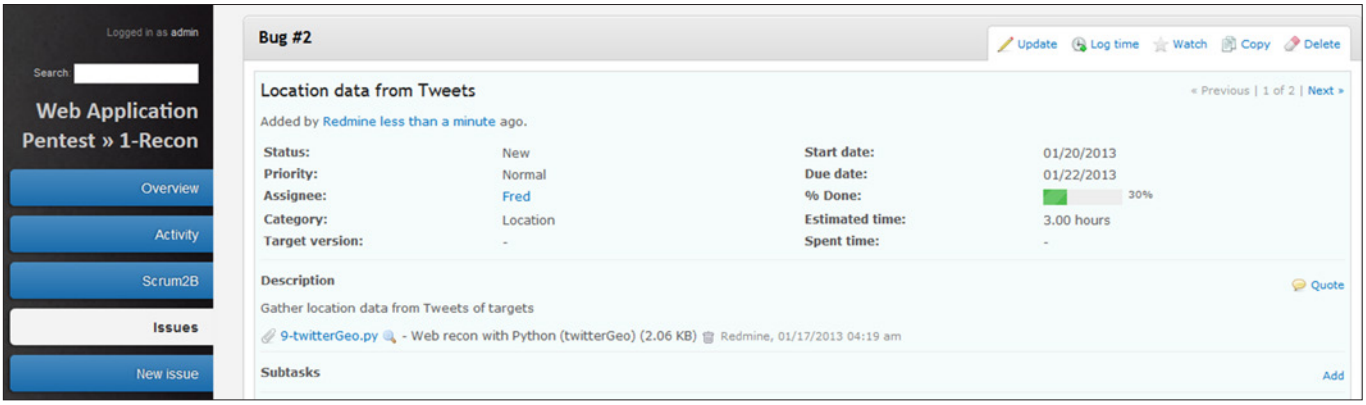
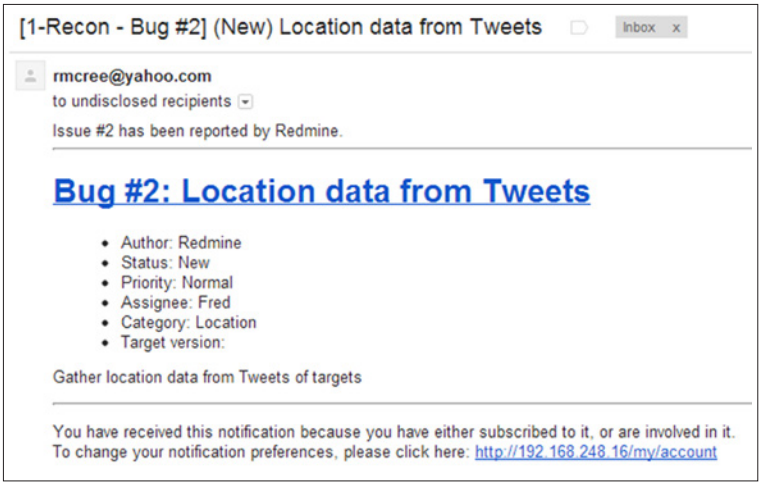


Figure 4 – Bug (task) assigned to Fred, with helper code


[1-Recon - Bug #2] (New) Location data from Tweets

Inbox x

rmcree@yahoo.com
to undisclosed recipients

Issue #2 has been reported by Redmine.

Bug #2: Location data from Tweets

- Author: Redmine
- Status: New
- Priority: Normal
- Assignee: Fred
- Category: Location
- Target version:

Gather location data from Tweets of targets

You have received this notification because you have either subscribed to it, or are involved in it.
To change your notification preferences, please click here: http://192.168.248.16/my/account

Figure 5 – Email notice for bug (task) filed

Python.[7] 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 4.

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 5 was received as a function of filing the bug in figure 5.

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. Overall project time allotments as defined in the bug then track quite nicely via the Redmine Gantt functionality as seen in figure 6.

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."[8] 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 (figure 7, next page). The pentestpm user, as project manager, could adjust task's percentage of completion right from Scrum2b using the time slider.

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.

Monitoring and 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,

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

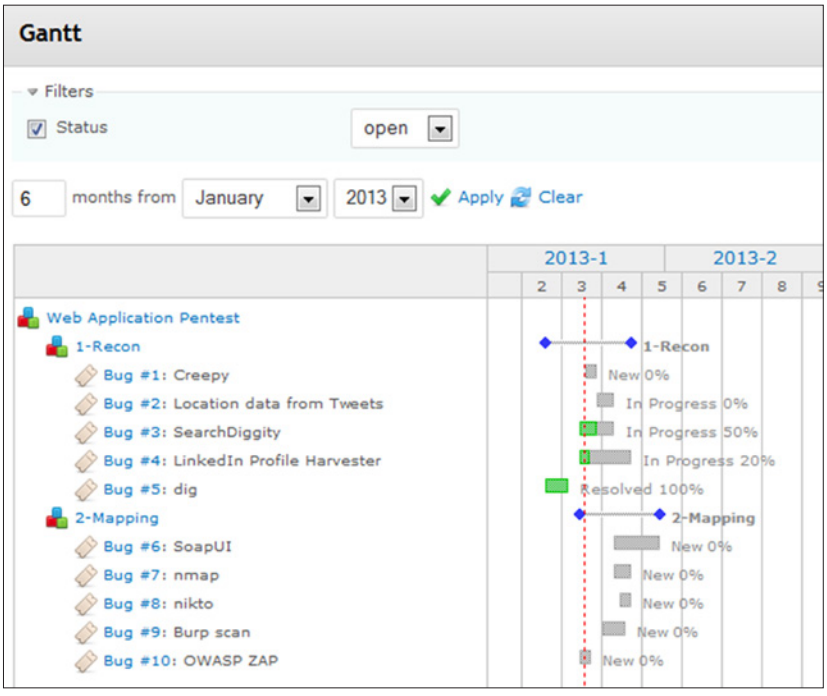


Figure 6 – Redmine Gantt functionality

8 Schwaber, K. (2004). *Agile project management with scrum.* Microsoft Press.

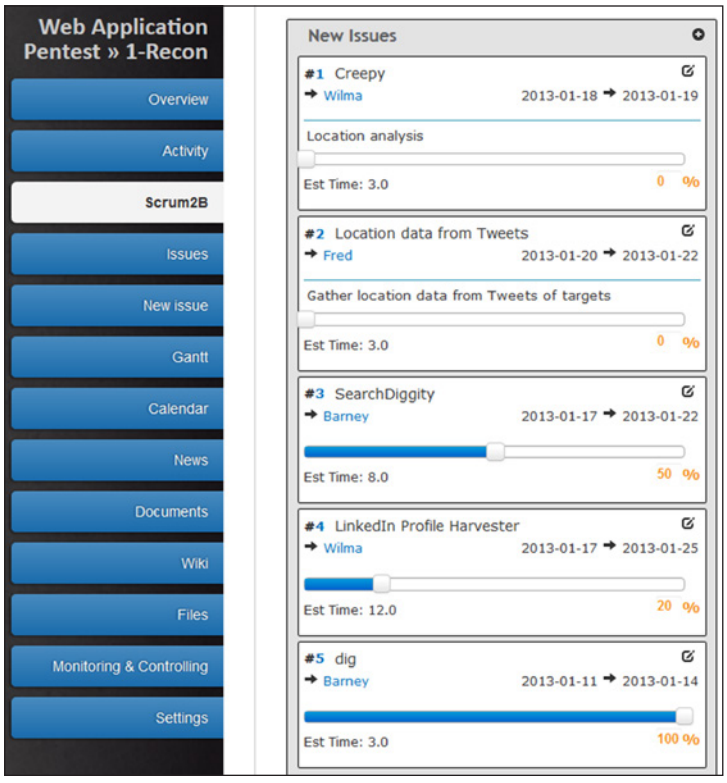


Figure 7 – Scrum2b Scrum Board for pen-testers

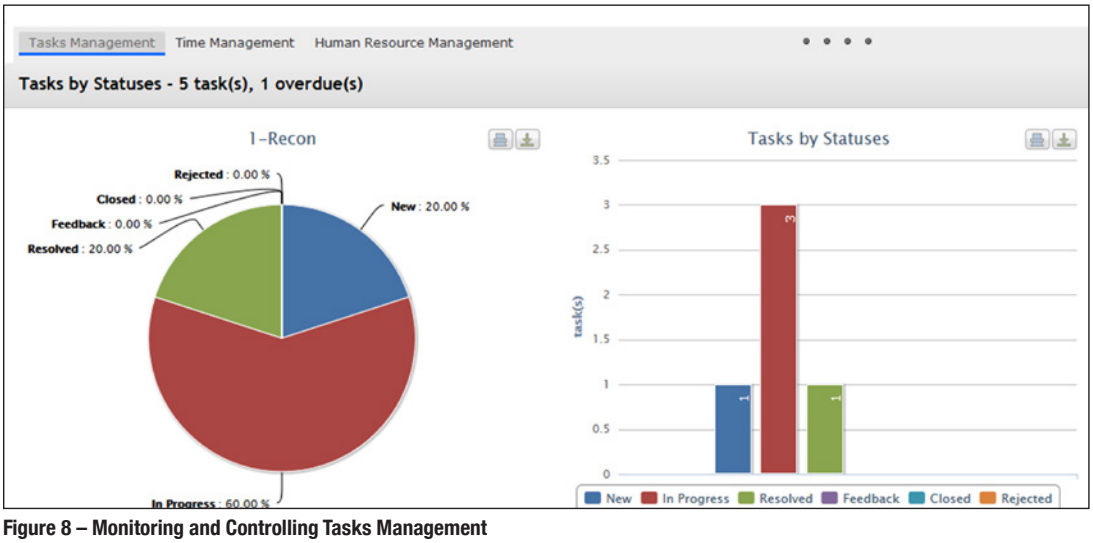


Figure 8 – Monitoring and Controlling Tasks Management

and others."[9] 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 8, represents the *Recon* phase of a pen-test.

Refer again to the SANS Reading Room version, page 17, for more regarding time and human resources management with the Redmine Monitoring & Controlling plugin.

In conclusion

Project management includes a certain amount of tedium, but Redmine configured with the aforementioned plugins allows for a refreshing, dynamic approach to the overall secure project management life cycle. 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.

A version of a TurnKey Redmine virtual machine as discussed here will be made available to readers via the HolisticInfoSec Skydrive.[10] 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.

Ping me via email if you have questions or suggestions for topic via russ at holisticinfosec dot org or hit me on Twitter @holisticinfosec.

Cheers…until next month.

About the Author

Russ McRee manages the Security Analytics team (security incident management, penetration testing, monitoring) for Microsoft's Online Services Security & Compliance organization. In addition to toolsmith, he's written for numerous other publications, speaks regularly at events such as DEFCON, Black Hat, and RSA, and is a SANS Internet Storm Center handler. As an advocate for a holistic approach to the practice of information assurance Russ maintains holisticinfosec.org. He serves in the Washington State Guard as the Cybersecurity Advisor to the Washington Military Department. Reach him at russ at holisticinfosec dot org or @holisticinfosec.

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

10 http://sdrv.ms/YQLAOA.