Preface

This packet contains information related to the development of the Real-time Network Analysis system (ReTiNA). This system was developed for use with the Cyberstorm hacking competition as a part of the Senior Capstone 2010 class at Louisiana Tech University. The goal of ReTiNA is to provide users with a real-time situational awareness of network activity, including active network nodes and network attack information. This system was built by the Cyberstorm Real-time Network Attack and Node Detection team, led by Michael King. The other members of this time were Delvin Jackson, Ryan Lockwood, Ryan Rollins, and Daniel Sawyer, with advisors Justin Poole and Dr. Jean Gourd.

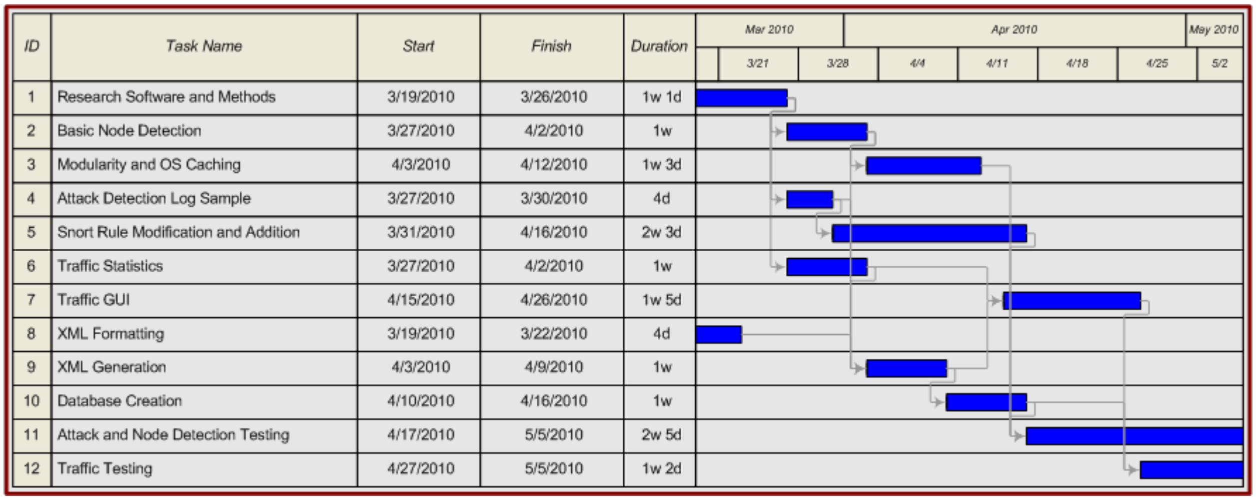
This packet first contains a timeline for the project, followed by the task trackers for each iteration of the project. Finally, the packet contains the Software Requirements Specification document, which covers the system features and provides an overview of the development cycle.

Table of Contents

List of Tables

Timeline

Meeting the deadline was the most pressing constraint of this project. Our system had to be working for the Cyberstorm competition where it would actually be used and presented before an audience. As such, time management was a prime aspect of the project's managements. Figure 1 shows a Gantt chart for this project that gives a broad overview of the project's scheduling. Due to the modular approach used to develop this system, the dependencies in the Gantt chart can be difficult to follow. Appendix A of the Software Requirements Specification document contains a PERT chart where the dependencies can be more easily tracked.



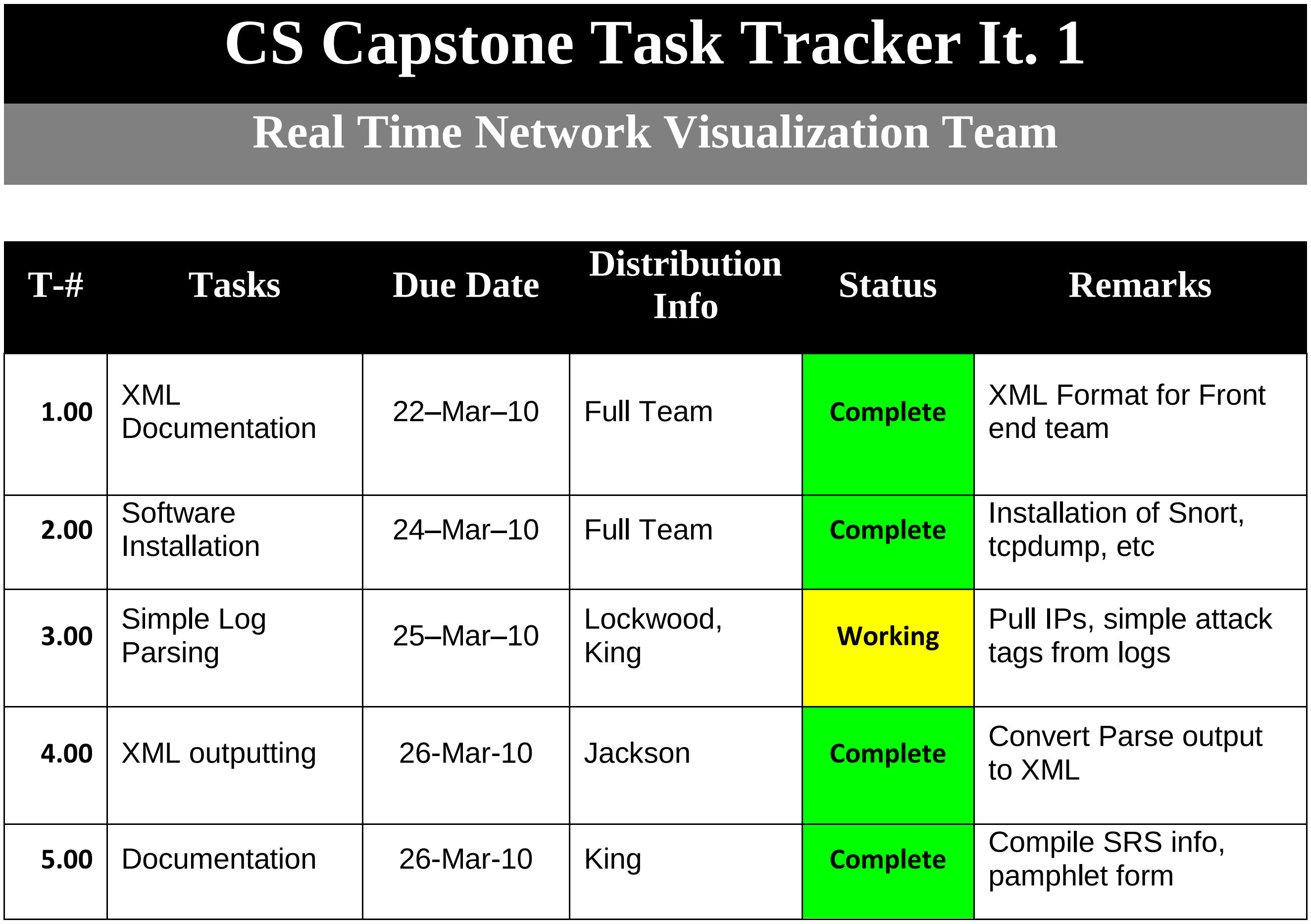
ID	Task Name	Start	Finish	Duration
1	Research Software and Methods	3/19/2010	3/26/2010	1w 1d
2	Basic Node Detection	3/27/2010	4/2/2010	1w
3	Modularity and OS Caching	4/3/2010	4/12/2010	1w 3d
4	Attack Detection Log Sample	3/27/2010	3/30/2010	4d
5	Snort Rule Modification and Addition	3/31/2010	4/16/2010	2w 3d
6	Traffic Statistics	3/27/2010	4/2/2010	1w
7	Traffic GUI	4/15/2010	4/26/2010	1w 5d
8	XML Formatting	3/19/2010	3/22/2010	4d
9	XML Generation	4/3/2010	4/9/2010	1w
10	Database Creation	4/10/2010	4/16/2010	1w
11	Attack and Node Detection Testing	4/17/2010	5/5/2010	2w 5d
12	Traffic Testing	4/27/2010	5/5/2010	1w 2d

Figure 1. ReTiNA Gantt Chart

Iteration Tracking

The Cyberstorm project used an iterative software development module to create weekly milestones toward completion. The ReTiNA system used a modular development method as well due to the separate modules required of our system. The modules were split into Node Detection, Attack Detection, Communications Interface, and Traffic Statistics. Each module was developed independently by different team members, with members helping on other's modules if they had slack time. To fit this in the iterative model used for the entire project, each module was expected to meet certain milestones within their iteration.

Figures 2,3,4 & 5 show our Iteration Task Trackers for each of the week-long project iterations. Figure 6 shows the Issue Tracker for what should have been just a testing phase, however, there was still some development occurring in this phase on the Traffic Statistics module due to scope creep. A detailed explanation of the Task and Issue Trackers and the ReTiNA development cycle can be found in the Software Requirements Specification document.

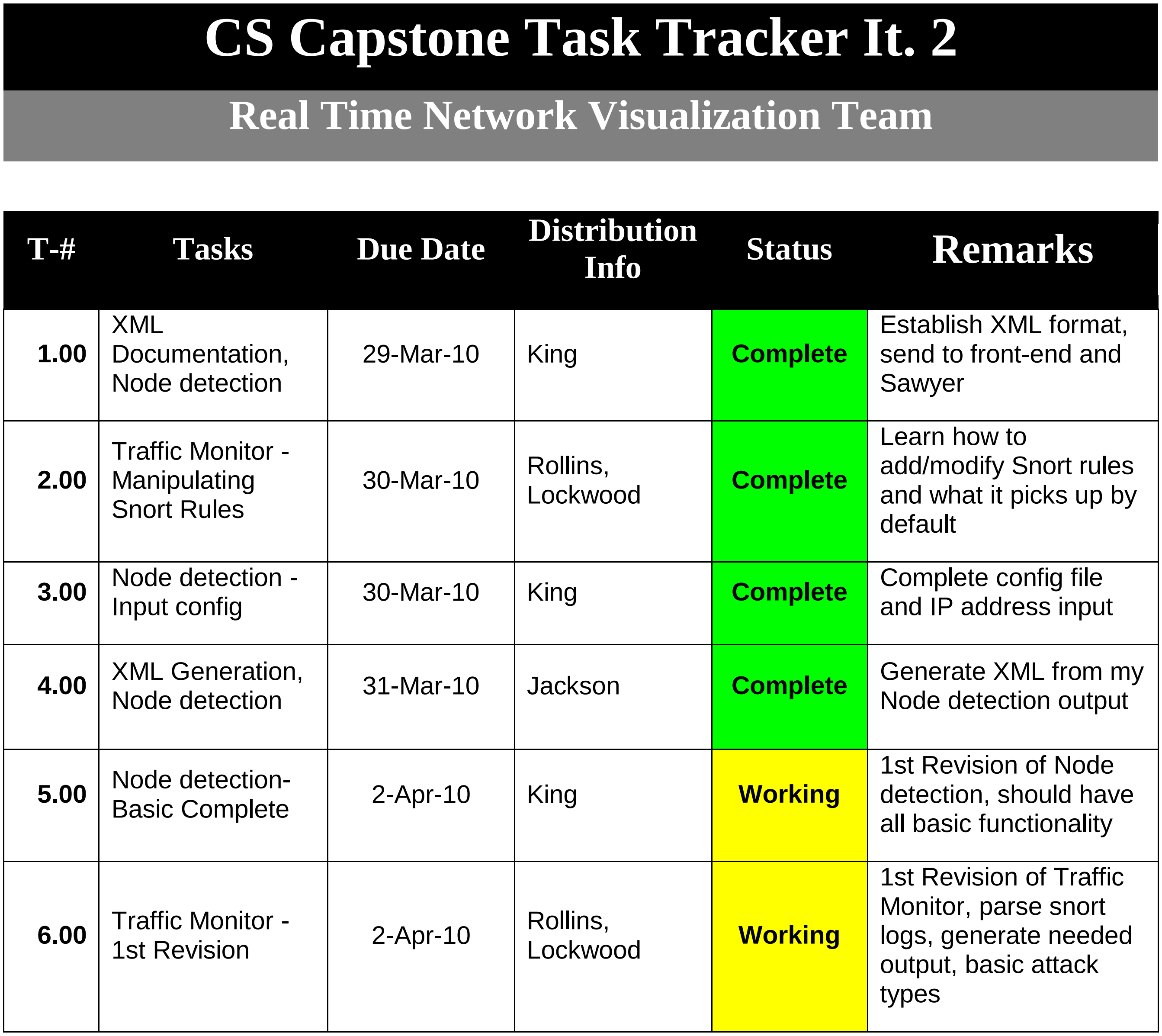


CS Capstone Task Tracker It. 1

Real Time Network Visualization Team

T-#	Tasks	Due Date	Distribution Info	Status	Remarks
1.00	XML Documentation	22–Mar–10	Full Team	Complete	XML Format for Front end team
2.00	Software Installation	24–Mar–10	Full Team	Complete	Installation of Snort, tcpdump, etc
3.00	Simple Log Parsing	25–Mar–10	Lockwood, King	Working	Pull IPs, simple attack tags from logs
4.00	XML outputting	26-Mar-10	Jackson	Complete	Convert Parse output to XML
5.00	Documentation	26-Mar-10	King	Complete	Compile SRS info, pamphlet form

Figure 2. Iteration 1 Task Tracker

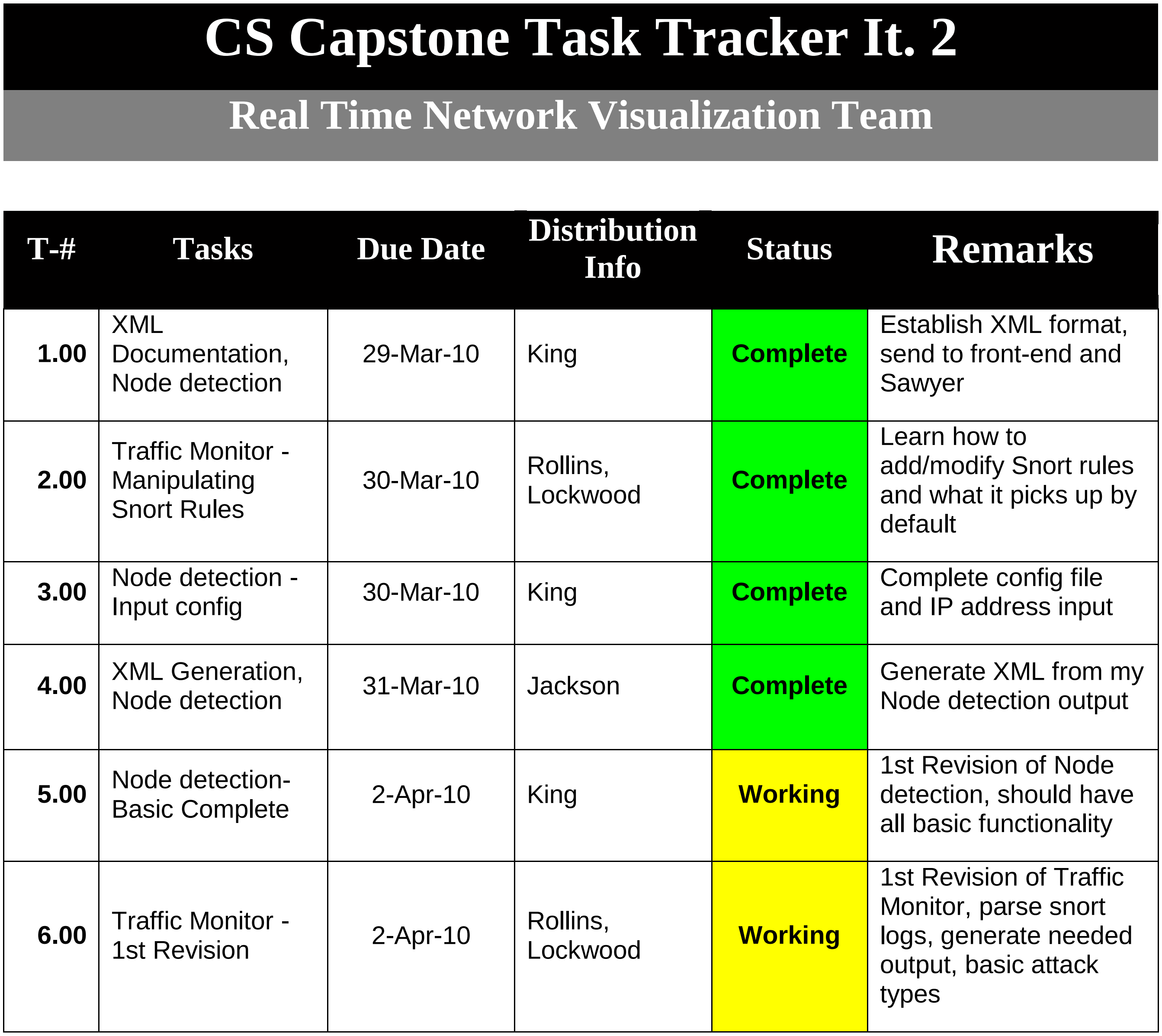


CS Capstone Task Tracker It. 2

Real Time Network Visualization Team

T-#	Tasks	Due Date	Distribution Info	Status	Remarks
1.00	XML Documentation, Node detection	29-Mar-10	King	Complete	Establish XML format, send to front-end and Sawyer
2.00	Traffic Monitor - Manipulating Snort Rules	30-Mar-10	Rollins, Lockwood	Complete	Learn how to add/modify Snort rules and what it picks up by default
3.00	Node detection - Input config	30-Mar-10	King	Complete	Complete config file and IP address input
4.00	XML Generation, Node detection	31-Mar-10	Jackson	Complete	Generate XML from my Node detection output
5.00	Node detection- Basic Complete	2-Apr-10	King	Working	1st Revision of Node detection, should have all basic functionality
6.00	Traffic Monitor - 1st Revision	2-Apr-10	Rollins, Lockwood	Working	1st Revision of Traffic Monitor, parse snort logs, generate needed output, basic attack types

Figure 3. Iteration 2 Task Tracker

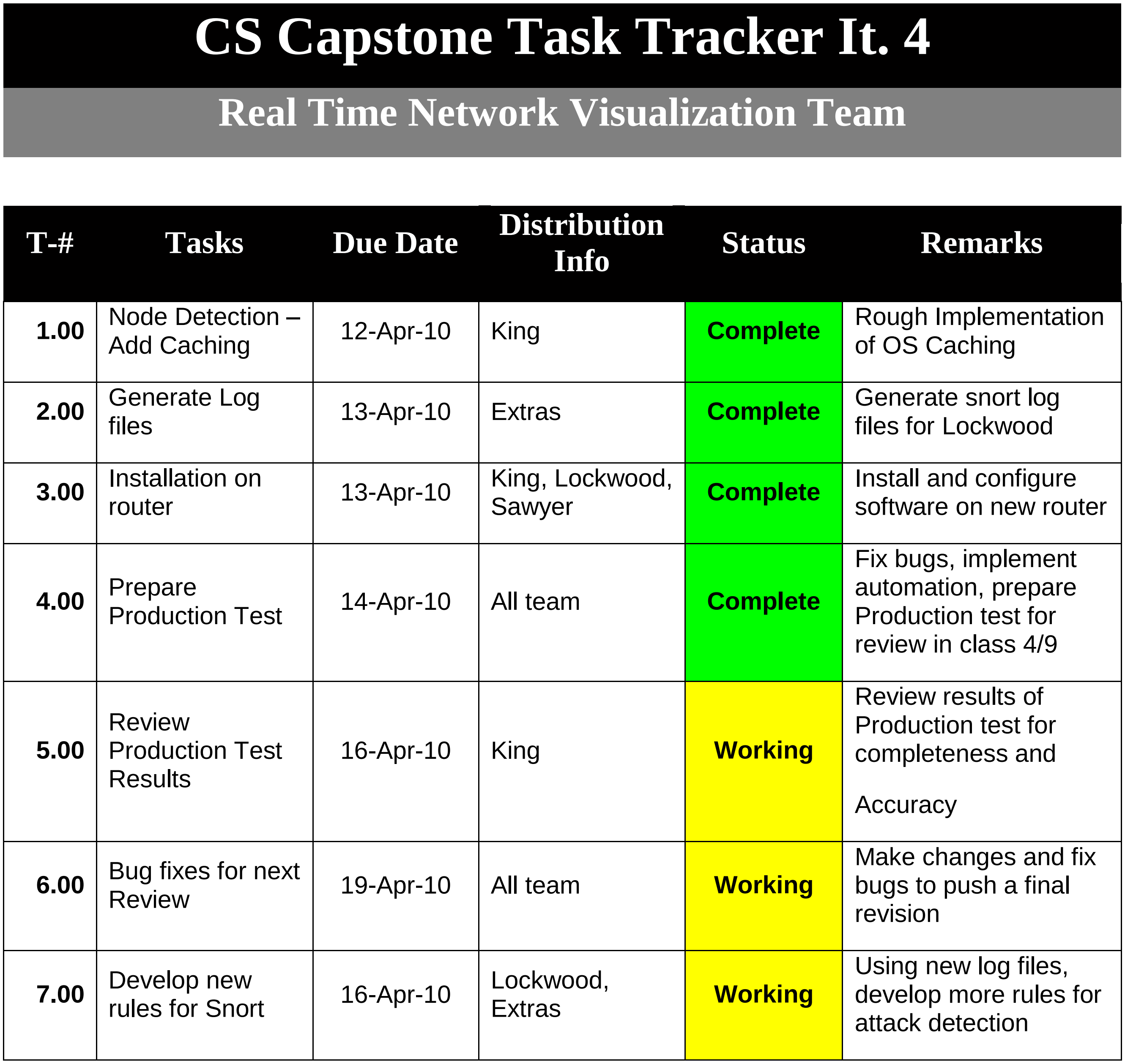


CS Capstone Task Tracker It. 2

Real Time Network Visualization Team

T-#	Tasks	Due Date	Distribution Info	Status	Remarks
1.00	XML Documentation, Node detection	29-Mar-10	King	**Complete**	Establish XML format, send to front-end and Sawyer
2.00	Traffic Monitor - Manipulating Snort Rules	30-Mar-10	Rollins, Lockwood	**Complete**	Learn how to add/modify Snort rules and what it picks up by default
3.00	Node detection - Input config	30-Mar-10	King	**Complete**	Complete config file and IP address input
4.00	XML Generation, Node detection	31-Mar-10	Jackson	**Complete**	Generate XML from my Node detection output
5.00	Node detection- Basic Complete	2-Apr-10	King	**Working**	1st Revision of Node detection, should have all basic functionality
6.00	Traffic Monitor - 1st Revision	2-Apr-10	Rollins, Lockwood	**Working**	1st Revision of Traffic Monitor, parse snort logs, generate needed output, basic attack types

Figure 4. Iteration 3 Task Tracker

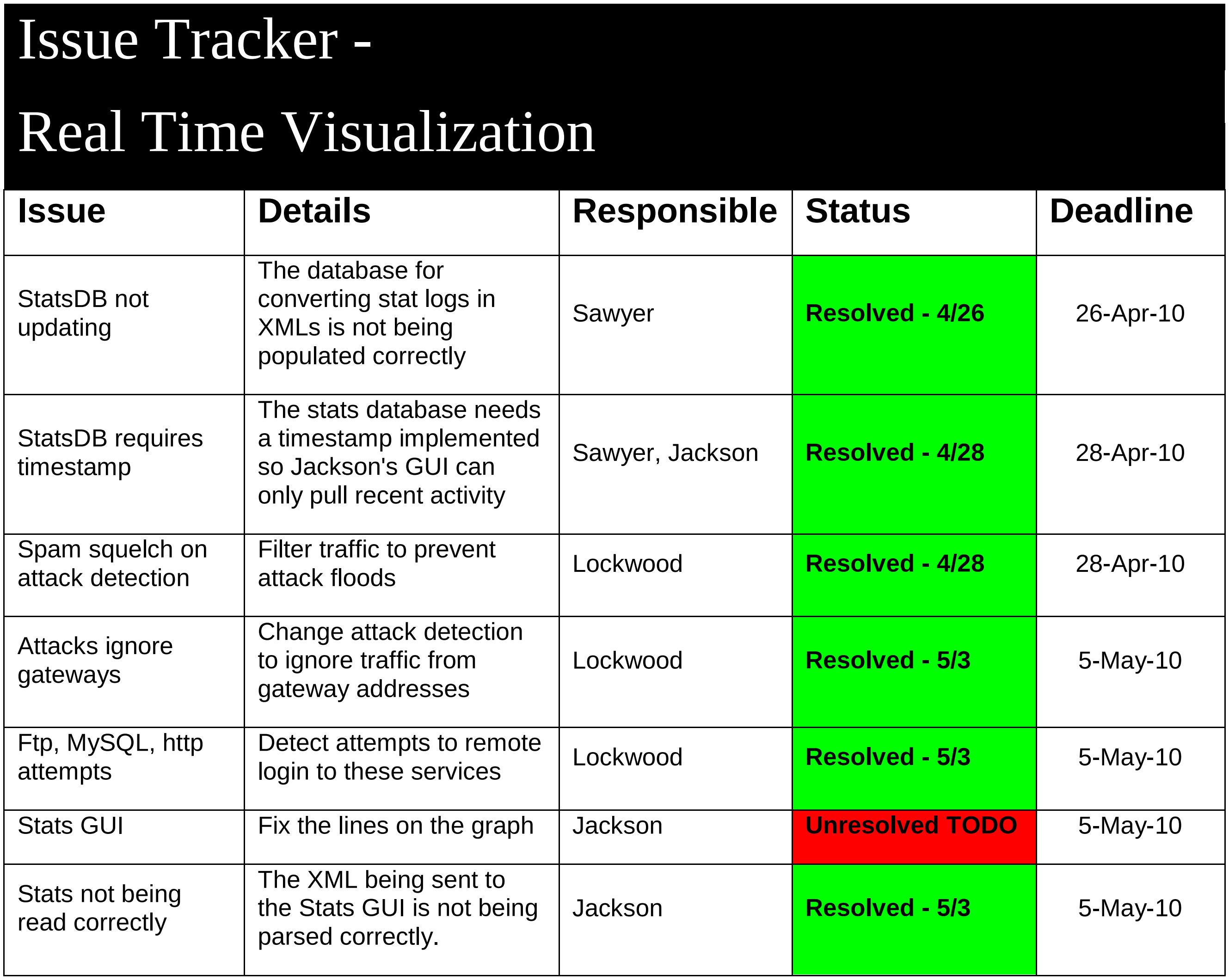


CS Capstone Task Tracker It. 4

Real Time Network Visualization Team

T-#	Tasks	Due Date	Distribution Info	Status	Remarks
1.00	Node Detection – Add Caching	12-Apr-10	King	Complete	Rough Implementation of OS Caching
2.00	Generate Log files	13-Apr-10	Extras	Complete	Generate snort log files for Lockwood
3.00	Installation on router	13-Apr-10	King, Lockwood, Sawyer	Complete	Install and configure software on new router
4.00	Prepare Production Test	14-Apr-10	All team	Complete	Fix bugs, implement automation, prepare Production test for review in class 4/9
5.00	Review Production Test Results	16-Apr-10	King	Working	Review results of Production test for completeness and Accuracy
6.00	Bug fixes for next Review	19-Apr-10	All team	Working	Make changes and fix bugs to push a final revision
7.00	Develop new rules for Snort	16-Apr-10	Lockwood, Extras	Working	Using new log files, develop more rules for attack detection

Figure 5. Iteration 4 Task Tracker



Issue Tracker -

Real Time Visualization

Issue	Details	Responsible	Status	Deadline
StatsDB not updating	The database for converting stat logs in XMLs is not being populated correctly	Sawyer	**Resolved - 4/26**	26-Apr-10
StatsDB requires timestamp	The stats database needs a timestamp implemented so Jackson's GUI can only pull recent activity	Sawyer, Jackson	**Resolved - 4/28**	28-Apr-10
Spam squelch on attack detection	Filter traffic to prevent attack floods	Lockwood	**Resolved - 4/28**	28-Apr-10
Attacks ignore gateways	Change attack detection to ignore traffic from gateway addresses	Lockwood	**Resolved - 5/3**	5-May-10
Ftp, MySQL, http attempts	Detect attempts to remote login to these services	Lockwood	**Resolved - 5/3**	5-May-10
Stats GUI	Fix the lines on the graph	Jackson	**Unresolved TODO**	5-May-10
Stats not being read correctly	The XML being sent to the Stats GUI is not being parsed correctly.	Jackson	**Resolved - 5/3**	5-May-10

Figure 6. Issue Tracker

Software Requirements Specification Document

PLACEHOLDER!!!!!!

TO BE REPLACED BY SRS DOCUMENT