TALLINN UNIVERSITY OF TECHNOLOGY
Faculty of Information Technology
Department of Computer Science
TUT Centre for Digital Forensics and Cyber Security

ITC70LT

Karl Kristjan Raik 144007IVCM

IMPROVING WEB ATTACK CAMPAIGN OVERVIEW IN CYBER DEFENSE EXERCISES

Master's thesis

Supervisors
Supervisor Elar Lang, MSc
Supervisor Rain Ottis, PhD

Tallinn 2016

Declaration

I declare that this thesis is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Karl Kristjan Raik

May 25, 2016

....
(Signature)

Abstract

Cyber Defense Exercises are an increasingly popular way to raise awareness and technical expertise for specialists about networks and cyberspace. Exercises that consist of one side defending and other attacking networks, use feedback as one learning method. Defending and attacking teams are referenced as Blue and Red Team accordingly and focus of this thesis is on Web application attack sub-team. Web Team ideally documents all their actions and presents those to Blue Teams after the exercise. This way Blue Teams are able to learn from mistakes they made.

Detailed feedback about exercise events helps Blue Teams reinforce learning goals. However Web Team is occupied with conducting and reporting attacks on Blue Teams and their priority is to provide equal attention to each Blue Team. Main problem is that Web Team is unable to provide feedback with enough detail about their campaign, due to heavy workload.

Main goal for the thesis is to collect and visualize important data from Cyber Defense Exercise environment to provide detailed information about Web Team campaign. Using the detailed information, Web Team leader can create feedback to Blue Teams. For this, a proof of concept framework was created, for Locked Shields 2016 Cyber Defense Exercise. The framework collects data from Cyber Defense Exercise environment, processes and visualizes it for overview of Web Team campaign.

Main outcome of the framework is a timeline displaying web attacks, Web Team objective data and web services availability and functionality tests. The outcome is used for creating the Locked Shields 2016 After Action Report.

The thesis is in English and contains 71 pages of text, 7 chapters, 22 figures.

Annotatsioon

ÜLEVAATE PARANDAMINE VEEBIRÜNNETE KAMPAANIAST KÜBERKAITSE ÕPPUSTEL

Küberkaitse õppused on populaarsust koguv meetod arendada spetsialistide teadlikust ja tehnilist taset võrkude ja küberruumi kohta. Õppused, mis koosnevad võrku ründavast ja kaitsevast poolest, kasutavad tagasisidet ühe õppimismeetodina. Kaitsevat ja ründavat meeskonnda nimetatakse vastavalt Siniseks ja Punaseks meeskonnaks ning töö keskendub Veebirünnete alam-meeskonnale. Ideaalis dokumenteerib Punane meeskond kõik oma tegevused ja esitab need peale õppust Sinisele meeskonnale. Seeläbi saavad Sinised meeskonnad õppida oma vigadest.

Detailne tagasiside õppuse sündmuste kohta aitab Sinistel meeskondadel kindlustada õpieesmärke. Kuid, Punane meeskond on hõivatud rünnete sooritamise ja raporteerimisega ning nende peamine eesmärk on pakkuda võrdset tähelepanu kõigile Sinistele meeskondadele. Seetõttu on peamine probleem, et Punane meeskond ei suuda, suure koormuse tõttu, pakkuda piisava detailsusega tagasisidet oma tegevuse kohta.

Käesoleva töö peamine eesmärk on koondada ja visualiseerida oluliseda andmed Küberkaitseõppuse keskkonnas ning pakkuda detailest infot Veebi rünnete kampaania kohta. Kasutades detailset infot, saab Veebirünnete meeskonna pealik koostadad tagasiside Sinistele meeskondadele. Selle tarbeks loodi kontseptsiooni tõendus raamistik Locked Shields 2016 õppusele. Raamistik koondab andmed õppuse keskkonnas, töötleb need ning visualiseerib veebirünnete kampaaniast ülevaate saamiseks.

Käesolev töö pakub probleemile lahenduse veebirünnete kontekstis, tuvastades vajalikud andmed ja luues raamistiku veebirakenduste rünnete ülevaate loomiseks. Kogutud lähteandmetest loodi ajatelg, mis sisaldas veebiründeid, Punase meeskonna ülessannete täitmise infot, veebiteenuste käideldavuse ja funktsionaalsuse testide andmeid. Käesoleva tööga loodud veebirünnete kampaania infot kasutati Locked Shields 2016 tagasiside raporti koostamiseks.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 71 leheküljel, 7 peatükki, 22 joonist.

List of Acronyms

ENISA European Union Agency for Network and Information Security

NATO North Atlantic Treaty Organization

CCDCOE Cooperative Cyber Defence Centre of Excellence

CDX Cyber Defense eXercise

AAR After Action Report

RDP Remote Desktop Protcol

HTTP HyperText Transfer Protocol

HTTPS HTTP Secure

APT Advanced Persistent Threat

JSON JavaScript Object Notation

PDF Portable Document Format

CEO Chief Executive Officer

CRX Cyber Range eXercise

SSH Secure Shell

Contents

List of Figures

1. Introduction

Information and communication technologies have become increasingly important to the world. In Estonia, for example, it is possible to start a company, do taxes and banking online [1]. Cyberspace is not used only for economical, political and social activities. States have also started to regard it as a fifth operational domain. Together with land, sea, air and space, cyber has become a domain for military operations [2]. A number of states have published cyber security strategies incorporating defensive and offensive intentions in cyber [3]. Importance and interest in cyberspace demands well trained specialists who can operate in the new domain.

Cyber Defense eXercises (CDXs) are beginning to rise in popularity as a good way to learn needed skills. ISO 22389:2013 standard defines exercise as follows: “Exercises are an important management tool intended to identify gaps and areas for improvement as well as to determine the effectiveness of response and recovery strategies.” [4]. Those different use-cases and goals for exercises described in the standard can be adapted to CDXs as well. There are several types of CDXs conducted in the world for validating policies and procedures, testing coordination, communication and response during cyber incidents. In European Union Agency for Network and Information Security (ENISA) report on Cyber Exercises, published in 2015, 8 different types were identified, including capture the flag, table-top, simulation and Red Team/Blue Team [5].

This thesis focuses on technical Red Team/Blue Team exercises, which can be conducted at different scales and have different goals. In those exercises, active attack and defense scenario is played out in environment as close to reality as possible [6]. Red Team is acting as an Advanced Persistent Threat (APT) and engaging with Blue Teams, simulating attacks to their network. Blue Teams must give their best to defend it under stress [6].

From 2010 NATO Cooperative Cyber Defence Centre of Excellence (CCDCOE)[1] has or-

[1] https://ccdcoe.org/about-us.html

ganized an international real-time technical cyber defense exercise called Locked Shields[2] to teach participants the skills needed to respond to actual cyber attack [6]. Looking at the increasing trend, serious cyber games are becoming more important and efficient tools are needed to focus more on educational purposes [5].

Technical exercises, like Locked Shields, are centered around Blue Teams. As exercise target audience, they are put under stress and need to keep control over a network infiltrated by Red Team [6]. "Defence is the focus and scope of the exercise, with Blue Teams being tasked to maintain their networks and services under intense pressure." [6].

Locked Shields is the largest and most advanced live-fire Cyber Defense Exercise in the world and its training audience has been increasing every year [7] [6]. In this real-time exercise, where Red Team has to keep pressure on all participating Blue Teams equally, has grown to twenty Blue Teams for year 2016 [7]. Red Team tasks involve carrying out attacks, reporting results and changing tactics in case of active defense [8]. All the work must be done to all Blue Teams in fixed time windows, which leads to the main problem handled in this thesis.

1.1 Problem Statement

This section introduces the problem this thesis is proposing the solution for and how it is relevant in example of Locked Shields. Although the focus of the thesis is on Locked Shields, the problems here are relevant to any CDX with large audience.

In Locked Shields, Red Team is split into three categories: Network, Client-side and Web sub-teams [9]. Each of those teams have separate objectives in Blue Teams networks and use different methods to achieve them. These objectives are a set of tasks, Red Team must complete, that support the game scenario, include target to attack and proof needed. Every successful objective gives negative score to Blue Teams [8]. For example, one of the objectives was to deface[3] the target site and take a screenshot as a proof.

Scope of the thesis is only Web sub-team, further referenced as Web Team, whose goal is to attack web applications Blue Teams maintain. Web Team uses mainly custom scripts and browser, with manual techniques, to exploit vulnerabilities in web services [8]. In year 2016 there were fourteen people in Web Team and nine targets in simulated networks

[2] In 2010 CCDCOE organized Locked Shields predecessor Baltic Cyber Shield

[3] An attack changing the visual appearance of the web page

under Blue Teams management [8]. All Blue Teams had to manage identical networks created for the exercise.

After the exercise Red Team gives feedback about their actions to Blue Teams, which further reinforces the learning goals [9]. ENISA describes the need for feedback as: “A critical part of exercise knowledge is to develop good methods for critical reflection and after-action reports — a part that ties directly into the goal of having as many actors as possible record their activities in the dataset, or indeed to make use of the dataset at all.” [5]. However, with already large workload there is little time for detailed documenting to give valuable feedback.

In Locked Shields 2016, Web Team was organized so, that one or two members focused on one web service for all twenty Blue Teams [8]. Objectives for each service had to be completed in a fixed time window, that is defined as a phase. Locked Shields main execution is a two day event, each day has 8 hours of game time [7]. Both days are split into two phases and each objective is repeatable during the phases [8]. When considering the number of objectives, services and Blue Teams, Web Team must report over 700 events in a two day exercise [8].

Reporting a single objective takes several actions from Web Team member [8]. First, the objective must be activated before any attacks are made. Then, after completing the task or reaching time limit, it must be marked successful or failed. In case of a successful attack the proof must be also submitted in predefined form.

To show the magnitude of the attack reporting, it is estimated that reporting one objective takes 1-3 minutes [8]. This time relies on reporters experience and complexity of the proof. However, as no analysis or measurements are made during previous exercises, the number only serves to show an estimation how much time Web Team deals with reporting.

When taking into account the number of events to report in Locked Shields and the worst case reporting time, Web Team as a whole spends 35 hours reporting. Although this time is distributed among all the Web Team members, the size of Web Team has its limits. Larger team size also has more overhead in form of communication.

Much of the attacks are scripted and automated, but all Blue Teams use different tactics to defend their web services. Also, Blue Teams have access to systems before execution so they can fix web services beforehand. Therefore, Web Team must find ways to circumvent the defenses using manual methods, which takes time.

With 20 Blue Teams in Locked Shields 2016, Web Team members face a challenge to conclude all actions into detailed notes. The scale of information Web Team members are able to record can be seen from redacted version of their feedback in appendix A.3. Blue Teams want a detailed feedback at the end of exercise, however, with large workload, there is little time to take detailed notes [10]. It is also difficult to recall what exactly happened because of the amount of events and teams.

Web attacks in CDX context have no uniform framework or common tools. This way it is harder for Blue Teams to identify attackers and there is also a training side for Web Team participants [8]. It should be added that in Lang's experience, the varying environment often requires to fall back on manual attack methods. Therefore every Web Team member has custom tools and methods to aid them during the exercise. For that reason, it is more difficult to gather information about web attacks.

Appendix A.7 features Locked Shields 2013 game environment, which is the only description of Locked Shields published by CCDCOE [10]. The environment has changed considerably for Locked Shields 2016, but A.7 gives an idea of the virtual network the game takes place in. In addition to web attacks, there is other important information in the exercise environment as well. For instance, all Blue Team services, which among others include ping[4], Remote Desktop Protcol (RDP) and HyperText Transfer Protocol (HTTP) for web servers, are tested every minute. This information is available in the cyber exercise environment, but it is in different formats and spread throughout the environment.

To summarize problems currently with Locked Shields Cyber Defense eXercise:

> *Detailed feedback about exercise events helps Blue Teams reinforce learning goals. However Web Team is occupied with conducting and reporting attacks on Blue Teams and their priority is to provide equal attention to each Blue Team. Main problem is that Web Team is unable to provide feedback with enough detail about their campaign, due to heavy workload.*

1.2 Main Goals

To get a detailed overview of the Web Team campaign, without adding additional tasks to Web Team members, various type data must be collected from exercise environment. Necessary data from exercise environment must be derived from Blue Team learning objectives and previous Web Team feedback. **Main goal for the thesis is to collect and**

[4] Utility, to check the connectivity of a host on the network.

visualize data from Cyber Defense eXercise environment to provide detailed information about Web Team campaign. Detailed information about Web Team campaign is important, to give Blue Teams feedback about what happened and when. For better overview, results must be visualized from the collected data.

Main goal consists of the following sub-goals:

- Identify important data for Web Team campaign overview;
- Propose data collection methods;
- Provide detailed information about Web Team campaign.

1.3 Outline

Thesis is split into seven chapters. First introduces problem and main goal of the thesis. Second gives a brief overview of Cyber Defense eXercises and third chapter reviews related works for the thesis. In fourth chapter, analysis on feedback in Cyber Defense eXercises is made and important data is identified for Web Team campaign overview. Fifth chapter proposes solutions and methods for data collection and processing in context of Locked Shields exercise. Sixth chapter evaluates the implemented solution in Locked Shields 2016 CDX execution and proposes future work. Last chapter is left for results and conclusions.

1.4 Acknowledgments

At this point the author would like to thank people who supported him. First, Elar Lang and Rain Ottis for supervising the thesis. Special thanks goes to friends and family for their support and to everyone who read the thesis and provided constructive feedback. Last, but not least, colleagues from Clarified Security for their insight and ideas for the thesis.

2. Cyber Defense Exercises

ENISA describes Cyber Exercises in their report as: “an important tool to assess the preparedness of a community against cyber crises, technology failures and critical information infrastructure incidents.” [11]. Cyber Exercises allow participating parties to prepare for cyber incidents and train skilled specialists. When incidents happen, community already has an idea how to respond and what are the procedures. Following sections introduce Locked Shields and similar technical cyber defense exercises in the world.

2.1 Locked Shields

This section describes Locked Shields Cyber Defense eXercise because the proof of concept relies on data from that particular exercise. It is based on Mehis Hakkaja’s [9] experience from the beginning of Locked Shields as Red Team leader, Elar Lang’s [8] experience as Web Team leader (interview with Web Team leader is in appendix A.1) and communication with Aare Reintam [7], Locked Shields 2016 manager. Also on authors experience in Locked Shields 2015 Estonian Blue Team and Locked Shields 2016 Web Team.

2.1.1 Locked Shields Structure

Locked Shields is a real-time network defense exercise with virtual network and fictional scenario as described in Locked Shields 2015 Executive Summary [6]. In this, two day exercise, Red Team is actively attacking Blue Team infrastructure.

Red Team’s tasks are put together as objectives, that support the game scenario. Both days are split into two phases that set the time constraints on Red Team for completing objectives. Each started objective must be “closed” in the end of each phase, which could mean successful or failed attack.

Objectives are set of tasks for Red Team and mainly achieved by exploiting vulnerabilities in Blue Team systems. Although the scenario has changed considerably, Locked Shields 2013 After Action Report [10] still offers a good overview of general scale of objectives in Locked Shields context. In general, objectives escalate towards more destructive actions. For example, Web Team starts with defacing web services and data theft. Later on, they focus on destroying applications and shutting down services. Successful completion of each objective reflects as negative score to Blue Teams and objectives are repeatable in each phase.

It is important to mention that Blue Teams' main objective is not to defend their infrastructure from all attacks completely, it is to respond to cyber attacks and maintain services [8]. In reality the Blue Teams are handed very vulnerable systems and the success of the exercise resides on how well they can continue to provide services and communicate [7] [8]. Because of Red Team's white-box[1] approach, Blue Team is generally unable to defend from all attacks.

2.1.2 Locked Shield Teams

Aside from Blue Teams, there are several different teams to manage, build and conduct the exercise. Following subsections shortly describes each team to show the scale of Locked Shields exercise.

2.1.2.1 White Team

White Team is the one assigning scores and keeping the teams competitive and playing fair [7]. All objectives completed by Red Team are scored according to points assigned to each objective. White Team controls Red Team campaign regarding pressure and also simulates the management of defended systems.

One sub-team of White Team is Media Team. They conduct injects or side tasks to test Blue Teams' communication and check situation awareness. For example, injects, where Blue Teams must comment current situation and answer media queries. Some injects are cooperated with Red Teams to check if Blue Teams are aware of the attacks and information leakage from their systems.

[1] Red Team is included in system development

Another sub-team is User Simulation Team that helps Client-side Team to infiltrate Blue Team system [7]. User Simulation Team acts as an ordinary user of the systems Blue Teams must manage. Their help includes clicking on links with malicious content that Client-side sub-team sends.

2.1.2.2 Green Team

Green Team is responsible for technical game infrastructure, which is divided into two parts – core infrastructure, where virtualized environment is built and game network, that Blue Teams must defend [7]. Administration and development of whole infrastructure is tasked to Green Team.

Along with those tasks, Green Team is also continuously testing all Blue Team's services for scoring during the exercise [7]. Those testing outputs serve as one dimension in Web Team campaign overview.

2.1.2.3 Yellow Team

Yellow Team provides situational awareness of the exercise, for example information about stress levels and discovered indicators of compromise [7]. This means that Yellow Team provides overview of Blue Team discoveries and comprehension of the game. Situational overview can be used to see how Blue Teams perceive the exercise.

2.1.2.4 Red Team

In Locked Shields Red Team is divided into three sub teams with different technical objectives – Web Application, Client-side and Network attacks sub-teams [9]. Network attacks sub-team is responsible for attacking Blue Team network infrastructure. Client-side attack sub-team attempts to compromise Blue Team workstations and domain controllers. They must bypass Blue Team firewalls inside Linux and Windows workstations. For initial compromise they have the help of User Simulation Team.

Web Team is responsible for attacking Blue Team web services. Attacks are carried out in "silent mode", which means that number of unnecessary requests to the target machine are kept to minimum. Because the vulnerabilities are known beforehand, there is no need to scan the network and Web Team can focus on attacking the Blue Teams. Also the growing size of the exercise and constrict time frame, makes it infeasible for black-box approach[2].

Red Teams are conducting attacks from simulated Internet that has large pool of IP addresses. Traffic generation and scoring systems use the same IP pool, so the attackers are not easily identified and whole range of IP addresses cannot be blocked [7].

2.1.3 After Action Review

Next subsection describes feedback sessions after execution. This is important for Blue Teams to learn from the experience and improve efforts next year.

After exercise, each organizing team makes their conclusion and gives feedback to Blue Teams. This reinforces Blue Team learning goals and gives information about exercise events. Also, important information, that happened behind the scenes, can be communicated to them.

There are three feedback sessions in Locked Shields, which serve slightly different goals.

- Short feedback session, right after execution
- Longer feedback and discussion, a day after execution
- After Action presentation, a month after execution

Shorter session is held immediately after execution to pull together exercise events. Each team, except Blue Team, has approximately ten minutes to make a conclusion of their activities [9]. Since it is a game, the winner is also declared. In short feedback, Web Team should make a conclusion of their success of completing objectives and used attack methods [8].

Longer feedback is given in a discussion and evaluation (hotwash) session where execution overview is made [8]. All teams analyze the game from their point of view and answer questions from Blue Teams. Here the prevalent mistakes are shown to improve next year execution and assert lessons learned for Blue Teams.

Third round of feedback is given in the presentation and discussion of After Action Report. This is done after a month, so that all teams have had time to analyze exercise

[2] methodology where attacker has no knowledge of the system beforehand

data [9]. When first two feedback sessions cover the immediate views of the team leader, After Action Report (AAR) discussion is an analysis of available data. Here, the outcome of the thesis helps to recall Web Team campaign and create a detailed analysis about it.

2.1.4 Test Run

Additionally, with large exercises like Locked Shields, a test execution can be used for testing infrastructure, communication and train new Red Team members. Similarly to final rehearsal in theaters, there is also an audience, in form of Blue Teams. For Locked Shields, the test execution, or test run, is a one day event with a single feedback session and limited number of objectives.

The main scenario stays the same overall. In any case, test run feedback is similar to short feedback sessions in real execution. However, there is no hotwash or After Action Report, so all important points and execution analysis must be done and communicated to Blue Teams right after execution [9].

2.2 Other Cyber Defense Exercises

Solution proposed by this thesis may be used in other CDXs. In technical exercises, where Red Team is used to provide an adversary for defensive teams, there should be some form of feedback from attackers[5]. This section describes some technical CDX in the world to show the possible relevancy of the thesis out of the context of Locked Shields.

Greece and Czech Republic have published information about their national CDXs. Greeks' exercise, called PANOPTES, is mostly an offline CDX with small scale real time attacks [12]. As Gritzalis et al. describe in [12] it has no scoring or evaluation from organizers and the solutions are provided after execution. In addition to self evaluation, there is some feedback to participants in form of full solutions. In case of web and client-side attacks, self evaluation in CDX is often difficult, because of covert techniques. When attacker compromises database and steals information, without leaving identifiable tracks in logs, then it is difficult for Blue Teams to detect an attack altogether.

Czech Republic CDX platform employs cloud technology to allow different use cases for example trainings, research and development, forensic analysis, network simulation and exercises [13]. Sample use case of training was done as a capture the flag game [14].

This means that there is no Red Team to attack systems, but competitor try to exploit vulnerabilities of systems for a proof or “flag”.

Cyber Europe has different approach to cyber exercise than Locked Shields [15]. It is focusing more on building international communication. Three levels of further escalating phases allow to improve technical, operational and strategic side of cyber incident mitigation. They use a Cyber Exercise Platform developed by ENISA, to plan, conduct and evaluate the exercise.

Another Cyber Defense eXercise that this thesis relates to, is called Cyber Range eXercise (CRX). It is conducted by Clarified Security OÜ as a Red Team/Blue Team type technical exercise. General information about the exercise comes from Clarified Security Chief Executive Officer (CEO), Mehis Hakkaja [9]. While the author of this thesis also participated in delivery of 2 CRX executions and preceding test run in 2015.

“Compared to Locked Shields large scale exercise series, CRX format differs from full simulation by not having media or legal injects and focuses only on technical game-play: Red Team attacks detection, reporting and defensive actions.” [9]. The Blue Teams, consisting of up to 12 members, are usually limited to just one or few teams. Therefore, having a more personal approach and more opportunities for feedback. With fewer Blue Teams it is possible to have feedback sessions between every phase or with the game being slowed down in areas where the Blue Teams seem to struggle.

Depending on the audience, parts of the exercise can be turned into an interactive technical training where Red Team directly communicates with the Blue Team. For example, to train Blue Team capabilities in detecting certain types of attacks like backdoor traffic and resulting anomalies, some hints or alerts are given directly. Involvement from Red Team can go against traditional view, however MITRE Cyber Exercises Playbook also recommends this, as one way for target audience to gain maximum value from the exercise [16]. While this kind of direct feedback during a smaller scale exercise is easier during the game-play, Red Team still has the need to have good chronological overview of its activities for the Blue Team after the exercise.

For competitions and trainings, there is a fully automated Cyber Defense Competition described by Ernits, Tammekänd and Maennel [17]. Their paper describes a solution for automating attack and scoring for competitions. One competition, held on that platform in 2015, was CyberOlympics 2015, where students had to defend their network from attacks [18].

3. Related Works

Subjects researched for the proof of concept were data collection and visualization. This chapter takes a look on some tools and methods used in central log management and data visualization techniques.

3.1 Log Management

Vaarandi and Niziński propose, among others, Syslog and ELK stack for log collection and visualization [19]. ELK stack is a combination of open source tools, Elasticsearch, Logstash and Kibana [20], which combines tools for fast search, processing and visualization. Elasticsearch is a document oriented distributed search store[1], which allows full text search and real-time data analysis. Logstash is an utility for event data processing and streaming[2]. Kibana is a event visualization tool[3] built on Elasticsearch database, that allows real-time data visualization.

Another similar tool to ELK stack is an enterprise software called Splunk [21]. It can analyze and visualize log data in real-time and also report and alert issues with infrastructure performance or security incidents. However, being enterprise software, it is dismissed for this thesis.

Graylog2 is a log management system using Elasticsearch as a storage [19]. As discussed by Vaarandi and Niziński, it can receive different message formats, parse them and visualize with built-in web service. There is also a possibility to configure real-time streams with alerts.

Syslog and its different flavors, described with more detail by Vaarandi and Niziński [19],

[1] https://www.elastic.co/products/elasticsearch

[2] https://www.elastic.co/products/logstash

[3] https://www.elastic.co/products/kibana

are event collection protocols used from 1980s. In 1980 Eric Allmann proposed the first version of Syslog Protocol, named BSD Syslog[19]. In general, Syslog is defined by IETF accordingly: “In its most simplistic terms, the syslog protocol provides a transport to allow a machine to send event notification messages across IP networks to event message collectors - also known as syslog servers.” [22]. The basis of this protocol proves to be useful to this day however the technical details have been improved with several implementations [19]. Although Syslog is good for transporting event logs to central location, this thesis needs to create and timestamp the events beforehand.

To look at real-time transport solutions one should turn to companies with large user bases like Facebook, Netflix, LinkedIn and Twitter. Real-time processing of large amount of events needs a high throughput data transport with low latency, to have minimal delay between event and processing. Netflix published its new Keystone pipeline in 2015 using Kafka, Elasticsearch, Hadoop and stream consumers in its architecture [23]. Similar architecture is used in LinkedIn data pipeline [24].

Central to real-time traffic stream in those applications is the Apache Kafka message broker. Apache Kafka is a scalable producer-consumer distributed messaging system developed in LinkedIn to transport 13 million messages per second [25]. Producer-consumer architecture allows to decouple components and have several consumers receive same message from the broker [26]. Also Kafka buffers messages, so in addition to real-time, consumers can be slow, batch processes.

3.2 Information Visualization

Visualizing information allows humans to interpret data more easily and detect patterns [27]. This is an important part of the thesis, to discover patterns in the data collected from exercise. Next section introduces related work in event visualization field.

Landstorfer et al. created a visualization for networks security on log records [28]. The article emphasizes on co-creative process between security engineers and visualization experts. Main ideas relating to this thesis are pixel map for creating patterns in log data and displaying raw data together with visualization.

Similar to Landstofer et al., Hao, Healey and Hutchinson describe a visualization system designed for security analysts [29]. They propose a web-based environment to visualize network alerts. To aide analyst, they aim for correlation of multiple data sources and customizable level of detail. In their paper Hao et al. propose several requirements for successful visualization tool, which mainly focus on existing improving analysts existing patterns. List from [29] by Hao et al.:

- Mental models;
- Working environment;
- Configurability;
- Accessibility;
- Scalability;
- Integration.

Livnat et al. propose a new visualization paradigm to support decision making and enhance situational awareness [30]. In the paper Livnat et al. describe a visual correlation and displaying framework. However situational awareness corresponds data to provide understanding of what is happening at the moment. The proposed model provides better perception and correlation of events, but this is more useful in real-time situations.

Shanks provides information on enhancing intrusion analysis by data visualization techniques [31]. In [31] Shanks describes different use cases to improve binary file, port scan and firewall log analysis through visualization. Also different tools like DAVIX and previously mentioned ELK stack with Haka and Hakabana additions to visualize data are discussed [31]. While, real-time data analysis is not in the scope of the thesis, it considered as a future work.

4. Specifications for Web Team Campaign Overview

This chapter focuses on analyzing current feedback and information necessary to create it. Learning objectives are also examined to find requisites for feedback. Detailed information about Web Team campaign is a base, for Web Team leader, to create the feedback. Therefore analyzing current feedback, together with learning objectives, help set specifications for campaign information.

4.1 Learning Objectives

This section describes Blue Team learning objectives, as described in Locked Shields scenario, related with Web Team campaign. There are over 10 training objectives defined specifically for technical specialists in Locked Shields 2016 [7]. Training objectives characterizes the Blue Team's technical goals for this exercise and implies, what questions feedback must answer, to fill the gaps. Feedback should reinforce those goals for Blue Teams [10].

Below are the specified technical learning objectives for Blue Teams, from Locked Shields 2013 After Action Report [10], that are most important to this thesis:

- Learning the network;
  - Blue Teams will be responsible for securing and maintaining systems unknown to them. They need to compile lists of assets and vulnerabilities, assign priorities to the assets, etc.;
  - There will be network segment(s) completely unknown to the Blue Teams before the game starts to put more focus on coping with unknown aspects.

- System administration and prevention of attacks;
  - Administrative tasks and hardening the configuration will be continuous activities.
- Monitoring networks, detecting and responding to attacks;
  - Capability to detect “quiet” activities of the Red Team will be evaluated higher.

4.2 Current Feedback in Locked Shields

This section looks at current feedback from Web Team given in Locked Shields from interview with Elar Lang A.1. In Locked Shields 2012 After Action Report [32], it is recommended that Blue Teams should be provided with more detailed feedback. This is similarly recognized as improvement point, to tell the “offensive story” in Locked Shields 2013 After Action Report [10]. Also, ENISA report on cyber exercises from 2015 emphasizes the need to have good methods for critical reflection [5].

Main goal of CDX is to provide learning experience for Blue Teams and feedback has a big role in providing that. In CDX, feedback sessions assess Blue Team actions in the exercise and explains Red Team campaign. As mentioned in chapter 2 there are multiple ways to conduct the sessions. Locked Shields has a short session right after execution, a longer one day after execution for immediate discussion and After Action Report meeting a month after the exercise.

One quantifiable property is objective list and the score from completed objectives. In Lang’s experience, Blue Team’s score does not help to determine if defense was implemented correctly, only which teams are doing better than others. Better score can also be achieved by “playing the rules”, which means focusing all effort to understand how score is computed [8]. For example, enabling some services only for scoring agents or disabling potentially dangerous functionality on websites. In addition, several Blue Team assets are attacked simultaneously and it is impossible to determine what exactly happened from scoring alone.

For short feedback session, to give some insight about attacks, each Red sub-team leader makes a summary of their team’s objectives [8]. The feedback is given for twenty Blue Teams at once. It is general and only pointing out the most important – what objectives failed and what succeeded. As there are several ways for the attacks to fail, team leaders cannot make confident conclusions about Blue Team defenses. Attacks can fail due to availability issues or removed functionality.

Currently Web Team members should take notes of services that are not available or not functional [8]. Correlating these notes to objectives that failed is manual work for Web Team leader. This information, compiled by Web Team leader, is a base for feedback to Blue Teams.

Red Team is providing a service in CDX context, attacking the Blue Team's infrastructure [8]. An equal treatment for all Blue Teams is the highest priority for them. This means that, when Red Team starts to take detailed information about their actions against one Blue Team, it delays actions towards another. Such conduct is therefore undesirable from Red Team, which means there is not enough time for note taking.

With general information about the campaign, it is harder to recall the exercise events in a detailed manner. After Action Report meeting takes place a month after the exercise. Therefore, it is important to have detailed information about Web Team campaign to help the team leader remember what happened.

4.3 Campaign Overview Specifications

For previous Locked Shields exercises, main information about Web Team campaign has been manually collected and first automated functionality tests for web services were carried out in Locked Shields 2015 [8]. This section proposes solutions how to increase level of detail in Web Team campaign overview for Web team leader. Solutions involve collecting data from exercise environment and visualizing it to give detailed information about Web Team actions.

4.3.1 Web Team Objectives

Campaign is organized so, that Web Team has to complete objectives, by exploiting Blue Team systems [8]. Each objective must be completed at the end of the phase or marked as failed. With successful result, it can be certainly said that Blue Teams failed to defend their services. The opposite is not always true.

Objective can be marked as failed, when Blue Teams have put up sufficient defenses, their website is not available or the required functionality is not working for exploit. For example, when exploit needs the web service's file upload functionality and Blue Team has disabled it. This means, that the objective fails, but disabling a service is not a viable defense [8]. However, when Blue Team checks the file type and content of an uploaded file and blocks malicious files, then objective failed due to sufficient defenses.

It is allowed and suggested for Red Team member to do extra attacks or expand the foothold on Blue Team systems [8]. Objective list is for supporting game scenario and other team's tasks, for example defacements support media injects. However, activities beyond those of the objectives can reflect on score different ways. When Red Team disables web services or defaces website, Blue Teams can lose availability points.

Therefore, looking only objective completion data does not give the full overview of Red Team campaign. Information from objective data should be enhanced with availability and functionality data. To have detailed information about completed objectives, Web Team leader must have background information about the web service and Web Team actions.

4.3.2 Availability and Functionality

To have background information about Blue Teams' web services, it must be noted when they were functional and available. In Locked Shields, Green Team is testing each service for scoring purposes and that has an additional meaning for Web Team campaign [8].

Availability tests check, if the service and the underlying infrastructure (server machine and network) is accessible from simulated Internet. These tests fail when Blue Teams are too restrictive with firewall rules or have misconfigured services.

Some Blue Teams remove or restrict the functionality of web services. This is not in accordance with game rules and scenario, as the Blue Teams objective is to keep services available and functional [8]. However, in those cases the availability checks pass, but Red Team could be unable to complete objectives. Starting from Locked Shields 2016, functionality test are carried out by Green Team to keep Blue Teams to the rules [8]. These tests keep Blue Teams from playing the game only for score, by disabling functionality on websites. Using the data from these checks together with objective data can improve the overview on Red Team campaign result.

4.3.3 Web Team HTTP Requests

One possible solution to give an overview of Web team actions is to record all HTTP requests towards Blue Teams. Collecting every request sent from Red Team to Blue Team systems, gives the possibility to create a timeline of most of important events. Web Team mainly uses HTTP requests to exploit web services, therefore recording only this kind of traffic is done for the proof of concept.

Web Team's process to complete one objective is described in figure 4.1. The process shows an ideal defacing attack situation, with working services and no defense. During the exercise, the process changes when Web Team member faces active defense or web services that are not working. However, the overhead of note taking remains in every variant of the process.

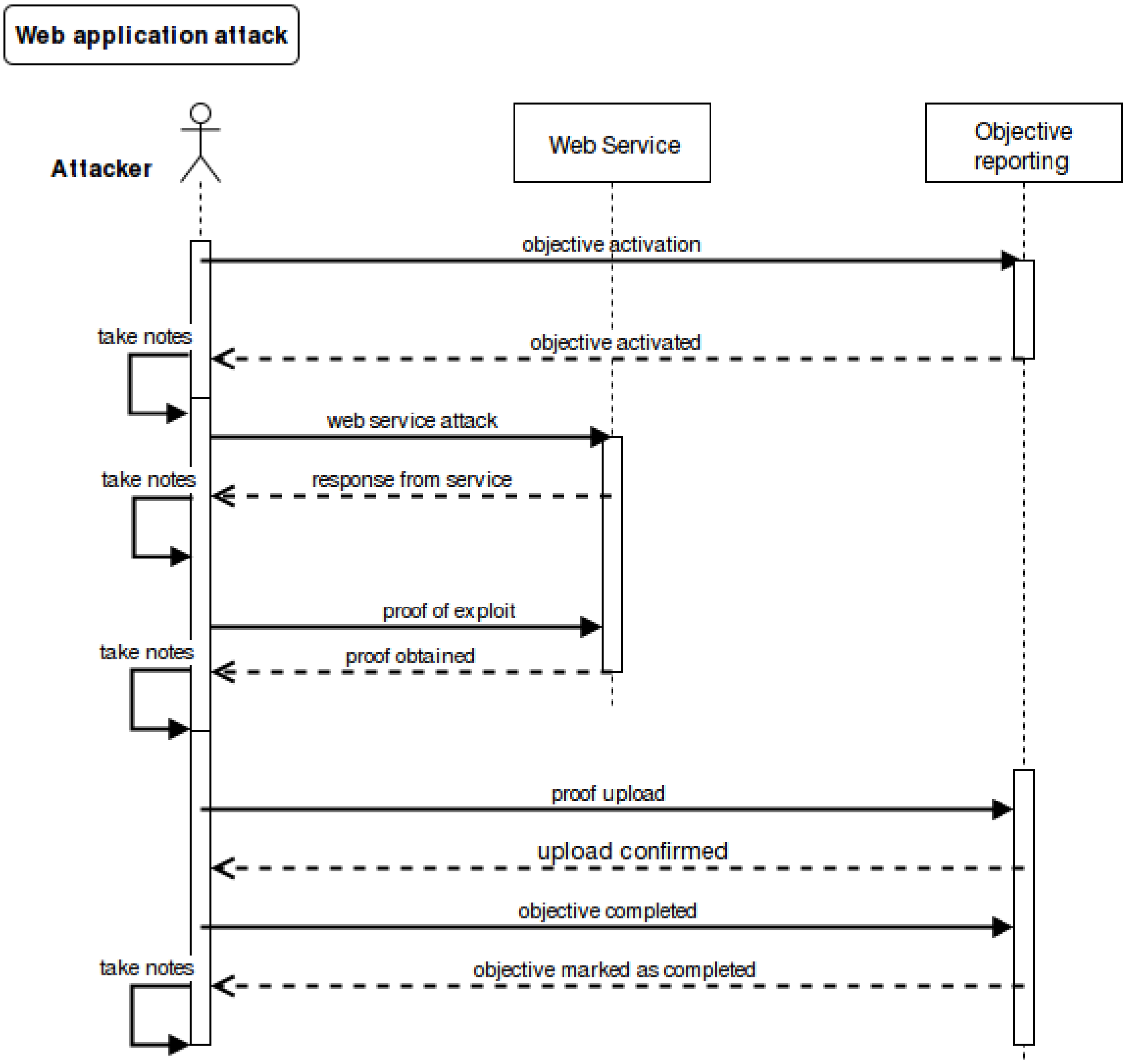


Figure 4.1: Web application objective completion process.

As calculated in problem statement section 1.1, there is a significant overhead by reporting. Although there are several Web Team members and the time distributes between them, it still considerable. This calculation does not take into account the time Web Team members take notes about their attack.

When faced with volatile environment in live exercise and reporting overhead, Web Team members have little time to take detailed notes. Level of detail of Locked Shields 2016 Web Team notes can be seen in the example added in appendix A.3. For example, about service 6 and 7 it is very general. Recording Web Team actions towards Blue Teams' web services allows to reduce the load of note taking in the process and improve level of detail.

Web Team uses different set of tools and scripts to complete their tasks. The solution should be made, to collect requests towards web services from browsers and command line tools, like curl and custom scripts.

Collected requests allow Web Team leader to gain overview of Web Team actions, to make after action analysis. HTTP requests also help to gain insight about possibilities why objectives failed or how they succeeded. For example, it can be seen that attacker failed to exploit a web service over IPv4, but when trying with IPv6 it was successful. This shows that the team was not giving as much attention to IPv6 traffic.

4.3.4 Campaign Details

Details of Web Team campaign should consist of important events covered in this section. Events considered important are web service requests, objective status changes, system availability changes and functional testing results.

To evaluate the proposed important data, a proof of concept is developed with specific architecture details covered in chapter 5. It must be taken into consideration that introduced tools and processes should have minimal interference with current Web Team processes. All design decision should avoid adding assignments to Web Team members and when necessary they must be documented and integrated in Web Team members training process.

The visualization of collected and chronological data must be created as a timeline of sequential events. To best analyze the data, different events and statuses should be grouped together by color to provide better pattern recognition [33]. Color coded timeline provides

Web Team leader a base to correlate and analyze events for After Action Report.

Next list is a concludes the necessary data from exercise environment:

- **Red Team objective information**;
  - Derived from objective status change events - active, success, failed;
- **Web Team HTTP requests**;
  - HTTP requests from Red Team member to Blue Team systems;
- **Blue Team systems availability information**;
  - Derived from Green Team availability check event data;
- **Blue Team web services functional testing information**;
  - Derived from Green Team web services functional test event data;
- **Comments from Web Team members**;
  - This part is optional and must be implemented with caution not to give more tasks to Web Team members.

4.3.5 Specifications for Visualization

This thesis aims to provide the tools to give detailed information about Web Team campaign. However, decision, of what information is given to Blue Teams, comes form team leaders. This section describes specifications for visualizing information for Locked Shields feedback.

There are two distinct feedback forms – report presentation and written report [8]. First, the presentation needs to give an overview of the Web Team campaign. It also should give a comparison of Blue Teams and their actions defending different systems. Derived from these points, the following information must be presented to Web Team leader:

- Web Team campaign summary about one web service for all Blue Teams;
- Web Team campaign summary for one Blue Team over all web services;
- Average objective statuses over all campaign objectives;

- Statistics of different objective statuses for one Blue Team over all web services;
- Statistics of different objective statuses for one web service for all Blue Teams.

Second is an individual report for each Blue Team. This is a written report for Blue Teams, to have an overview of Web Team campaign on their web services. For this, summary visualizations must be made to limit the information detail but give Blue Teams useful information at the same time. Derived from these points, the information in the report must include:

- Web Team campaign summary for individual Blue Teams;
- Statistics of different objective statuses for all web services separately for individual Blue Teams;
- Average objective statuses over all campaign objectives.

Completed objectives have three statuses, when considering availability and functionality data:

- Failed objectives;
- Successful objectives;
- Objectives failed due to not available or not functional web services.

Campaign summary is a graphical representation of objective statuses, per phase, for each web service. There, one phase should be summarized for one web service and in addition to objective status, the availability and functionality, must be taken into account. The summary shows, which Blue Teams and which services Web Team exploited most successfully.

Statistical views of objective statuses for one Blue Team give a general view, how well Web Team did against them. Calculating average objective statuses over all Blue Teams aides in comparing Blue Teams. Comparing individual statistics with overall average shows how well Blue Teams did in comparison to other Blue Teams.

5. Implementing Data Collection and Creating Overview

In this chapter, technical considerations are described, for data collection and Web Team campaign overview creation. Design for a framework is constructed, to implement improvements proposed in chapter 4. It outlines the overview contents regarding information from multiple sources and correlation of different events.

Developer of Red Teaming tools like Cobalt Strike and Armitage, Raphael Mudge, has written and article [34] on the importance of feedback. There he says: “If a team can’t find a red event in their logs, then they have a blind spot and they need to put in place a solution to close this gap.” [34]. Feedback should show the Blue Teams what they are not able to see in their logs. When using frameworks, like Cobalt Strike, it is possible to log all events from single place and easily generate the reports needed [35].

However, web attacks usually employ different scripts and manual work to complete objectives. For that reason, tools and methods to collect data and process them are proposed in this chapter. It is also important to mention that, although, these events are important to Blue Team, the framework is only for Red Team use. Detailed event information created by this framework is the basis of feedback to Blue Teams.

5.1 Architecture

This section describes the high-level architecture of the proposed framework, which is generalized for any technical CDX. Meaning that the architecture proposed is not specific for Locked Shields, but ideal representation of the framework. It introduces the abstract components for data collection, from different sources, and processing.

Main points of the proposed architecture are derived from works related to central log management and real-time event messaging systems viewed in chapter 3. Real-time campaign overview is considered in the architecture for future work, but is not in scope of this thesis. For now, real-time data is only used for assuring that data collection works in live exercise and for displaying error messages to author.

Proposed architecture components and data flow is shown on figure 5.1. All data is collected to central management machine and processed there. Before processing, all raw data is saved to datastore for backup.

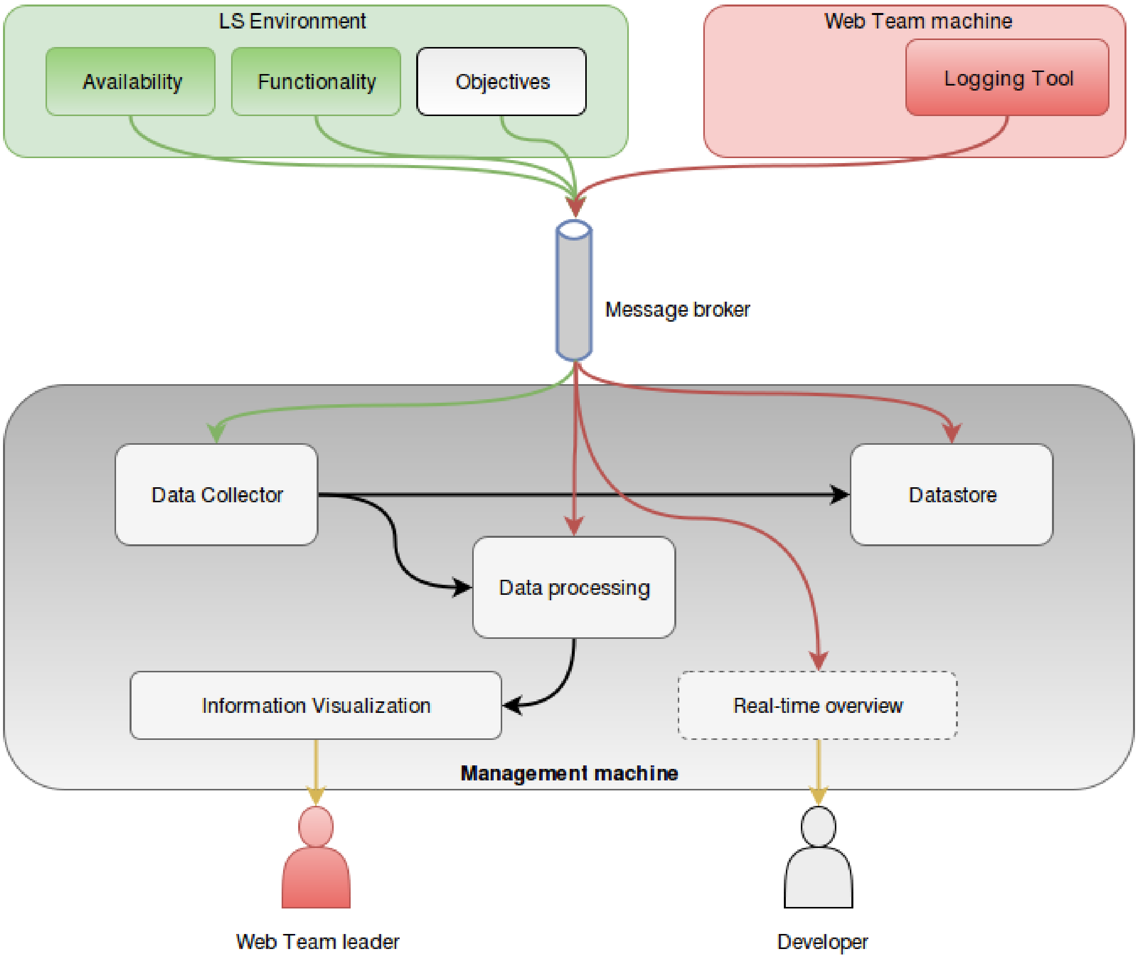


Figure 5.1: System architecture.

The pipeline carries two kinds of data. First the red line reflects data from controlled source, already formatted. In this case it denotes the HTTP requests from Web Team.

Second form of data comes from exercise environment and is denoted by green line. Proposed method for collecting data is writing those events to message broker as they occur. From broker, the collector consumes the messages and formats to required JavaScript Object Notation (JSON) format. After formating the data it is passed to processing and database storage.

Architecture is simplified, showing only one Web Team machine. In reality, architecture supports multiple machines, sending HTTP requests to central management machine.

Black lines denote the internal messages that are already formatted. This data flow carries event data between different parts of the framework. Creating a unified structure for internal data allows adding or changing parts of the framework.

Data processing deals with aggregating collected data for visualization. This includes creating correlation between events and services. Before processing the data was considered as chronological events and not relational. This step creates a relation between all events by services and Blue Teams.

These structures are created internally and forwarded to visualization. Information visualization displays processed data for Web Team leader and is denoted with yellow line.

5.2 System Design

This section describes chosen components based on system architecture in context of Locked Shields 2016. Main restrictions for the framework was to minimize interference with Web Team processes and development from Green Team.

Proof of concept design is built on top of open-source tools, using Python programming language to integrate different parts, where needed. Python was chosen by author because of previous experience in that language, which minimizes the effort needed to develop the proof of concept.

Final design of data flow between components is shown on figure 5.2 and differs from proposed architecture due to restrictions mentioned.

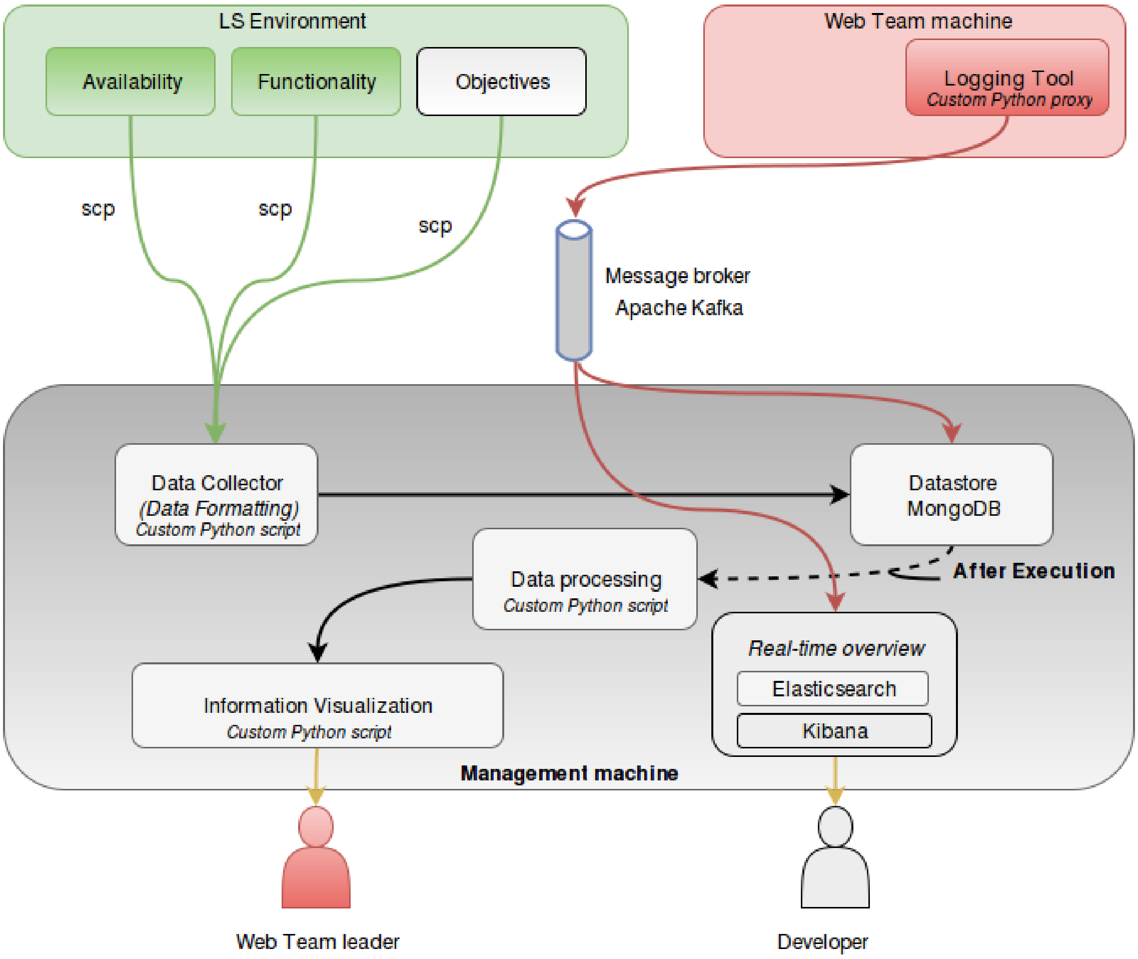


Figure 5.2: System data flow for Locked Shields 2016.

5.2.1 Internal Message Structure

All red and black lines on figure 5.2 denote data flow in JSON message format. It was chosen for its human readable form, which in initial development and testing improved the speed of debugging the application. Also, JSON can be directly inserted to elasticsearch index[1] and is included in Python standard library with good documentation[2].

[1] https://www.elastic.co/guide/en/elasticsearch/reference/current/docs-index_.html

[2] https://docs.python.org/2/library/json.html

5.2.2 Message Broker

Message broker acts as a buffer between data publisher and subscriber, while publisher is independent of subscriber [26]. This is useful, when there are multiple consumers for one data stream, as seen on design figure 5.2. Both, real-time overview and datastore consume same message from Web Team logging tool.

Kafka message broker was chosen, to the part of message broker, to buffer messages for central server consumer. Main reason for broker choice was that Kafka is scalable, does not need complex setup for initial development and it has well documented Python library [24].

Since author did not have estimations on how much traffic Web Team could generate in Locked Shields 2016, while attacking Blue Teams, a scalable solution for brokering messages was selected. This solution is able to buffer messages for reader, so processing delays should not result in message loss. Kafka has proved itself in LinkedIn to be able to scale and handle over 2.75 gigabytes of data in a second [25]. Also, it is important that batch messaging and compression is built in for Kafka to save network bandwidth capacity [24].

5.2.3 HTTP Request Collection

To collect data about Web Team attacks, author built a logging tool installed in Web Team members workstation. This collects all HTTP/HTTPS traffic towards Blue Team networks and forwards it to central server. Logging tool uses a HTTP/HTTPS proxy as a request collection method, which needed no development from Green Team side. Specific development decisions for the tool are described in software development section (5.3).

For real-time overview of request collection tool, Elasticsearch is used as a search and indexing service. On top of that is a Kibana instance, that allows on the fly customizable graphical views of data provided by Elasticsearch. Collecting Web Team requests and errors there, during the exercise, is used to make sure the tool is working properly.

5.2.4 Environment Data Collection

Environment data in this context is functionality and availability tests from Green Team and objective statuses from White Team. Since minimum development effort from Green Team was expected to collect data, the proposed method was not implemented. The design was simplified for Locked Shields 2016, as shown on figure 5.2.

Data was transfered over Secure Shell (SSH), using SCP[3], by Green Team. As the data was given after the execution, it eliminated the chance of real-time data visualization. However this was not in the scope of the thesis, therefore it is an acceptable solution for proof of concept.

5.2.5 Data Storage

Data was also saved to a database as a backup solution and to be manipulated afterwards. In place of database MongoDB[4] was used, to store all events. The main objective during the execution of Locked Shields 2016 was to collect the data needed for processing and visualization. Therefore data was stored during the exercise.

5.2.6 Processing and Visualization

Researched options did not provide all features needed for Web Team leader to get full overview of Web Team campaign. Therefore custom Python scripts, using plotly[5] library, were constructed to create necessary visualizations. Examples from Landstorfer et al. [28], Hao et al. [29] and Livnat et al. [30] were taken for building the visualization. These articles provided the insight of grouping and displaying events, as well as raw data, to bring out general patterns and provide enough detail.

5.3 Software Development

This section describes authors development process for the proof of concept framework that provides detailed information about Web Team campaign. Here, more detailed design

[3] `http://linux.die.net/man/1/scp`

[4] https://www.mongodb.org/

[5] `https://github.com/plotly/plotly.py`

decisions for Web Team request logging tool and data processing tools are characterized.

Incremental model was used to develop all parts of the framework because next component was dependent on previous [36]. The implementation for complete framework was done in three increments (figure 5.3).

First increment was to develop data collection tool for Web Team requests and exercise environment data. This part was essential for Locked Shields 2016 because other parts of the framework relayed on the data collected with that. Before Locked Shields test run there was no possibility to test the implementation on real exercise environment. Therefore with the data collection tool, iterative model [36] was used together with continuous deployment methods. This provided the possibility to quickly provide fixes and new features in live environment [37].

Second and third increments added data processing and visualization. Those were done after the Locked Shields main execution and incorporated real data collected from the exercise. Because author had no knowledge of data formats provided by Green Team, the processing decisions and specifications were done after the exercise.

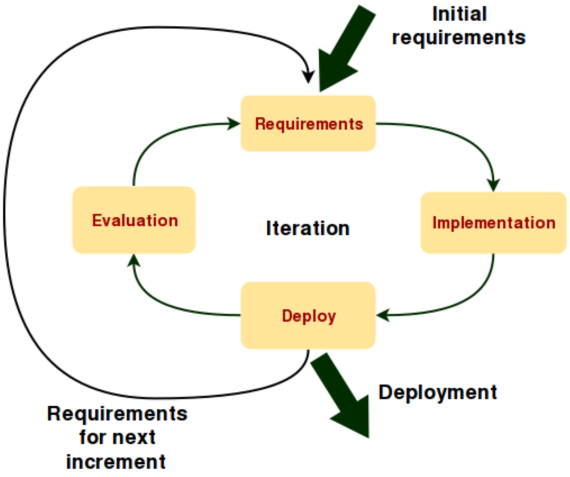


Figure 5.3: Developent model.

5.3.1 Web Team Request Logging Tool

First task was to develop preliminary data collection tool for Web Team requests to Blue Team web services. Author tested his ideas on local development machines using VirtualBox[6] and Vagrant[7]. Preliminary development was done on Debian[8] Linux operating system which is a base for Kali Linux. Kali Linux[9] is a primary operating system for all Red Team members in Locked Shields 2016.

To improve development process in an unknown environment, continuous deployment model was used, as shown on figure 5.4 [37].

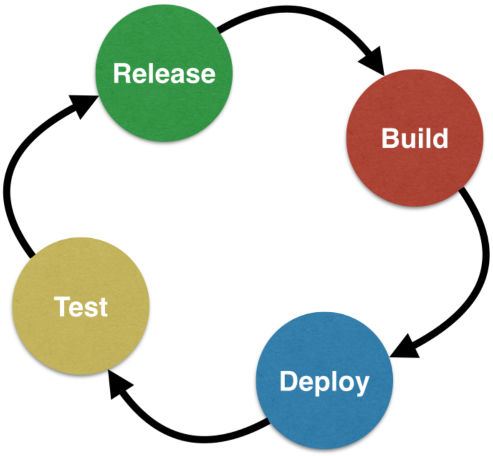


Figure 5.4: Continuous deployment model.

First increment was done for Locked Shields test-run, where it was tested with actual exercise network. For first increment following requirements were specified:

- **Requirement 1** Logging tool is able to intercept HTTP traffic from machine it is installed;
- **Requirement 2** Logging tool is able to intercept HTTPS traffic from machine it is installed;
- **Requirement 3** Intercepted traffic is filtered to only process traffic towards Blue Team network;
- **Requirement 4** Filtered traffic is formated to JSON format described in appendix 5.1;
- **Requirement 5** Logging tool saves all formatted traffic on a file on machine it is running;

[6] `https://www.virtualbox.org/wiki/VirtualBox`
[7] `https://www.vagrantup.com/about.html`
[8] `https://www.debian.org/intro/about`
[9] `https://www.kali.org/`

- **Requirement 6** All formated, intercepted traffic is sent to message broker;
- **Requirement 7** All system error messages are sent to message broker;
- **Requirement 8** Logging tool checks if update is available from management server;
- **Requirement 9** Logging tool updates itself from updated package obtained from management server.

Logging tool with simple HTTP traffic intercepting proxy as central component was devised for Requirement 1. It also had SSL/TLS[10] capabilities for requirement 2 to log HTTPS communications that are used for the attacks. Proxy was one solution to record HTTP and HTTPS traffic from Web Team machine without extensive knowledge about exercise architecture. It is important to note that the logging tool was installed on each Web Team members' machine. When done centrally, Blue Teams would have identified the single IP address for all attacks.

Logging tool filters and sends data to broker only if the target address is in Blue Team's network. Filtering is done by checking if the HTTP(S) *Host:* header matches Blue Team Network addresses for web services, set in configuration file. This avoids unimportant data sent over network, like Google searches.

The data is also formated before sending, considering to Raphael Mudge's timeline specifications for his tools [34]. JSON format for request logging logging is shown in code example 5.1 This format contains all necessary details about request made by Web Team towards Blue Team systems and is used for collecting data.

```
{
        "timestamp":"UTC",
        "source": {
            "mgmpt_ip": "IP",
            "attack_ip": "IP list",
        },
        "target":{
            "host": "target DNS name",
            "target_ip": "target IP address",
            "blue_team": "team number",
            "ssl":"ssl/tls boolean"
        },
```

[10] Cryptographic protocols to encrypt traffic over computer network

```
        "event":{
            "request":{
                "resource": "path in URI",
                "method": "HTTP method",
                "req_body": "HTTP request body",
                "req_head": "HTTp request headers",
            },
            "objective":{
                "nr": "objective number",
                "count": "objective completion count",
            },
            "response":{
                "headers": "HTTP response headers",
            }
        }
    }
```

Code example 5.1: Web Team request event log format.

It is possible that Red Team member attacks other Blue Teams from one compromised Blue Team systems. In that scenario the traffic is not coming from attackers machine, but from the compromised system. This is an acceptable problem and for now, in such specific cases, Red Team members should take additional notes about the actions.

One problem with the proposed solution is that it must be configured on the machine and in case of failure it block any traffic from the machine. With Web Team using pre-configured Kali Linux machines in Locked Shields, configuring is simply setting environment variables for HTTP and HTTP Secure (HTTPS) traffic. In case of failure, a command is added to user environment to easily toggle proxy settings from command line.

Web Team uses different tools and scripting languages and a proxy is one of the solutions to unify event logging with minimal interference to existing tools. Also it is possible to identify Red Team members through management interface in their system that never changes. The Kali machines have two network interfaces configured, one for management and other for attacking Blue Teams. Using a tool inside Web Team Kali machine it is possible to identify both interface addresses.

For Locked Shields main execution automatic software update was implemented. The main data flow of automatic update is shown in figure 5.5 as a part of design scheme (figure 5.2). It used a process in continuous loop which checked if an update is ready using Kafka. Polling avoids creating a listening port on Web Team members machine and avoids creating a requirement for local firewall configuration.

Update controller was a python script that sends update message through message broker to logging tool in Web Team machine. Update process in Web Team machine then asked update controller for the update package. URL to ask the update package is set in logging tool configuration file.

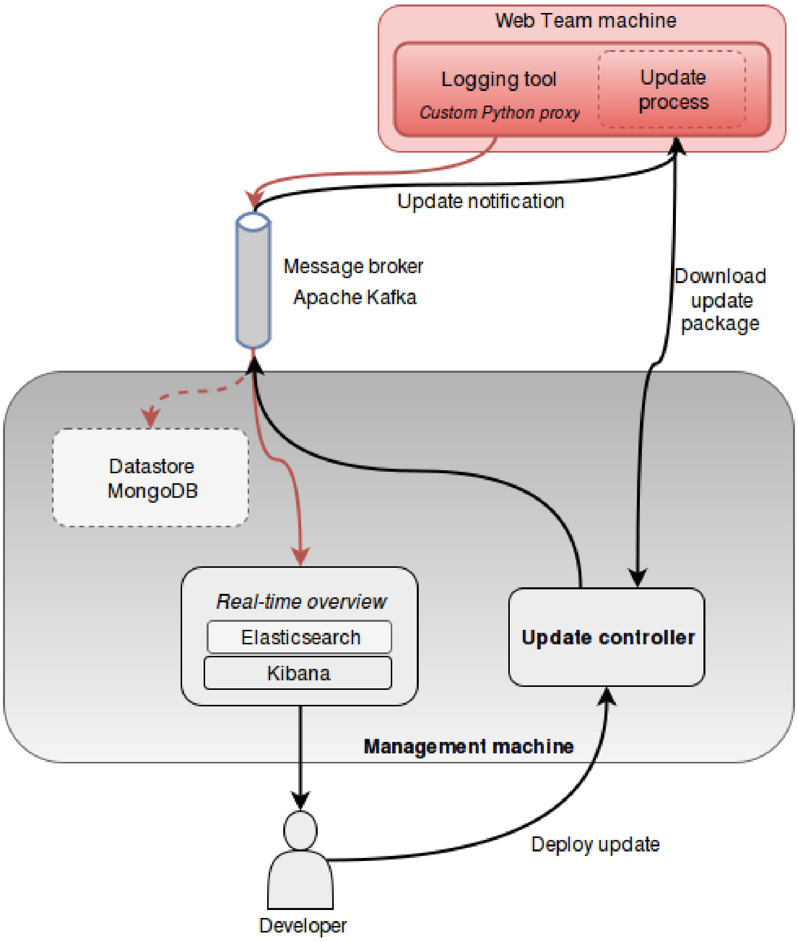


Figure 5.5: Automatic update data flow.

With fourteen Web Team machines, an install script was also needed, because the software had several dependencies. In addition to logging tool, this script also installed Firefox profile that had proxy CA certificate and FoxyProxy[11] add-on installed for convenience. The tool was pre-installed on all Web Team machines for Locked Shields main execution.

[11] https://addons.mozilla.org/en-US/firefox/addon/foxyproxy-standard/

For setup, Portable Document Format (PDF) document, shown in appendix A.4, was created to present in Locked Shields 2016 main execution. To familiarize Web Team members with logging tool, the setup was presented during training day of main execution.

5.3.2 Collecting Exercise Environment Data

In this section, data collection possibilities and data formats are further described. During the initial design of the framework author had no information about data formats in Locked Shields environment. Therefore, data processing was done after all data was collected.

In Locked Shields 2016 scoring is done from availability checks, objective completions and functional testing [8]. Availability and functional testing support overview by filling the gap of Red Team activities.

Functional testing is important for Web Team, because several web objectives expect fully working targets to exploit and complete objectives. Also, to discourage Blue Teams from "score play" - focusing only on score in the game. Objectives marked failed due to missing functionality do not reflect correctly the situation of game. Functional testing information is important for Web Team leader to see reason behind failed objectives and make conclusions. When the attack failed it cannot be concluded was it on account of good defense measures or unavailable services.

Following events are collected from CDX environment to improve overview of Web Team campaign:

- Objectives event;
  - Includes service identification, start time, completion time, status.
- Availability check event;
  - Includes service identification, test timestamp and service status.
- Functionality test event.
  - Includes service identification, test timestamp and service functionality status.

The CDX environment data format is described for the framework and it formats the data according to internal structure. To unify all separate events in different format a JSON structure is proposed and described in code example 5.2

```
{
    "timestamp":"UTC timestamp",
    "team":"team nr",
    "target":"host|ip",
    "type":"attack|availability|functionality|objective",
    "data": {
        //extra data about event
    }
}
```

Code example 5.2: Unified event format

All listed events in CDX have a timestamp, target team and target system or service (team information can be combined with system or service data). Further, events are classified as attacks, availability checks, functionality checks, objective data or other events. This list can be expanded when other event types are added and found important for the report.

5.4 Data Processing and Overview Generation

This chapter describes data processing using collected data from Locked Shields 2016 main execution. Proof of concept processes data offline, but it is possible to apply these methods to real-time data processing, when optimized. One of the outcome of this thesis is an overview of Web Team campaign. This overview is generated in two formats. Objective based summary of each phase and timeline, based on all events, visualizing each minute of the exercise.

5.4.1 Creating Phase Summary

First overview is a summary of Web Team objective completion per phase. This summary is created with multiple views to gain insight from different perspectives. Main views are chronological summary of events by phase and statistical summary of objective statuses. Basis of the summary are objective status information, availability and functionality testing events per service. Code example **??** shows how the data is consolidated for further visualization.

When objective failed, the availability and functionality tests data is evaluated to improve the feedback on why attackers did not succeed. In each phase attackers have finite amount of time to pursue the objectives. When deciding, if the service was unavailable or not functional, it is reasonable to set a threshold in a time window. To summarize the availability times in Locked Shields 2016, 10% was set as an example threshold [8]. Threshold means that web services must have been available and functional more than the specified time to be considered "ok" in the summary. It does not relate to real exercise availability and functionality scoring in any way. Availability and functionality statuses are calculated from all testing events per phase.

To generate the phase summary view, processed data is analyzed according to figure 5.6. If there were multiple objectives for one web service, they are analyzed all together per phase.

When all objectives succeeded the phase status is set as "objectives completed". Availability and functionality information is considered when at least one objective failed. Finally, if the site was available, functional and all objectives failed phase status is set as "objectives failed". This implies that the service had good defense from Blue Team in that phase, but that implication is not definitive.

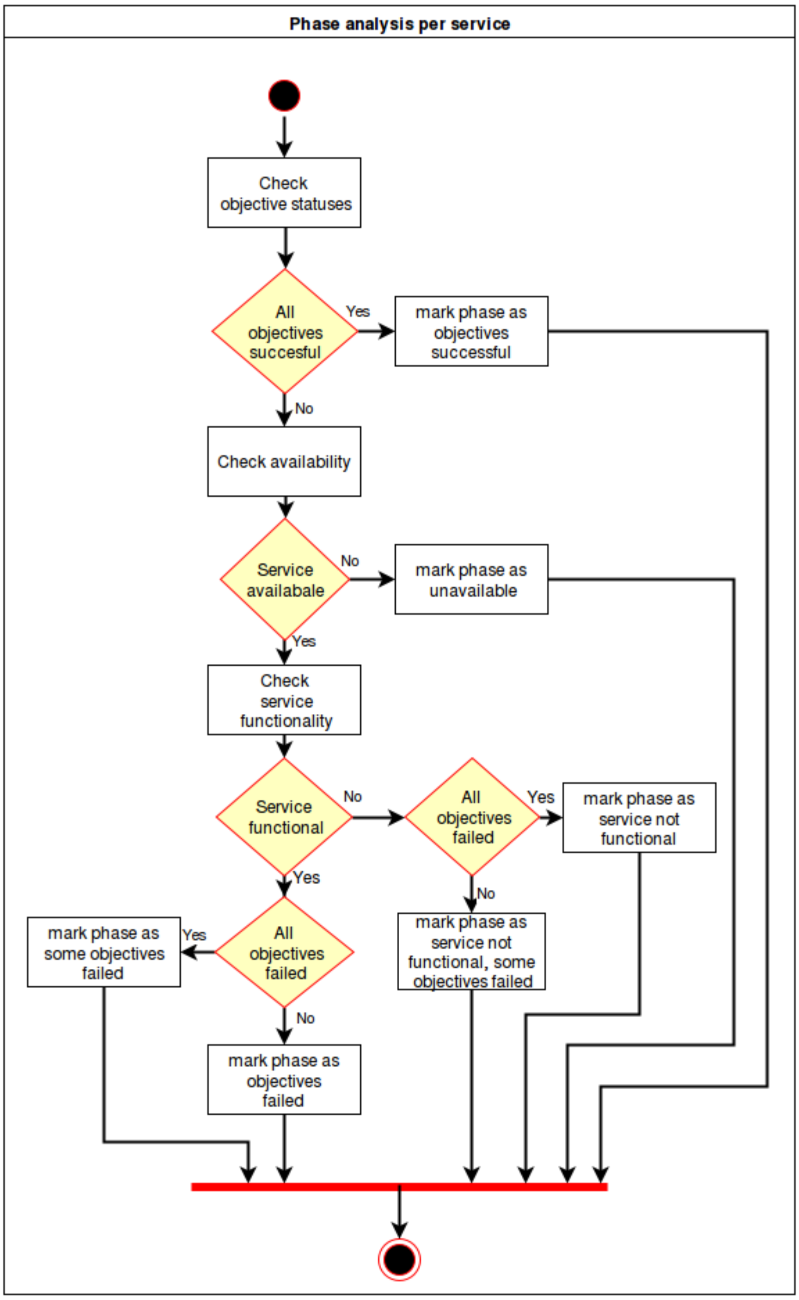
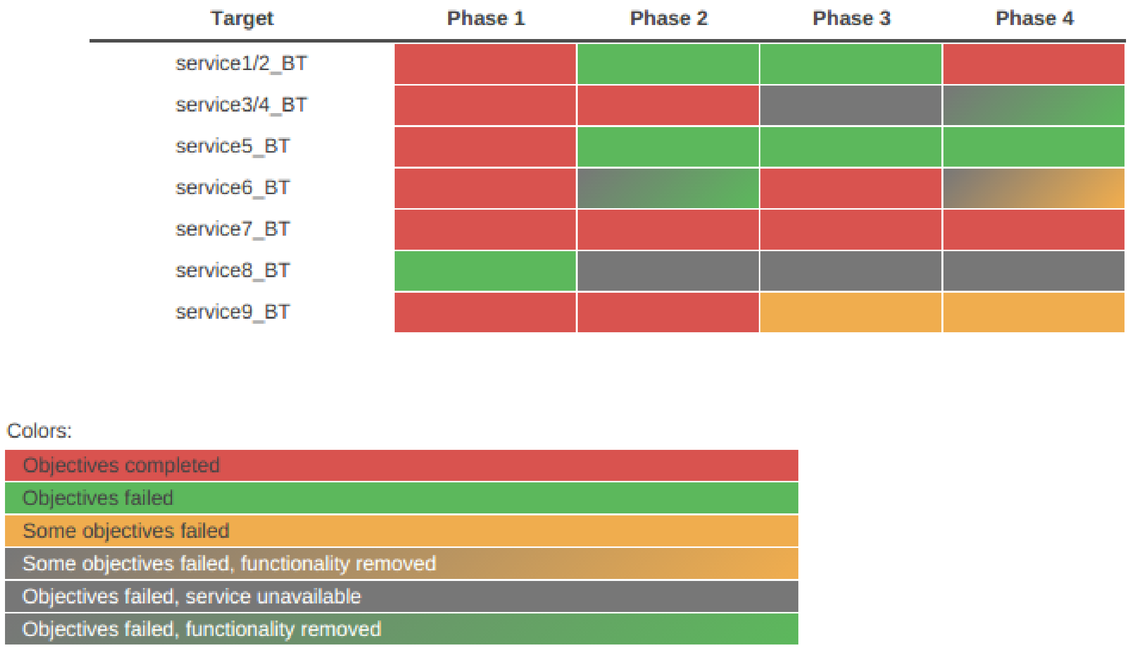

Phase analysis per service
Check objective statuses
All objectives succesful
Yes
mark phase as objectives successful
No
Check availability
Service availabale
No
mark phase as unavailable
Yes
Check service functionality
Service functional
No
All objectives failed
Yes
mark phase as service not functional
No
mark phase as service not functional, some objectives failed
Yes
All objectives failed
Yes
mark phase as some objectives failed
No
mark phase as objectives failed

Figure 5.6: Phase objectives analysis.

All figures in this thesis do not correspond to specific Blue Teams from Locked Shields 2016. Also, colors chosen to represent objective statuses are from Blue Team point of view. Therefore objective status "failure" is colored green and "success" red.

Figure 5.7 shows a sample overview of all web services for one Blue Team as specified in section 4.3.5. Phase is colored red when objectives were completed by Red Team member. Other colors show some sort of failure in objective completion. For example, when target has multiple objectives in same phase and some of them were unsuccessful, then it is shown in yellow.

It is also important to distinguish when objectives failed for working services, or when web service was unavailable, or functionality was hindered. Yellow with black and green with black emphasize that the application was partly unavailable and the objectives could have failed due to that. All black phase shows that the service was not available most of the time during that phase. Pure green shows that objective failed and the web service was functional according to functionality and availability tests.



Target	Phase 1	Phase 2	Phase 3	Phase 4
service1/2_BT	🟥	🟩	🟩	🟥
service3/4_BT	🟥	🟥	⬛	⬛🟩
service5_BT	🟥	🟩	🟩	🟩
service6_BT	🟥	⬛🟩	🟥	⬛🟧
service7_BT	🟥	🟥	🟥	🟥
service8_BT	🟩	⬛	⬛	⬛
service9_BT	🟥	🟥	🟧	🟧

Colors:

- 🟥 Objectives completed
- 🟩 Objectives failed
- 🟧 Some objectives failed
- ⬛🟧 Some objectives failed, functionality removed
- ⬛ Objectives failed, service unavailable
- ⬛🟩 Objectives failed, functionality removed

Figure 5.7: Blue Team exercise summary sample.

Figure 5.8 shows a sample phase summary to have an overview of one web service over all Blue Teams.

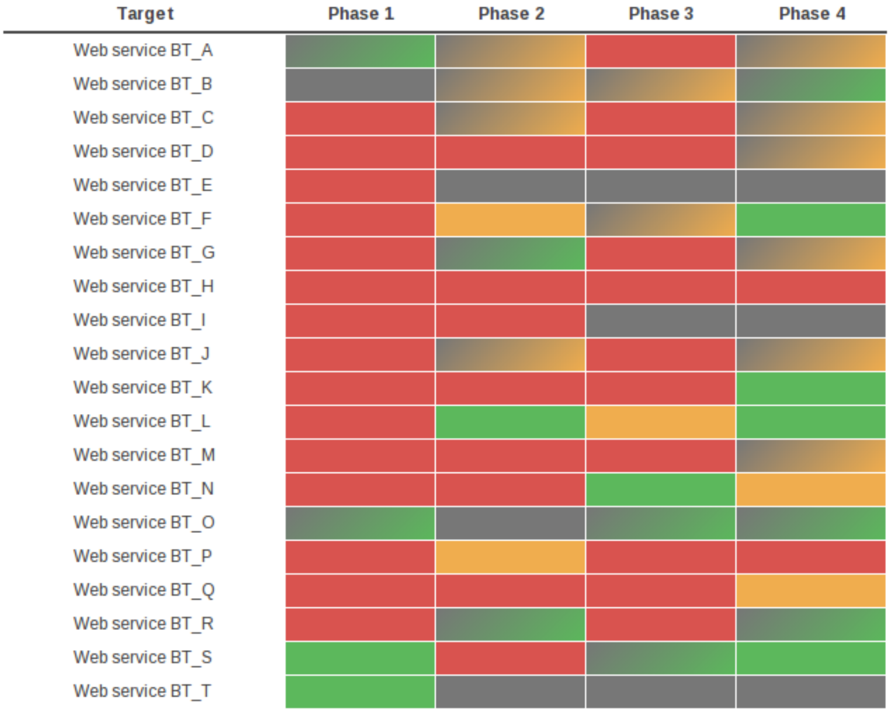


Figure 5.8: Phase analysis by web service.

While chronological summary gives a timeline view of events, statistics gives more general information about the campaign. All views are split into three main parts – successful objective, failed objective and unavailable web service. In this case it was marked unavailable, when the web service was available or functional less than the 10% threshold set previously. The processed event data is visualized for one Blue Team over all web services (figure 5.9) and for one web service over all Blue Teams (figure 5.10).

Figure 5.9 shows that 8 objectives were completed with success. This is 21.6% of all objectives Web Team had to complete for this Blue Team. Sector in green shows objectives that failed and the web service was functional and available. Gray sector also shows failed objectives, however there were some problems with the web service during the phase the objective was meant to be completed. In case of gray sectors Web Team was not successful, but neither was Blue Team, because their web service had problems.

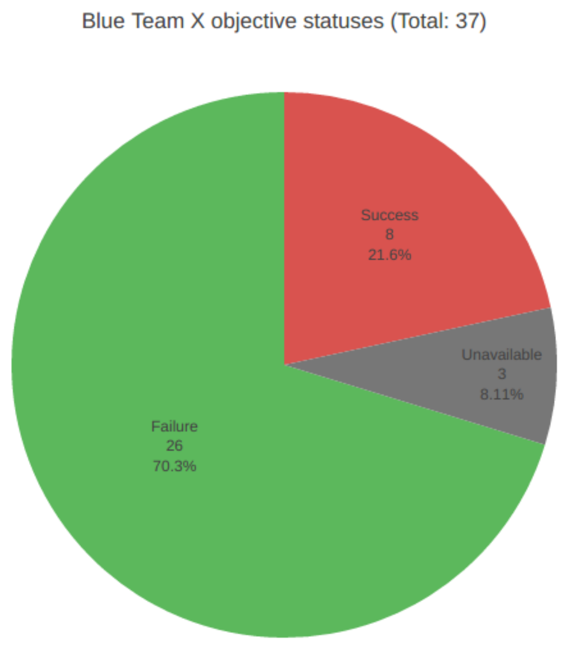


Figure 5.9: Phase statistics for one Blue Team.

Figure 5.10 displays objective statuses for one web service compared to all Blue Teams. The colors have same meaning as in figure 5.9. This view is useful for Web Team leader to compare the achievements of Blue Teams on one web service.

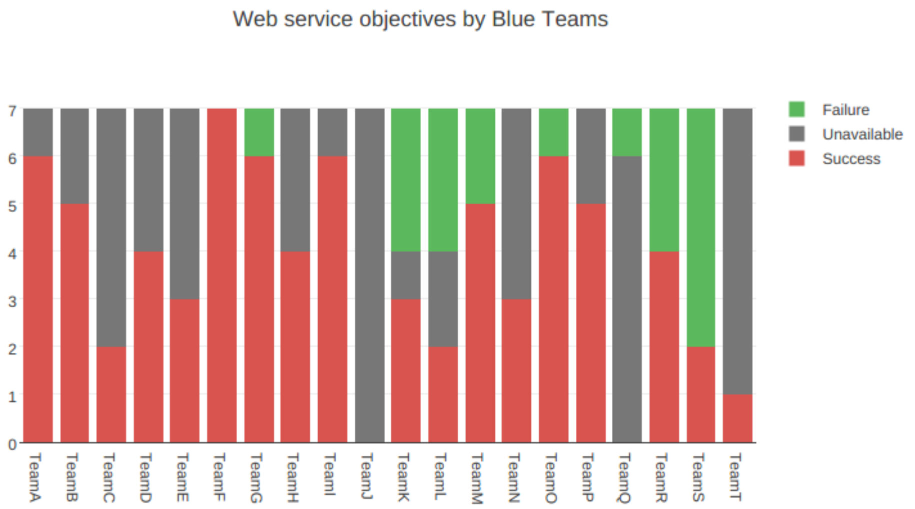


Figure 5.10: Phase statistics for one web service.

To evaluate Web Team campaign, a summary combined from all objectives of every Blue Team is created, similar to figure 5.9. This can be used as an average of Web Team campaign so Blue Teams can compare their endeavors to exercise average.

In addition, the total summary is split by Blue Teams to give a comparison between Blue Teams as shown in sample 5.11. This gives a more general overview of objectives than figures 5.10 and 5.9. From that view Web Team leader has, per Blue Team, overview of all objective statuses.

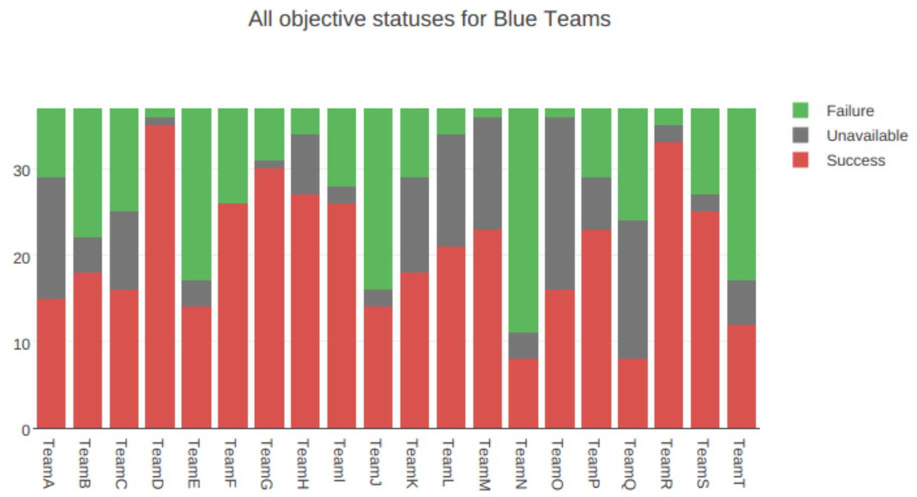


Figure 5.11: Phase statistics for all objectives for all Blue Teams.

5.4.2 Web Team Campaign Timeline

All events collected from exercise environment and Web Team machines are put together on one timeline. Visualizing chronological events allows to identify patterns between different events.

Sample figure of full timeline is shown in appendix A.5. Resolution of the timeline is one minute, which means each block on the timeline represents a minute of a phase.

In each minute of the exercise the timeline displays all events for one web service. The visualization is scaled so all web services for one team fit onto single timeline for each phase. This means that the Y-axis of the timeline is split into parts by number of phases and event types.

For each service the timeline shows following events:

- Phase start time;
- Objective activation;
- HTTP(S) requests from Red Team;

- Service availability status;
- Service functionality status;
- Objective completion;
- Phase end time.

This timeline helps Web team leader to recall Web Team campaign against a web service or compare it to other teams. There are detailed information included with every event so it can be seen what attack vectors Web Team used and how it affected service availability or functionality. Detailed information aides in evaluating the phase summaries.

6. Evaluation of Solution

This chapter evaluates the solution created for Locked Shields 2016. Main points evaluated are visualization of Web Team campaign and data collection tools. The solution was tested in two parts. First part was done during Locked Shields 2016 test-run that was used for testing web request logging tool with Web Team attack scripts. Second part was to collect all necessary data in Locked Shields 2016 main execution.

6.1 Preliminary Test in Locked Shields 2016 Test-Run

Only the logging tool's behavior and suitability for Web Team workflow was tested in test run. The logging tool was provided to three Web Team members each attacking one target. During one day of test exercise approximately 6000 requests were collected. Main problems were involved with traffic collection tool and it's installation.

Initial installation and configuration proved to be difficult, because there were no install scripts prepared. This was one of the reasons so few Web Team members were testing the tool. Logging tool needed to be manually installed and configured.

Proxy portion also had problems with unusual requests and IPv6 traffic. For example one attacker used ICMP protocol, in addition to HTTP, only over IPv6. This proved to be a serious setback because the proxy was built to listen only on HTTP protocol traffic on IPv4. IPv6 listening problem was easily solvable with additional features to program code, but solution for other protocols is left for future work. For this thesis other protocols are left aside and only HTTP(S) traffic is collected and analyzed.

Due to the decreased objectives in test-run not all cases were covered using the logging tool. Some problems with requests and attack types were only found out during main exercise. For example, two Red Team members installed a reverse shell on targets which called them back for instructions. This proved to be difficult to collect.

The proposed architecture worked in test run using Kafka, Elasticsearch and Kibana. In addition to log messages, errors were also sent to central management server through Kafka broker. Elasticsearch with Kibana proved to be useful in discovering the problems with IPv6 and faults in program code.

6.2 Main execution of Locked Shields 2016

This section further explains what was done and what results were concluded in Locked Shields 2016.

6.2.1 Exercise Execution Results

In live exercise run of Locked Shields 2016 information proposed in chapter 5 was collected. To simplify installation, logging tool was updated with install and configuration scripts. Author pre-installed the tool in Web Team Kali machines. During initial install proxy configuration settings were not activated because they needed additional work from machine users. To instruct the activation and usage of the logging tool, simple setup presentation, shown in appendix A.4, was presented to Web Team.

Information collection through message broker proposed in chapter 5 was not prepared for Locked Shields execution. Data was copied from Green Team to central machine instead. This collection methods simplified the design of the proof of concept.

Availability and functionality tests, objective statuses and attack information were gathered from Locked Shields execution environment. There were approximately 16.3 million availability check events, over 2000 objective updates in total. Implemented logging tool collected over 25 000 requests towards Blue Team web services from Red Team.

Availability data included events for every minute about every service for every team. Considering the amount of events happening in Locked Shields environment, manually processing everything is prone to errors or broad generalizations. For example, only considering subjective feedback from attackers or only taking into account objective statuses can lead to too generalized conclusions.

To analyze the data, it had to be converted from format, provided by Green Team, displayed in code example 6.1 to unified format described in code example 5.2. In Web

Team context only nine targets were important so the dataset was reduced by filtering out unimportant events. Information about specific Blue Teams are removed, but these are real events from Locked Shields 2016 main execution.

```
"service6_team_x"|"http-ext"|"OK"|"HTTP OK: HTTP/1.1 200 OK - 3653 bytes in 0.044 second response time"|"1461152650"|"1461152650"
"service6_team_y"|"http.ipv6"|"CRITICAL"|"connect to address target1_team2 and port 80: No route to host"|"1461152648"|"1461152650"
```

Code example 6.1: Availability test format in Locked Shields 2016

Similarly to availability checks the functional test events had to be formated. The format was same as for availability events, so no separate script had to be built.

```
"service6_team_x"|"webservice"|"OK"|"OK: Successfully checked https://service6_team_x"|"1461133527"|"1461133534"
"service6_team_y"|"webservice"|"CRITICAL"|"CRITICAL: Failed check for https://service6_team_y: URL 'https://service6_team_y' didn't load. Error: 'timeout'"|"1461133525"|"1461133534"
```

Code example 6.2: Functionality test format in Locked Shields 2016

During Locked Shields 2016 main execution logging tool collected over 25 000 web requests from Web Team to 20 Blue Teams. Some attackers did use complex techniques, other than simple HTTP requests, to compromise the targets and the logging tool was not able to capture these attacks. Some did not use prepared Kali machines, on which, the logging tool was not tested and did not work. Due to these problems, only five web targets out of nine yielded considerable amount of attack data to be analyzed. These problems are taken into account in future work context.

Continuous deployment of logging tool proved to be useful because of several unforeseen complications with logging tool. During main execution author deployed twelve iterations of logging tool with software fixes.

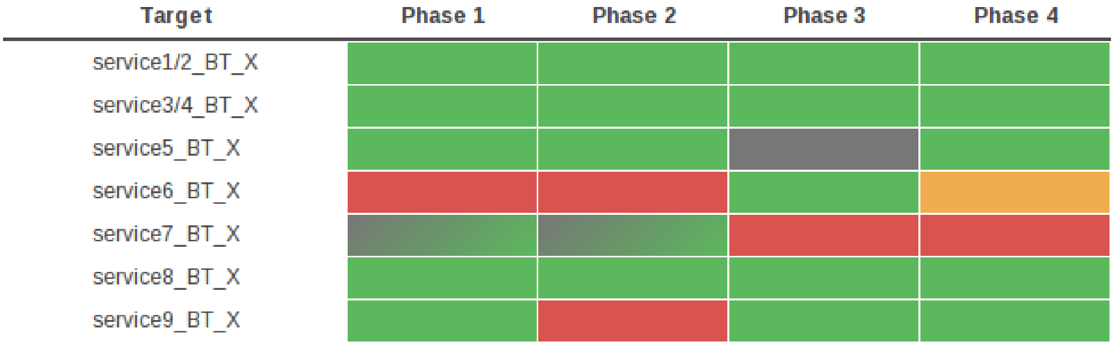
As additional feature, storing data to local machine before sending it to management server, was also implemented. The data saved locally was used to assess the Kafka pipeline efficiency in Locked Shields network with execution loads. Comparing locally saved results and ones in the central server showed approximately 80% loss of data from one Web Team member. This could be due to large loads and insufficient memory of Kafka process. Comparison with locally saved data showed that the architecture is not yet perfect and additional test must be done to validate it further.

6.2.2 Phase Overview from Locked Shields 2016

From all collected data, a detailed information about Web Team campaign was compiled using several different views as described in chapter 5. To evaluate the information compiled from collected data, a one time feedback form was created. The form was kept simple and had one feedback field as free text. This form, shown in appendix A.2, was filled by Web Team members after Locked Shields 2016 execution. It was subjective information from attacker about their actions to show the level of detail in their response and to correlate it to generated information. Results of the feedback are added to appendix A.3.

Phase summary views were created from collected data as described in Section 5. These views were correlated with Web Team feedback results where the level of detail allowed.

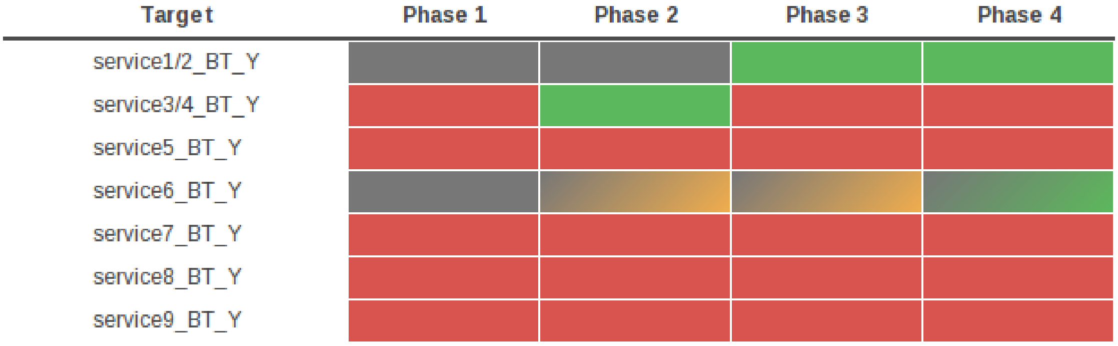
In almost all responses team BT_X is mentioned for good defense. Comparing team BT_X timeline on figure 6.1 with team BT_Y's timeline on figure 6.2 it shows visible difference in their service availability and completed Red Team objectives.



Target	Phase 1	Phase 2	Phase 3	Phase 4
service1/2_BT_X	🟩	🟩	🟩	🟩
service3/4_BT_X	🟩	🟩	🟩	🟩
service5_BT_X	🟩	🟩	⬛	🟩
service6_BT_X	🟥	🟥	🟩	🟧
service7_BT_X	▨	▨	🟥	🟥
service8_BT_X	🟩	🟩	🟩	🟩
service9_BT_X	🟩	🟥	🟩	🟩

Figure 6.1: Blue Team X exercise summary.

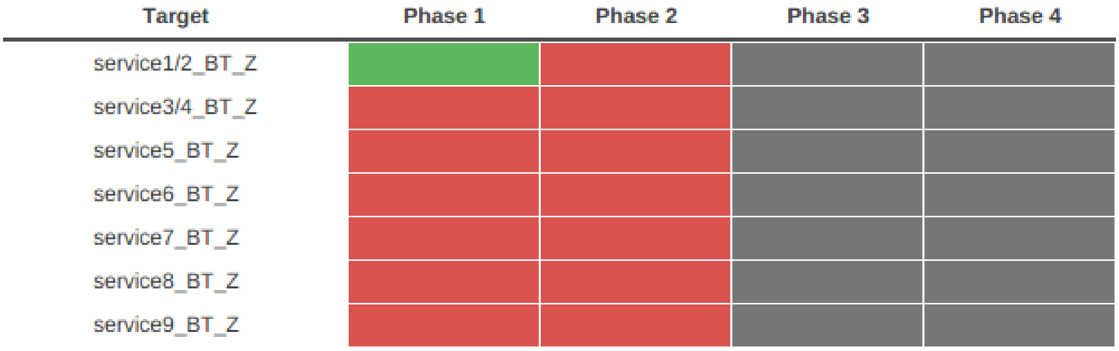
Blue Team Z phase summary view, on figure 6.3, shows that their services were not available starting from phase 3. From the Web Team feedback responses, it is mentioned on multiple occasions that BT_Z was not available since phase 2. This demonstrates that



Target	Phase 1	Phase 2	Phase 3	Phase 4
service1/2_BT_Y	⬛	⬛	🟩	🟩
service3/4_BT_Y	🟥	🟩	🟥	🟥
service5_BT_Y	🟥	🟥	🟥	🟥
service6_BT_Y	⬛	⬛🟧	⬛🟧	⬛🟩
service7_BT_Y	🟥	🟥	🟥	🟥
service8_BT_Y	🟥	🟥	🟥	🟥
service9_BT_Y	🟥	🟥	🟥	🟥

Figure 6.2: Blue Team Y exercise summary.

generated views show similar result than Web Team members experienced.



Target	Phase 1	Phase 2	Phase 3	Phase 4
service1/2_BT_Z	🟩	🟥	⬛	⬛
service3/4_BT_Z	🟥	🟥	⬛	⬛
service5_BT_Z	🟥	🟥	⬛	⬛
service6_BT_Z	🟥	🟥	⬛	⬛
service7_BT_Z	🟥	🟥	⬛	⬛
service8_BT_Z	🟥	🟥	⬛	⬛
service9_BT_Z	🟥	🟥	⬛	⬛

Figure 6.3: Blue Team Z exercise summary.

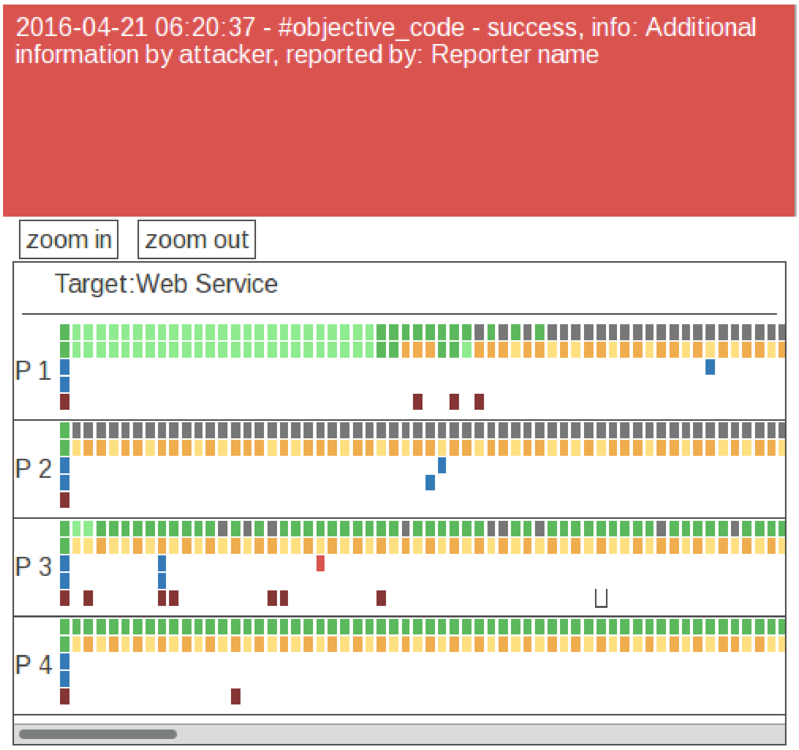
In addition, it gives more clear information in comparison to notes from Web Team members, for example when looking at service 6 feedback. Blue Team A is shown in service 6 feedback both as bad team and good team with no additional details. In result, this framework allows Web Team now focus more on objectives and less on taking notes about service availability.

Similar summaries were made manually for Locked Shields 2015 After Action Report [8]. This framework adds more detail including availability and functionality to the analysis. More importantly this is automated and scalable to any number of Blue Teams. Also with the addition of specific threshold for availability, automated analysis can be more accurate than human estimation.

6.2.3 Locked Shields 2016 Web Team Campaign Timeline

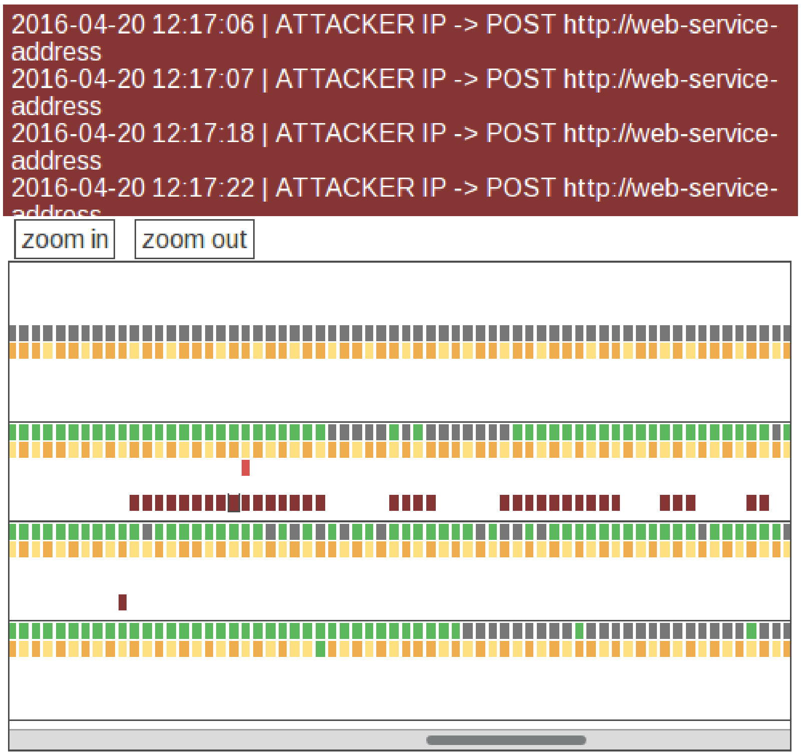
While views by phase give an overall understanding of the campaign, a detailed timeline with all events ables to see specific details about a web service per Blue Team. Full scale timelines about one web service of two Blue Teams are shown in appendix A.5.

A sample, from full scale timeline on figure 6.4 shows detailed information about a successful objective. It is possible to correlate successful objectives with Web Team requests shown in sample 6.5. A possible attack payload could be extracted from the detailed request information.



2016-04-21 06:20:37 - #objective_code - success, info: Additional information by attacker, reported by: Reporter name

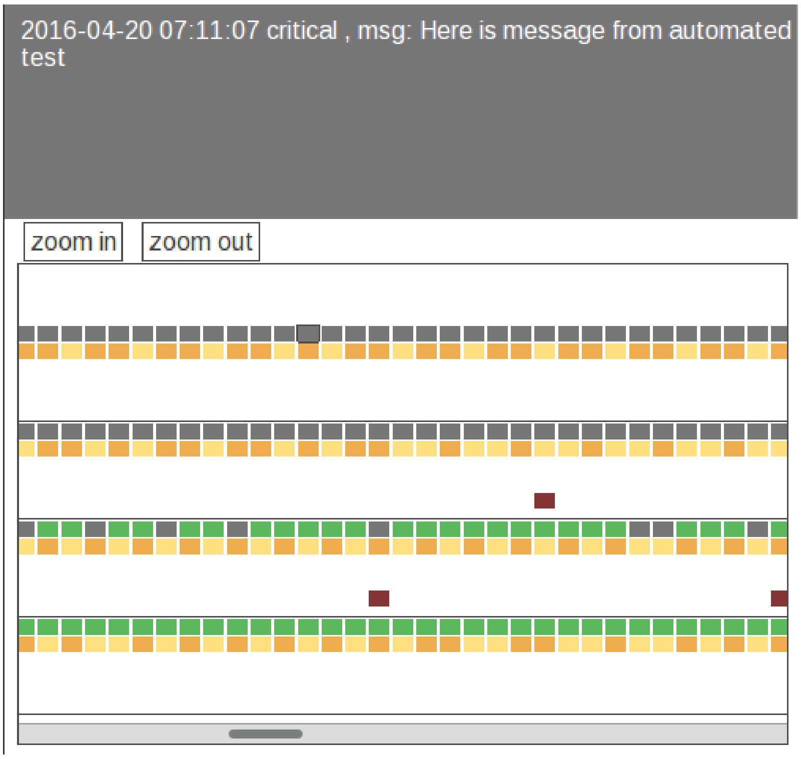
Figure 6.4: Web Team campaign timeline objective example.



```
2016-04-20 12:17:06 | ATTACKER IP -> POST http://web-service-address
2016-04-20 12:17:07 | ATTACKER IP -> POST http://web-service-address
2016-04-20 12:17:18 | ATTACKER IP -> POST http://web-service-address
2016-04-20 12:17:22 | ATTACKER IP -> POST http://web-service-address
```

Figure 6.5: Web Team campaign timeline HTTP request example.

It is also possible to zoom in the timeline to enlarge the event blocks for easier navigation. This is included, because the detailed information is shown when user clicks on the blocks and in full scale, it is more difficult.



2016-04-20 07:11:07 critical , msg: Here is message from automated test

Figure 6.6: Web Team campaign timeline zoom example.

The full scale timelines shown in appendix A.5 give a possibility to correlate different events for one web service. In addition, they allow Web Team leader to recall the campaign in detail after execution.

6.2.4 Statistical Views

Statistics from Web Team attacks was done, to give Web Team leader more general overview of the campaign than timelines.

Figure 6.7 shows statistics on how many attacks used encrypted connections for attack (HTTPS) and how many attack payloads may have been hidden in POST request body. The figure shows data about the exercise in general but similar figures can be generated for each team. Additional figures about relationship between DNS names and IP addresses usage is show in appendix on figure A.5.

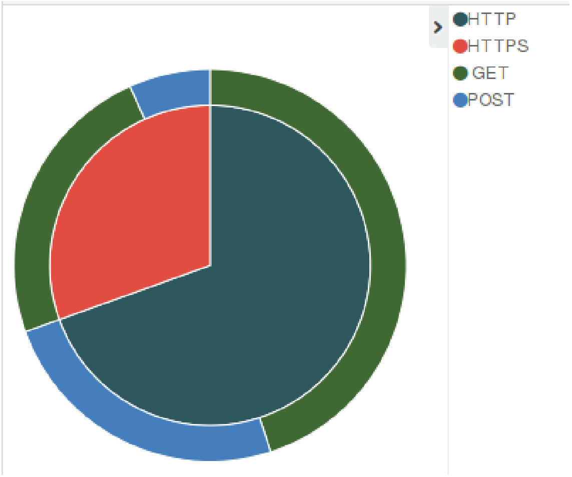


Figure 6.7: HTTP protocol analysis.

Level of detail added by availability tests can be seen when comparing different statuses on figure 6.8 and figure 6.9. On those figures, 6.8 shows all successful objectives split by Blue Teams and 6.9 adds objectives failed because of availability. When looking only successful objectives, Blue Team O and Blue Team Q seem to have had good defenses. However, when adding availability factor then it is clear that their web services were not available most of the time. Ability to make this distinction allows better evaluation of Web Team campaign and comparison of Blue Teams.

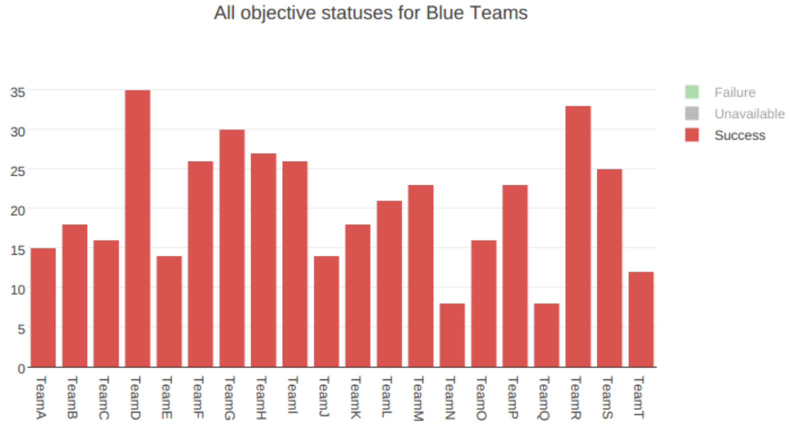


Figure 6.8: Summary of all successful objectives.

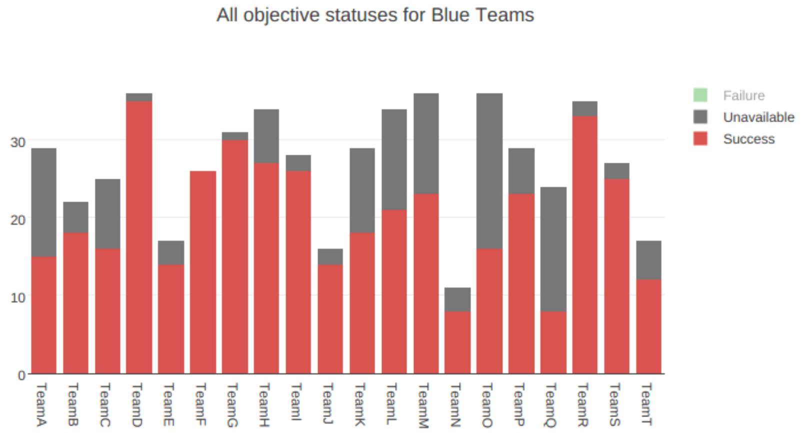
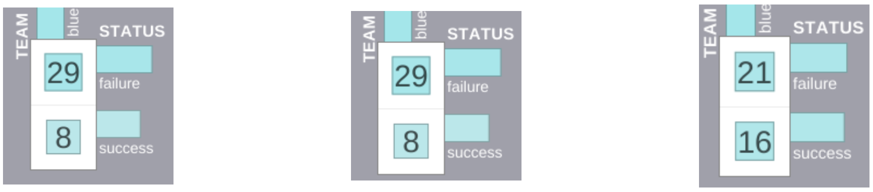


Figure 6.9: Summary of all successful and unavailable objectives.

Objective summaries are also part of situational overview managed by Yellow Team. On figure 6.10, objective statuses, as seen by Yellow Team, are shown [8]. This shows the same shortage of details as shown on figures 6.8 and 6.9. When looking only success and failure statuses Blue Teams N and Q have same same numbers. However figure 6.9 clearly shows that Blue Team Q failed to keep their services available most of the time.

Web Team objectives for Blue Team O (figure 6.10c) failed more than succeeded according to Yellow Team. However, taking into account the times site was unavailable, it is seen that most attacks failed due to poor availability.



(a) Blue Team N.

(b) Blue Team Q.

(c) Blue Team O.

Figure 6.10: Objective statuses from Yellow Team.

For Locked Shields 2016 After Action Report, views shown in figure 5.9 were created for every Blue Team. Previously, it would have been extensive work due to large amount of teams and events to create these statistical views. As mentioned in introduction, the number of Blue Teams has grown every year, therefore scalable, automated tools are very useful.

6.3 Future Work

This section includes future work author plans on doing on the framework. Initial testing and proof of concept showed a lot of promise in regards of Web Team campaign overview. Plan is to develop fully featured framework that would be useful for different technical CDXs.

6.3.1 Improvements on Web Team HTTP Request Collection

Request collection showed some problems in Locked Shields 2016 main execution. Regarding the proxy solution it only worked for those who used Kali machine provided in Locked Shields. Main problems were with difference in software versions the tool was dependent on.

Also the improvements could include network sniffing solution to capture more than HTTP(S) traffic towards Blue Teams. Proxy solution has some advantages over network sniffing. Mainly it is possible to identify Web Team member who made the requests even if they change IP address. However it is not able to capture other traffic than HTTP and HTTPS.

Some problems with data transport implementation caused a loss in transmission. When comparing data stored locally and transported through network, it revealed a 35% of loss, due to heavy loads. Therefore the whole architecture could be tested more comprehensively.

The attacks regarding vulnerabilities and payloads could be mapped to add additional level of detail to attack events. However, in changing environment of Blue Team systems in real exercise, pre-mapped attacks have little use. Best would be to apply an intrusion detection rules[1] to capture requests, but Web Team uses tactics to bypass known rules [8]. Therefore the classification of HTTP requests to attacks and simple reconnaissance is substantial work.

[1] `https://rules.emergingthreats.net/`

6.3.2 Real-Time Situation Awareness of Web Team Campaign

For future work, improvements on the collection tool and real-time overview could be done. The real-time overview with Kibana and Elasticsearch worked for development purposes. Similarly the availability and functionality test data could be added to the real-time view. This way, Web Team members can have an overview of web service situation regarding availability.

With real-time overview the tool helps Clarified Security to conduct CDXs in slow-play format as described in section 2.2. Real-time overview of execution data gives instant feedback to Blue Teams and allows to focus on weak points.

On top of the real time data, alerts can be implemented, when a web service becomes available. Some services are only available small amount of time. The attacker has too much work to check service status every minute. This would help to complete the objectives set.

6.3.3 Visualization Improvements

Scripts used for creating the views could be brought together into a web application based tool. These scripts output HTML formated views so they would integrate easily to a web application. The visualizing tool could propose all possible views and generate them in background when requests. For some views, like full scale timeline, the generation takes time and in such case would be a one time action. All views could be cached and only re-generated when new data is available.

Also the views could be linked together. For example the phase summary could provide a detailed timeline when clicked on a phase. Currently, this was not necessary, but would simplify post execution overview and analysis even more.

From the generated campaign overview the Web Team leader must still create After Action Reports. This is manual work and although the solution in this thesis reduces that workload it is not scalable. The solution could be developed to automatically combine a base for the report, so less manual work should be done.

7. Conclusion

Popularity of Cyber Defense Exercises grows each year and so does the number of Blue Teams in Locked Shields, which is the largest CDX in the world. In 2016, there were 20 Blue Teams in the exercise. The large workload does not leave time to make detailed notes about the Web Team actions. That in turn limits the level of detail in feedback to Blue Teams.

Detailed feedback about exercise events helps Blue Teams reinforce learning goals. However Web Team is occupied with conducting and reporting attacks on Blue Teams and their priority is to provide equal attention to each Blue Team. Main problem was that Web Team is unable to provide feedback with enough detail about their campaign, due to heavy workload.

Main goal for the thesis was to collect and visualize data from Cyber Defense eXercise environment to provide detailed information about Web Team campaign. For that, important data from exercise environment was identified and gathered to central location. After the exercise, all data was processed and a detailed overview about the campaign was provided. Two timelines with different levels of detail were provided in addition to other statistics.

Objective data was analyzed, to see how well Blue Team did on defending web services. Availability and functionality data gives additional dimension to that. Furthermore, including web request data from Web Team gives a more detailed overview of the whole campaign.

A logging tool was proposed to gather data about requests made from Web Team to Blue Team web services. The proof of concept tool was installed in Web Team Kali machines in Locked Shields 2016 environment, where it intercepted all HTTP(S) traffic. Intercepted traffic that was going towards Blue Teams was also sent to central management server for processing. During Locked Shields 2016 execution, the data was also displayed in real-time using Kibana and Elasticsearch, to more quickly add more features and debug errors.

Continuous deployment method was employed for Web Team HTTP traffic logging tool. This meant that it could be updated at any time during the exercise execution. The possibility was used to deploy twelve iteration of logging tool during two days of execution.

From collected data, different detailed views about each team were created, to provide overview about Web Team campaign. Views included objective summaries by phase and full timeline with detailed event descriptions. Objective status summaries provided more general overview for individual Blue Teams and Web Team leader. The created views were evaluated by Web Team leader, by using them as a base for After Action Report of Locked Shields 2016.

Set goal for the thesis was accomplished – detailed information about Web Team campaign was collected and visualized. The solution is also scalable and improves the level of detail in campaign overview it was useful for Web Team leader for creating After Action Report. Until now, collecting information to create the reports were done manually and data was spread through exercise environment. This framework collects all data together, allows creation of different views and statistics. Scalability of the solution means that it can be used for increasing number of Blue Teams.

References

[1] Hille Lepp. e-Estonia. URL http://estonia.eu/about-estonia/economy-a-it/e-estonia.html. Accessed 2016-05-01.

[2] Gen. Larry D. Welch USAF (Ret.). Cyberspace – the Fifth Operational Domain, 2011. URL https://www.ida.org/~/media/Corporate/Files/Publications/ResearchNotes/RN2011/2011%20Cyberspace%20-%20The%20Fifth%20Operational%20Domain.pdf. Accessed 2016-05-01.

[3] CCDCOE. Cyber Security Strategy Documents, 2016. URL https://ccdcoe.org/cyber-security-strategy-documents.html. Accessed 2016-05-01.

[4] ISO. ISO 22398:2013 Societal security – Guidelines for exercises (Preview), 2013. URL https://www.iso.org/obp/ui/#iso:std:iso:22398:ed-1:v1:en. Accessed 2015-05-07.

[5] Adrien Ogee, Razvan Gavrila, Panagiotis Trimintzios, Vangelis Stavropoulos, and Alexandros Zacharis. The 2015 Report on National and International Cyber Security Exercises Survey, Analysis and Recommendations, 2015. URL https://www.enisa.europa.eu/activities/Resilience-and-CIIP/cyber-crisis-cooperation/cce/cyber-exercises/latest-report-on-national-and-international-cyber-security-exercises. Accessed 2016-04-11.

[6] NATO CCDCoE. Locked Shields 2015 Executive Summary. Technical report, NATO Cooperative Cyber Defence Centre of Excellence, 2015.

[7] Aare Reintam, Locked Shields 2016 Exercise Manager, personal communication, April 2016.

[8] Elar Lang, Web Application attack sub-team leader in Locked Sields 2016, personal communcation, April 2016.

[9] Mehis Hakkaja, CEO of Clarified Security and Red Team Leader in Locked Shields 2016, personal communication, April 2016.

[10] NATO CCDCoE. Cyber Defence Exercise Locked Shields 2013. Technical report, NATO Cooperative Cyber Defence Centre of Excellence, Apr 2013. URL `https://ccdcoe.org/publications/LockedShields13_AAR.pdf`.

[11] Panagiotis Trimintzios and Razvan Gavrila. National and International Cyber Security Exercises: Survey, Analysis & Recommendations, 2012. URL `https://www.enisa.europa.eu/publications/exercise-survey2012`. Accessed 2016-05-01.

[12] Dimitris Gritzalis and Spyros Papageorgiou. PANOPTES: The Greek National Cyber Defence Exercise, 2016. URL `https://www.cis.aueb.gr/Publications/CEER-ENISA-2016%20Gritzalis%20Papageorgiou.pdf`. Accessed 2016-04-13.

[13] Pavel Čeleda, Jakub Čegan, Jan Vykopal, and Daniel Tovarňák. KYPO – A Platform for Cyber Defence Exercises, 2015. URL `https://is.muni.cz/repo/1319597/kypo-paper-msg-133.pdf`. Accessed 2016-04-13.

[14] Jakub Čegan, Martin Vizváry, et al. Lessons learned from kypo–cyber exercise & research platform project. 2015.

[15] Razvan Gavrila, Adrien Ogée, Panagiotis Trimintzios, and Alexandros Zacharis. ENISA CE2014 After Action Report Public Version. Technical report, ENISA, 2015. URL `https://www.enisa.europa.eu/activities/Resilience-and-CIIP/cyber-crisis-cooperation/cce/cyber-europe/ce2014/ce2014-after-action-report`.

[16] Jason Kick. Cyber Exercise Playbook, 2015. URL `https://www.mitre.org/publications/technical-papers/cyber-exercise-playbook`. Accessed 2015-01-01.

[17] Margus Ernits, Johannes Tammekänd, and Olaf Maennel. I-tee: A fully automated cyber defense competition for students. *SIGCOMM Comput. Commun. Rev.*, 45(4): 113–114, August 2015. ISSN 0146-4833. doi: 10.1145/2829988.2790033. URL `http://doi.acm.org/10.1145/2829988.2790033`.

[18] Küberolümpia, 2015. URL http://www.kyberolympia.ee/competition. Accessed 2016-04-05.

[19] Risto Vaarandi and Paweł Niziński. A Comparative Analysis of Open-Source Log Management Solutions for Security Monitoring and Network Forensics. Technical report, CCDCOE, 2013. URL https://ccdcoe.org/sites/default/files/multimedia/pdf/VaarandiNizinski2013_Open-SourceLogManagementSolutions.pdf.

[20] John Vanderzyden. Welcome to the ELK Stack: Elasticsearch, Logstash, and Kibana, 2015. URL https://qbox.io/blog/welcome-to-the-elk-stack-elasticsearch-logstash-kibana. Accessed 2016-04-14.

[21] Splunk Enterprise. Splunk Enterprise - The Platform for Operational Intelligence, 2015. URL http://www.splunk.com/content/dam/splunk2/pdfs/data-sheets/splunk-product-data-sheet.pdf. Accessed 2016-04-14.

[22] C. Lonvick. The BSD syslog Protocol, 2015. URL https://www.ietf.org/rfc/rfc3164.txt. Accessed 2016-05-01.

[23] Steven Wu. Evolution of the Netflix Data Pipeline, 2016. URL http://techblog.netflix.com/2016/02/evolution-of-netflix-data-pipeline.html. Accessed 2016-04-05.

[24] K. Goodhope, J. Koshy, J. Kreps, N. Narkhede, R. Park, J. Rao, and V. Yang Ye. Building LinkedIn's Real-time Activity Data Pipeline, 2012. URL http://sites.computer.org/debull/A12june/pipeline.pdf. Accessed 2016-04-05.

[25] Todd Palino. Running Kafka At Scale, 2015. URL https://engineering.linkedin.com/kafka/running-kafka-scale. Accessed 2016-04-05.

[26] AbdulFattaah Popoola. Design Patterns: PubSub Explained, 2013. URL https://abdulapopoola.com/2013/03/12/design-patterns-pub-sub-explained/. Accessed 2016-05-11.

[27] Colin Ware. *Information Visualization: Perception for Design*. Elsevier, 2012.

[28] Johannes Landstorfer, Ivo Herrmann, Jan-Erik Stange, Marian Dörk, and Reto Wettach. Weaving a Carpet from Log Entries: A Network Security Visualization Built with Co-Creation, 2014. URL http://complexdatavisualized.com/

wp-content/uploads/2014/07/pixelcarpet-VAST2014_v104-novideo.pdf. Accessed 2016-05-10.

[29] Lihua Hao, Christopher G. Healey, and Steve E. Hutchinson. Flexible Web Visualization for Alert-Based Network Security Analytics, 2013. URL https://www.csc.ncsu.edu/faculty/healey/download/vizsec.13.pdf. Accessed 2016-05-10.

[30] Yarden Livnat, Jim Agutter, Shaun Moon, and Stefano Foresti. Visual Correlation for Situational Awareness. URL http://www.sci.utah.edu/~yarden/papers/VisAware.pdf. Accessed 2016-05-10.

[31] Shanks Wylie. Enhancing Intrusion Analysis through Data Visualization. Technical report, SANS, 2014. URL https://www.sans.org/reading-room/whitepapers/detection/enhancing-intrusion-analysis-data-visualization-35757.

[32] NATO CCDCoE. Cyber Defence Exercise Locked Shields 2012. Technical report, NATO Cooperative Cyber Defence Centre of Excellence, Apr 2012. URL https://ccdcoe.org/publications/LockedShields12_AAR.pdf.

[33] S. Foresti, J. Agutter, Y. Livnat, S. Moon, and R. Erbacher. Visual correlation of network alerts. *IEEE Computer Graphics and Applications*, 26(2):48–59, March 2006.

[34] Raphael Mudge. Telling the Offensive Story at CCDC, 2013. URL http://blog.cobaltstrike.com/2013/05/30/telling-the-offensive-story-at-ccdc/. Accessed 2016-04-05.

[35] Raphael Mudge. Rethinking Reporting for Red Team Operations, 2015. URL http://blog.cobaltstrike.com/2015/09/09/rethinking-reporting-for-red-team-operations/. Accessed 2016-04-05.

[36] Alistair Cockburn. Using Both Incremental and Iterative Development, 2008. URL http://www.se.rit.edu/~swen-256/resources/UsingBothIncrementalandIterativeDevelopment-AlistairCockburn.pdf. Accessed 2016-05-07.

[37] Arun Gupta. Continuous Integration, Delivery, Deployment and Maturity Model, 2015. URL http://blog.arungupta.me/continuous-integration-delivery-deployment-maturity-model/. Accessed 2016-04-05.

A. Appendixes

A.1 Interview With Elar Lang

Following section is a semi-structured interview with Locked Shields 2016 Web Team leader Elar Lang. Interview was conducted in 9th March, in person, at Locked Shields Red Team workshop.

Q: What are your responsibilities in Locked Shields?

> I have participated in Locked Shields, as Web Team member, from 2012. Since 2013 I have been Web Team leader. This means I manage Web Team work- and information flow. Also, I am responsible for training new members and relaying experience from previous exercises in a one day workshop. After the exercise I must compile information about the exercise for feedback to Blue Teams as they are the main training audience.

Q: Describe the Web Team organization and structure.

> In Locked Shields 2016 Web Team has 14 members and 9 targets to exploit. This year one or two members will attack one web service for all Blue Teams. That allows them to prepare scripts for attacks. When Blue Teams start to defend their web services and prepared attack scripts will fail, then Web Team members can use manual methods to exploit vulnerabilities. Web Teams has to complete objectives in each phase, by attacking Blue Teams systems. These objectives are tasks Web Team must do and what can be the proof needed to provide. If there is a successful attack Blue Teams lose points. Attacks range from defacements to data stealing. Later in the exercises we are allowed to destroy web services and shut down servers.

Q: What tasks Web Team members have?

One or two web team members will be responsible of one web service. Depending on the service it will have one or two objectives, that could be repeated each phase. Web Team members' tasks are to complete the objective by exploiting vulnerabilities and report success or failure to White Team. It is important that each Blue Team has equal attention from Web Team. Then Web Team has to document their actions and observations. This is necessary for feedback, however it is done when there is time. Between reporting and attacking 20 Blue Teams there is little free time for attackers.

Q: How much time reporting the objective usually takes?

This has never been measured before and would need an extra person for each Web Team member to measure. If we consider only the reporting, depending on the experience of the reporter, it can take 1 to 3 minutes. This means activating it, uploading the proof, adding some comments and closing it. However in exercise situation, often the services do not respond or function as needed. Since the objective can be pursued during one phase and if Blue Teams cannot repair their services the objective has failed. When considering, that objectives can be repeated and there are 20 teams, Web Team must report over 700 events during the exercise.

Q: What tools and techniques Web Team uses to achieve their objectives?

Web Team does not use common framework or tools. This makes it harder for Blue Teams to discover attackers. Often it is necessary to go back to manual methods for exploits due to Blue Teams' defenses. Web Team members cannot always rely on tools and have to be able to exploit vulnerabilities only using a browser or command line. When every team member builds their own attack scripts and exploits it is harder to detect and there is a learning side for Web Team as well.

Q: How Web Team campaign relates to Locked Shields scenario?

Web Team's objectives are scored by White Team. Each successful objective gives minus points to Blue Teams. In addition to objectives, Web Team is urged to extend their presence in Blue Team systems. When attackers shut down web services it manifests in scoring as negative availability. Web Team is like a service provider in Locked Shields and so our highest priority is to provide equal attention to every Blue Team.

Q: Could you describe the targets in Locked Shields?

Targets are built using different web technologies to provide variety of services. These targets, that are given to Blue Teams, are built in cooperation with Web Team, therefore we know attack vectors and vulnerabilities beforehand. Blue Teams usually cannot defend against all attacks and that is not really their goal. Web Team must create a stressful situation and Blue Teams have to maintain control over services, report the attacks and cooperate.

Q: What are Blue Teams' tasks regarding the Web Targets?

Their main task is to keep control of the services, keep them up and functionality working. Therefore it is not acceptable defense to just shut the service down if they are required to maintain it. Also removing functionality, like file upload or commenting, is really breaking the rules. Everything that must work is described in game rules. Sometimes Blue Teams try to play the rules and calculate if it is more beneficial for them to shut down the service than to be hacked by web team. Previous years they have changed dynamic websites with static HTML to remove attack vectors and keep availability scores.

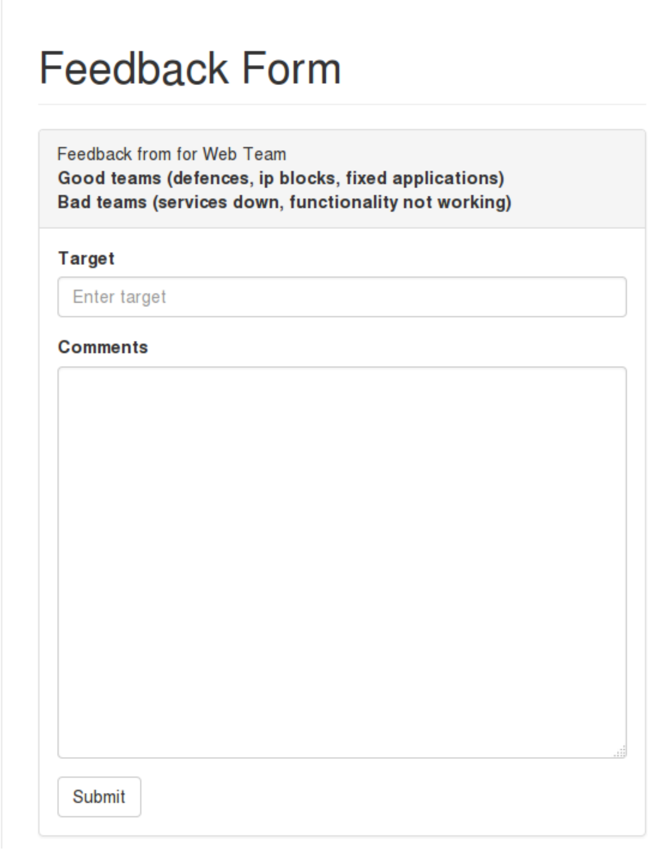
Q: What information you give for Blue Teams in feedback sessions?

Feedback is given to Blue Team in two forms – presentation and written report. First, a short conclusion is made immediately after execution to give general information from web team point of view. The information is collected from Web Team members and presented to Blue Teams. It also includes how many objectives failed and how many were successful and general description of attack methods. Second is a hotwash session next day after execution. There each team can bring out important details about execution and have a longer overview than on previous day. Third is a presentation of after action report, where a more detailed look on the campaign should be made. This presentation takes place a month after main execution, so all teams have time to go over the notes. The report itself is kept quite general, because we don't want to give out detailed descriptions of exploit and web team scripts. However, currently there isn't a possibility to give very detailed feedback. Most information consists of subjective observations of Web Team members, if they have time to write it down.

Q: How feedback could be improved?

In Locked Shields 2015 there were initial tests regarding web service functionality, starting from Locked Shields 2016, Green Team started to make functionality tests that are also scored. This information should be reflected in feedback. Currently all information is in different places. Objective reports are in White team hands, availability test results are held by Green Team. Each Web Team member tries to make his own notes and observations. To look at all the data and make conclusions for all 20 teams is substantial work. This information should be all in one place and easy to interpret. I would like to see a timeline of all events during the exercise, results of objectives in each phase and statistics of how many of the objectives were completed and how many failed. Also, it is good to have a comparison between Blue Teams. Automated tools would improve the situation a lot, currently, creating the feedback is manual work left for Web Team leader. Creating feedback from large datasets that are not nicely organized or even in single place takes too much time to be beneficial. Therefore it is no realistic to provide valuable feedback to Blue Teams.

A.2 Web Team Feedback form



Feedback Form

Feedback from for Web Team

Good teams (defences, ip blocks, fixed applications)

Bad teams (services down, functionality not working)

Target

Enter target

Comments

Figure A.1: Feedback form for Web Team.

A.3 Web Team feedback response

```
Service1:
    We were able to exploit the vulnerabilities of the web
   interface and plant a backdoor on the server for the
   following BTs:
    BT_V, _, _, _, _, _, _, BT_Z, BT_W
    The web service of the Service1 has been available for
   the first 2 hours of day 1 for those BTs that we could
   successfully attack.
    The other BTs shut it down since the beginning of the
   exercise since it was not listed in the required
   services.
    Other countermeasures were probably taken but since we
   could not connect to the web servers (filtered traffic?)
    we cannot tell if they could actually fix the
   vulnerabilities.

    _, _, _, _, _, and BT_W did not notice the installed
   backdoor, and it worked for 2 days probing back to us.

    BT_V, _, and BT_Z found out our backdoor and cleaned
   the target server. then shut down http service.

Service2:
    We were able to exploit the vulnerabilities ... and
   plant a backdoor on the server for the following BTs:
    _, BT_V _, _, BT_U, _, BT_Z
    We did not use the vulnerability of the WEB interface
   since it has been disabled for most of the BTs since it
   was not listed in the required services.
    ...
    Other countermeasures were probably taken but since we
   could not connect to the web servers (filtered traffic?)
    we cannot tell if they could actually fix the
   vulnerabilities.
```

Service3/4:

BT_U - were vulnerable in the beginning, but fixed the vulnerablities on the second day

BT_Z - were vulnerable in the beginning, and then down

BT_W - have been vulnerable to defacement in all phases , and also were down at the end

BT_X - good defence, remove pdf thunbnail generating vulnerability //// GOOD, BEST ONE

BT_Y - no proper defences, removed users from db, after adding them back they were vulnerable

BT_V - Vulnerable during day 1, during day 2, DOWN

_ blocked file upload functionality for users, but I was able to exploit the vulnerability with previously uploaded pdf

_ had excellent SQL-Injection defense and also weren't vulnerable to shellshock. But they also deleted application users

_ were vulnerable in the first phases and down in the last phases. Not much defense visible

_ was down in 3 of 4 phases - without our intervention

_ was vulnerable in the first phases; then they deleted the backdoors, fixed shellshock, but also deleted the application users (at least 3)

_ were vulnerable in the beginning, and then down

_ blocked file upload functionality for users, but I was able to exploit the vulnerability with previously uploaded pdf

_ When it was running it was vulnerable, no defences

_ When it was running it was vulnerable, no defences

_ Fixed pdf upload vulnerability bad way. But they do not remove previously uploaded backdoor before fixing vulnerability, so remain vulnerable during whole excercise

_ Vulnerable during whole excercise, no defences at all

_ was vulnerable in the beginning, but fixed the vulnerabilities on day 2
_ was down most of the time
_ Vulnerable during day 1, during day 2 DOWN
_ Vulnerable during day 1, during day 2 DOWN

To sum it up Only team number BT_X use proper defence.

Service5:
...(Description of target)...

I would like to mention teams number BT_X, BT_V and _ who kept up the service most of the time, and were the fastest to fix the vulnerabitlities.

Service6:
bad teams:
* site not accessible, no functionality: _,_,_,BT_U
* no or missing defenses: _,_,_,_,_,_,BT_W

good teams/solutions:
* _
* BT_U

Service7:
_:best defence
BT_X:good defence
BT_U:good defence

Other teams:no defence

Service8:
Best Teams:
BT_X, _ - vulnerabilities patched very fast and were available at all times

Good teams

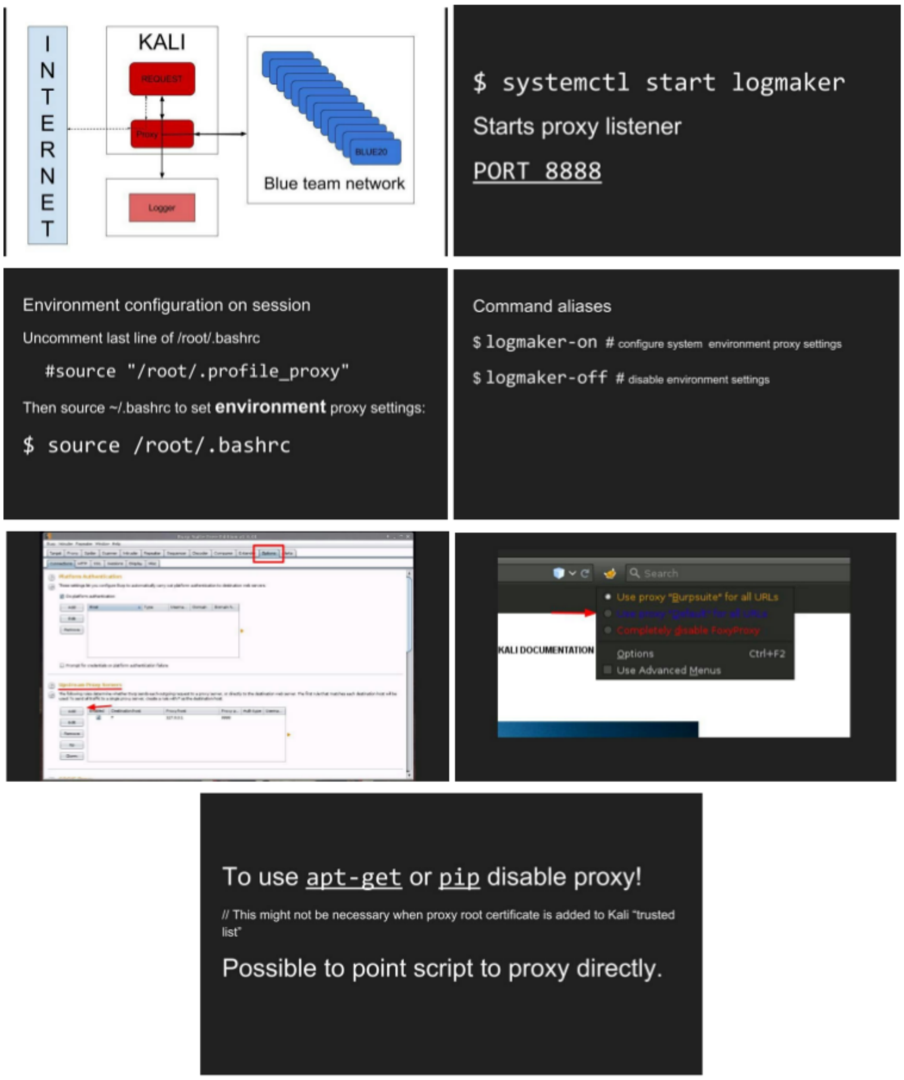
```
 BT_U _ - vulnerabilities patched after a while but were
 available not from start in PHASE 2, later in PHASE 3-4
 patched and available most of the time
 _, _, _, - patched at last but not available mostly

 Bad teams
 PHASE 2:
 BT_V - service not available all the time
 PHASE 3:
 BT_V, _, _, _, BT_Z - service not available all the
time
 PHASE 4:
 BT_V, _, _, _, BT_Z - service not available all the
time
```

Code example A.1: Web Team feedback response.

A.4 Web Team tool setup guide

This appendix includes the install manual presented to Web Team memebers for logging tool installation.



```
$ systemctl start logmaker
```

Starts proxy listener

PORT 8888

Environment configuration on session

Uncomment last line of /root/.bashrc

```
#source "/root/.profile_proxy"
```

Then source ~/.bashrc to set **environment** proxy settings:

```
$ source /root/.bashrc
```

Command aliases

```
$ logmaker-on # configure system  environment proxy settings
$ logmaker-off # disable environment settings
```

Search
Use proxy "Burpsuite" for all URLs
Use proxy "Default" for all URLs
Completely disable FoxyProxy
KALI DOCUMENTATION
Options Ctrl+F2
Use Advanced Menus

To use apt-get or pip disable proxy!

// This might not be necessary when proxy root certificate is added to Kali "trusted list"

Possible to point script to proxy directly.

Figure A.2: Setup guide for Web Team logging tool.

A.5 Deatailed timeline of Web Team campaign

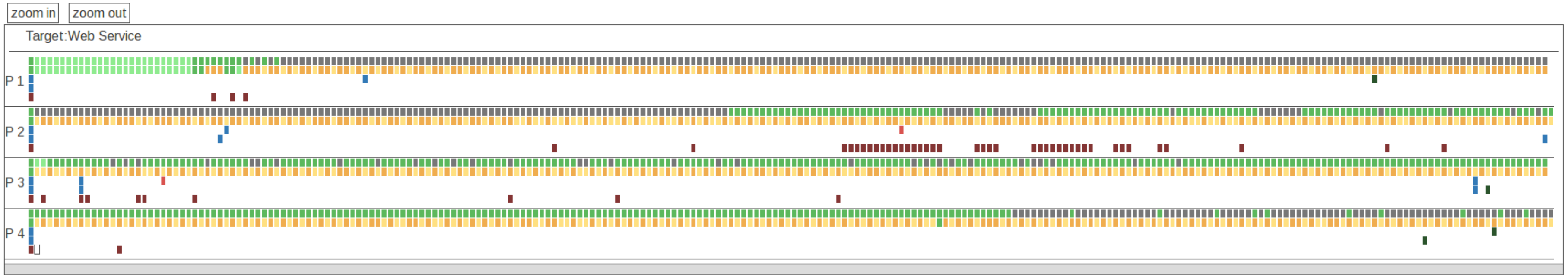


Figure A.3: Detail timeline of Web Team campaign for Blue Team A.

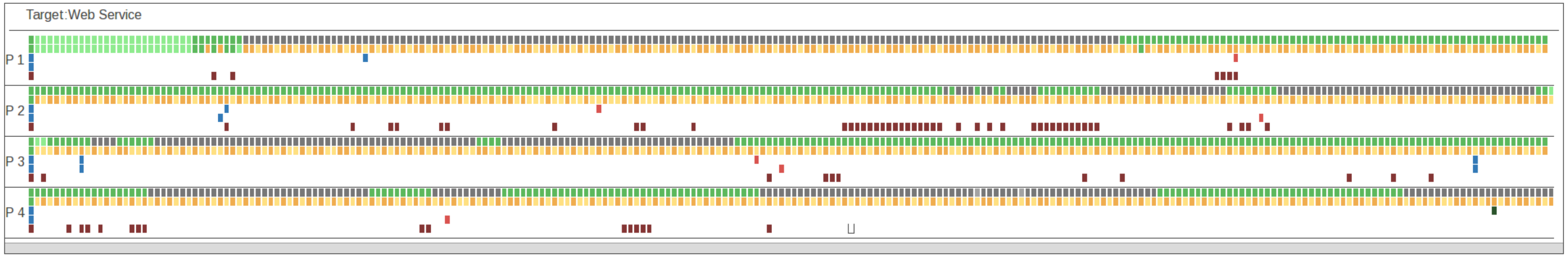


Figure A.4: Detail timeline of Web Team campaign for Blue Team B.

A.6 Attack methods

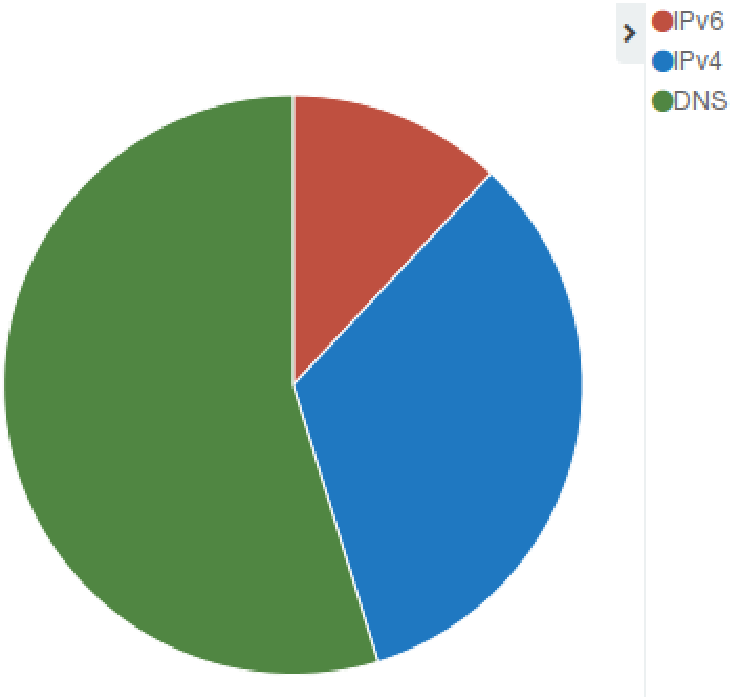


Figure A.5: IP protocol analysis.

A.7 Locked Shields 2013 Exercise Environment

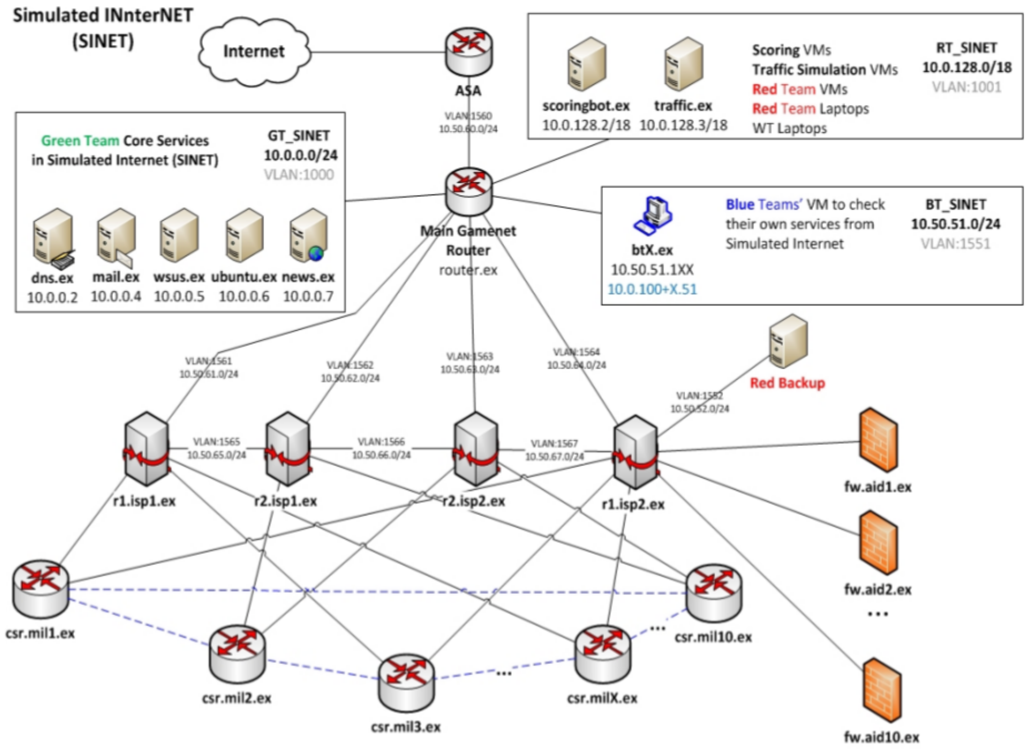


Figure A.6: Locked Shields 2013 exercise environment [10].