

AGENT-BASED BDI MODEL FOR AUTOMATED PENETRATION TESTING

A Project Proposal Report Submitted in partial fulfilment
of the requirement for the award of degree of

**MSC IN COMPUTED NETWORK AND
CYBER SECURITY**

BY

FATEMA NAMAZI 2119925



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

School of Applied Computing

**UNIVERSITY OF WALES TRINITY SAINT
DAVID (UWTSD)**

2022-2023

INDEX

CONTENT	PAGE NO.
List of Abbreviations	3
List of Figures & Tables	4-5
Abstract	6
Chapter 1 - Introduction	8
1.1 Motivation 1.2 Problem Statement 1.3 Objective of the Project 1.4 Project Overview 1.5 Research Question	11
Chapter 2 – Literature Review	15

2.1 Literature review of previous work 2.2 Penetration Testing 2.3 BDI Model	16
Chapter 3 - Methods	25
3.1 Information Gathering Attack 3.2 Configuration Attack 3.3 Buffer Overflow Attack 3.4 Password Attack 3.5 Web Attack 3.6 Sniffer Attack 3.7 Social Engineering Attack 3.8 Denial of Service Attack	26
Chapter 4 - Result	
4.1 Experimental Result 4.2 Report Generation 4.3 Experiment 4.4 Target Agent 4.5 BDI Agent 4.6 Reasoning Process between BDI Agent and Target Agent	31
Chapter 5 – Evaluation	40
References	44
Annexure 1.1 Gantt Chart	50

LIST OF ABBREVIATIONS

PC	Personal Computer
ICD	In-Circuit Debugger
IoT	Internet Of Things
IoE	Internet Of Everything
M2M	Machine to Machine

LIST OF FIGURES

Figure No.	Description	Page No.
2.1	A Representation of Penetration Testing	18
2.2	Block Diagram of the BDI agent reasoning cycle	20
3.1	Ontology for Penetration Testing	26
4.1	A part of Jason code for Penetration Testing	32
4.2	The interaction between BDI agent and Target	33
4.3	BDI Agent result in simulation 1	35
4.4	BDI Agent Belief set in simulation 1	36
4.5	BDI Agent result in simulation 2	37

LIST OF TABLES

Figure No.	Description	Page No.
1.1	OWASP 2013	9
4.1	Target Information	33
2	Gantt Chart	51

ABSTRACT

Penetration testing is the most frequently used techniques, it's a comprehensive guide to secure an organizations IT network by simulating severe hacking attempts. It identifies and test vulnerabilities by forensics and anti-forensics, information gathering and exploitation. Penetration Testing is assessed by several attacks i.e., password and wireless attacks, web applications and shells, surface-level vulnerabilities, and reverse engineering. The automation can remarkably improve the efficiency, availability and lower the cost of penetration testing. Existing approaches to the automation include those which map vulnerability scanner results to the corresponding exploit tools, and those addressing the pen testing as a planning problem, expressed in terms of attack graphs. In this project proposal, we propose an automated penetration testing approach based on the belief-desire-intention (BDI) agent model. Ontology is a widely used knowledge representation formalism in artificial intelligence area in recent years. In this project proposal, we propose an ontology based automated penetration testing approach. We use protégé to create the ontology, which is constructed based on attack taxonomy. It is central in the research on agent-based processing in that it deals interactively with dynamic, uncertain and complex environment. Penetration testing actions are defined as a series of BDI plans and the BDI reasoning cycle is used to represent the penetration testing process. This model is extensible and new plans can be added, once they have been elicited from the human experts. We report on the results of testing of proof of concepts BDI-based penetration testing tools in the simulated environment.

CHAPTER 1

1.INTRODUCTION

In the past few years, organisations have been advancing functionality in the current technologies, and refurbishing new applications to provide more safety against the cyber-attacks to provide better service for customers and/or employees[1]. Spending millions of dollars to recover from a security violation due to notification costs, remediation efforts, decreased productivity and lost revenue, the CSI study evaluates recovery efforts alone to be \$167,713.00 per incident [2]. There has been increase in the percentage of threats and attacks against IoT devices and services [3] even when they are safeguarded with encryption and authentication, every component in the system is exposed both to wireless attacks from inside the 6LoWPAN network and from the Internet [4]. The 2011 attacks against RSA (the security division of the EMC) are an example of how easy it can be to penetrate even the most security conscious organization[5]. The Internet of Things (IoT) network associate resource-constricted things to the information superhighway (World Wide Web) through the internet protocol version six (IPv6) and internet protocol version six Low-power Wireless Personal Area Network (6LoWPAN) [6]. To solve the problems related to security and act in accordance with the authorized cyber security regulations, security experts have designed and developed numerous IT security guaranteed methods including proof of correctness, layered design, software engineering environments and penetration testing [7]. Penetration testing is an extensive method to test the complete, integrated, functional, and trusted computing base that consists of hardware, software and people [8].

Moreover, it is imperative to competently protect an organization's information and assets by following a comprehensive, and structured approach in order to keep an organization secure from attacks [9].

In this project we will conduct IoT penetration testing to identify and test vulnerabilities of every component of the Internet of Things (IoT) system using software and hardware pen testing techniques. Most organizations also include OWASP Top 10 vulnerabilities (OWASP 2013) for IoT in their testing scope [10]. Table 1 (OWASP 2013) reveals the root causes of security risks in web application [10]. It is evident from Table 1 that web application security is significantly influenced by its surrounding [10].

Vulnerability	Root Cause
A1-Injection	SQL query manipulation, system call execution, lack of client code validation
A2-Broken Authentication	Unsecure transport layer, password handling and management, session ID management
A3-XSS	Lack of server side script validation
A4-Insecure DO References	Object exposure to unauthorized clients
A5-Security Misconfiguration	Versioning, system hardening, access control, verbose messaging (error, debug)
A6-Sensitive Data Exposure	Data format, encryption key compromise, unsecure data transport
A7-Missing Functional Level Control	Access control to sensitive request handlers
A8-CSRF	Caused due by session cookie stealing
A9-Using components with known Vulnerabilities	Exploitation of vulnerable framework components/libraries from 3 rd party
A10-Unvalidated Redirects and Forward	Attacker redirects client to his site with the home page look and steals the login information

Table 1.1: - OWASP 2013

The main objective of IoT penetration testing is to identify security vulnerabilities under measured conditions so they can be abolished before unauthorized users exploit them. Cyber security experts use penetration testing to address challenges deep-rooted in vulnerability assessment, focusing on high-severity susceptibilities [7].

1.1 Motivation

Real motivation of this work is to develop an efficient system, expand the security of the computing assets being tested and to protect the connected devices and network from all the security risks with the help of penetration testing.

In this project, we will explore several types of penetration tests and tools commonly used in safeguarding an IoT device. The results produced after the penetration test will deliver affirmation and confidence to the users and business owners that their connected devices and networks are secure and the level of risk to the connected device system is within acceptable levels. Penetration testing framework used for the testing was based on the National Institute of Standards and Technology (NIST)[11]. There are four phases involved in the framework, i.e.: - Planning Phase, Discovery Phase, Attack Phase and Reporting Phase [12].

1.2 Problem Statement

Assessing the security of services in IoT system can sometimes be very demanding, because these systems often use newer protocols assisted by very few security tools, it's important that we learn which tools we can use and whether

we can expand those tools proficiencies [13]. Demonstrating the automation of application, end to end security testing, providing static and dynamic analysis for both iOS and Android using Mobile Security Framework (MobSF) , setting up a Burp Suite (a set of tools used for pen testing) using it for web app testing , identifying UART and JTAG pinouts, solder headers and hardware debugging , getting solutions to common wireless protocols exploring the mobile security and firmware best practices [14].

1.3 Objectives of The Project

- 1.Reconnaissance: The information is gathered regarding the target system, software and users for using it later to attack the target system [15].
- 2.Scanning: In this phase, there is an intensive use of technical tools to further the attacker's knowledge about the target system [15].
- 3.Gaining Access: In this phase, the attacker proactively tries to exploit the systems by making use of the information gathered in the previous phases [15].
- 4.Maintaining Access: It is required for the attacker to constantly be able to access the target system to gather as much data as possible [15].
- 5.Covering Tracks: An attacker is compelled to erase any traces, such as log files, in order to remain unidentified [15].

IoT-Penetration testing uses the national vulnerability database (NVD), which is government archive of standards-based vulnerability administration data characterized using the security content automation protocol (SCAP). It's updated regularly to recognize the possible vulnerabilities on a particular version of the software [16]. NVD examined information about vulnerabilities of different domains i.e., software and hardware.

The vulnerabilities were recorded to Common Vulnerability Enumeration (CVE) [17] Ids (CVE-ID) that assists to lay out an acquainted name for openly accessible vulnerabilities [16]. Each announced CVE is examined by the NVD team using the Common Vulnerability Scoring System (CVSS) structure [16]. Each CVE is associated with a Common Platform Enumeration (CPE) that designates the vulnerabilities [16]. CPE is an organized naming scheme normally used for information technology systems, packages and software [18].

A systematized and all-inclusive directory Common Weakness Enumeration (CWEs), which list the defects found in software, has been created by MITRE [16]. The main objective of CWE is to recognize the defects in software and for advancing automated tools for securing and anticipation of the lacunae [16]. The details provided by NVD and CWE plays an essential part in comprehending trends and patterns in software and hardware vulnerabilities [16].

1.4 Project Overview

An IoT system comprises of sensors, actuators, bridges, gate, routers, etc [16]. Every component is a complete

system. Thereby, we shall consider an IoT network equivalent to an SoS [16]. Penetration testing is a broadly used technical proposition which assesses the conventional internet or systems security through substitution of a real attack [19]. In consonance with PTES (Penetration Testing Execution Standards) [20], the procedure of penetration testing composes of Pre- Engagement Interaction, Information Gathering, Threat Modelling, Vulnerability Analysis, Exploitation, Post Exploitation and Reporting [21].

1.5 Research Questions

1. What artificial intelligence (AI) proposals can be used to communicate with interactive, dynamic, uncertain and complex real-world penetration testing designs [22] ?
2. How can penetration testing problems be modelled with these methods [22] ?
3. How can an ontology be used to improve reasoning ability and enable knowledge reusability within penetration testing scenarios [22] ?
4. How can automated penetration testing be carried out in the IoT environment [22] ?

CHAPTER 2

2. LITERATURE REVIEW

2.1 LITERATURE REVIEW OF PREVIOUS WORK

In this part, we briefly mention the associated work on security and privacy of IoT pen-testing. According to our preceding research workers the amount of IoT devices is increasing very swiftly [23], [24]. The gaining increment in the number led the internet-connected objects to exceed the world's population in 2008, while reaching 50 billion in 2020 [23]. According to a research report let by the LHS Markit it was reported that 108 million network security cameras were dispatched all over the globe [24].

Xue Qui et al. [25] suggested a mechanized method of penetration testing named AEPT (Automata Model of Penetration Testing) based on multiple phase model of penetration testing, i.e., testing time, target time, scheme, plan, the assortment of examining information in addition to the evaluation of fraudulent vulnerabilities. This model was then afterwards used to develop the automated scheme of penetration testing [26]. Therefore, they suggested the automated administrating method of penetration testing scheme by demanding the exploitation module [26]. Cynthia and Swiler [27] introduced a graph-based pliable approach to present system vulnerability analyses. This scrutiny system broken database of regular attack into nuclear steps, particular network configuration, topology data and attacker profile [26]. They also directed apprehensions about how and why new malware targets IoT

data directly [28] . Their research paper adopted Nmap and Wireshark, as two procedures for gathering data and assessing protection modules [28]. Nmap was enforced to detect the internet protocol address of the camera and Wireshark was for examining network-level packets [28]. Therefore, it outlined the welfare of conserving IoT through certain ideas on how to upgrade the protection of IoT systems and contributing basic safety measures to reduce exploitation chances [28]. Establishing a set of goals and plans plays a vital role in gaining success in penetration testing [21]. The BDI agent is an optimal model to attain automation because of its potential to communicate with the target via perception and action during penetration testing [21]. BDI model can be used for penetration testing of IoT devices.

AEPT attempts to utilize all susceptibilities of the target after going through the scanning report as an input and fails to consider the changing and undetermined nature of a condition in a real-life penetration testing scenario [26].

Moreover, a cyber security penetration tester mostly attempts to understand a target through a series of multiple attacks by using exploitation tools, specifically to regain from failed attempts [26] . AEPT is incapable of dealing with such attack chain situations [26].

A great number of approaches to time the address mechanical penetration testing as a planning problem for an attack graph [26] .

There are innumerable implementations of agent-based models in the real world such as in agriculture, air traffic control, economics, emergency evacuation, healthcare and social behaviour [26].

2.2 PENETRATION TESTING

A penetration testing is the accredited, organized and structured procedure of using known vulnerabilities to accomplish an incursion into host, network or application resources [29]. The penetration test can be managed on internal (a guiding approach or host security system) or external (the organization connection to the internet) measures [29]. It usually comprises of using an automated or manual toolset to test the organisation assets [29]. If vulnerability is applied by an uncertified particular to access organisations assets, companies' resources which can be compromised [29].

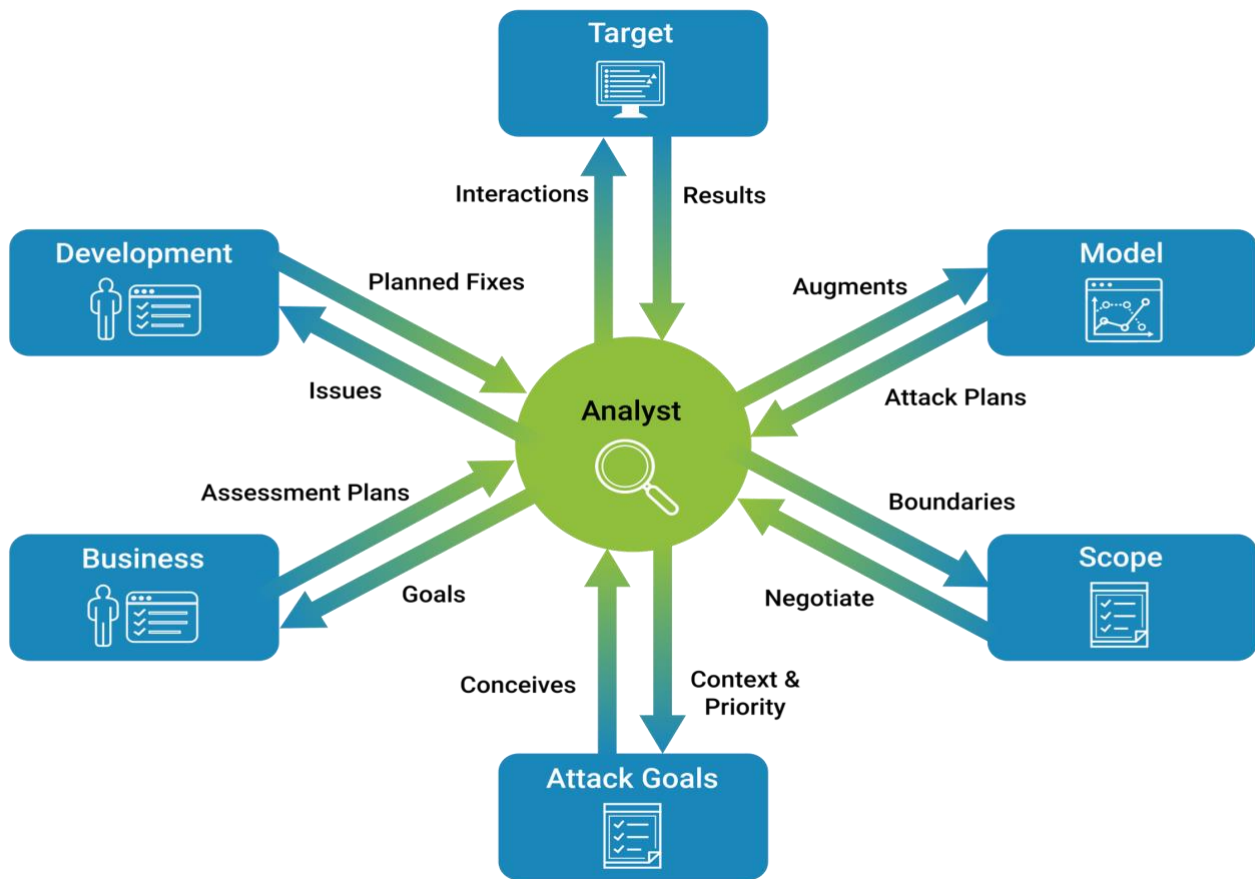


Figure 2.1 A representation of Penetration Testing

2.3 BDI MODEL

The BDI model reports the procedure of how a representative can select measures in relation to the object information during penetration testing [21]. The BDI is described by three logic elements: belief, desire and intention [21]. Our model act in accordance with the agreements adapted in the Jason Interpreter, a model based on Procedural Reasoning System (PRS) [30].

BDI agent is explained as a tuple (Ag, B, D, I, P, A, S), where : Ag is the name of an agent, B is belief set, which constitutes the details about the object and it will be amended after

implementing actions; D is desire set , which constitutes all the options or possible applicant plans of penetration testing for the agent ; I is an Intention set, which constitutes the agents goals or which scheme the agent decides to carry out ; P is a plan set , which constitutes of available plans, each giving data about how to reach the goals ; A is an action set, which build the actions the agent can perform ; S is a perception set , which gathered the information from the surrounding [21].

The reasoning cycle of BDI agent is shown in Figure 2 as below:

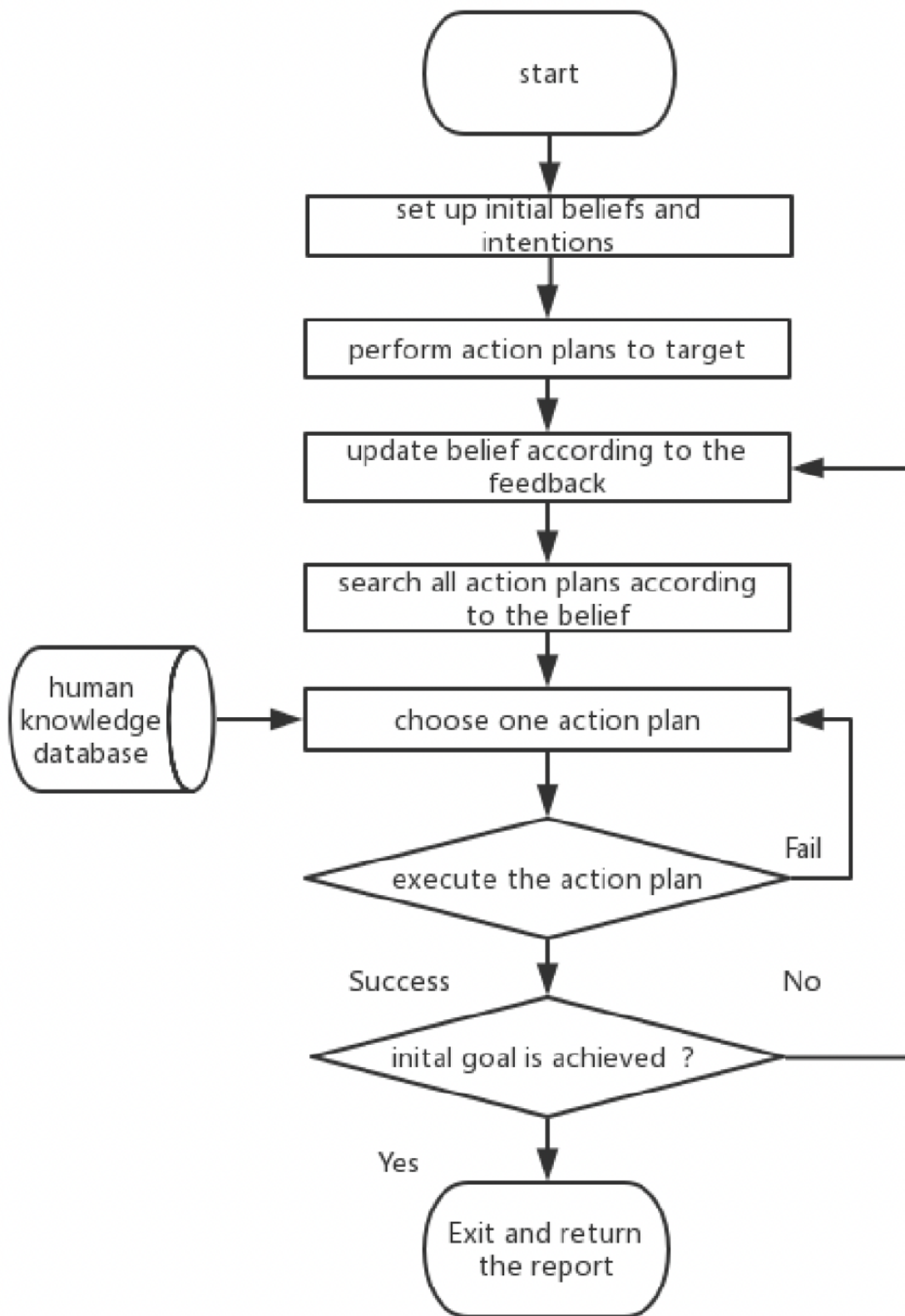


Figure 2.2 Block diagram of the BDI agent reasoning cycle

The BDI agent reasoning cycle for penetration testing is described below: -

1. Initial dependence and purpose will be built by the penetration tester and usually constitutes information concerning the target such as the domain or IP address and the authority which the penetration testing must attain considerably [26].
2. The BDI agent considers the target data by conducting several information gathering measures [26]. For example, Nmap can assemble OS type and ports opened at the charge [26] .
3. After recognizing the result, ongoing data will be updated [26]. At this time, the BDI agent should clasp the ongoing data about the attack [26].
4. According to the recent credence all the applicable action schemes will be found [26]. For example, if port 80 of the attack is revealed, then password attack, buffer overflow attack, SQL injection attack will all be considered options for the penetration testing researcher [26].

5. The BDI agent selects one plan from the individual action plans to enhance the intention and stand by execution as reported by the context of the plan and the human intelligence database which selects the plan subjected to human penetration testing circumstances in real world [26]. We predetermine the importance of the given actions in human intelligence database [26] .
6. The BDI agent implement the given plan [26]. If the plan crashes, then the agent selects another plan [26].
7. The BDI agent examines whether the inception goal is attained or not and pronounce either to output the report which records the process of the whole penetration testing or to return to the new reasoning cycle [26].

In this report , we put forward an agent based BDI model with the point to upgrade the efficiency and possibility of success for automated penetration testing [31].Penetration testing measures are explained as a series of BDI plans and the BDI reasoning cycle is used to describe the penetration testing procedure [31].To authenticate this model, we execute a prototype system and resembled simulated real life world penetration testing scenarios using agent-based programming language Jason [32]. Our model is executed in Agent Speak Jason which is a multi-agent system

programming language based on belief-desire-intention standard (BDI) [31]. We user-define multiple actions to protect whole the penetration testing steps from data collection to report [31]. The pretence experiment comprises of two agents to describe the BDI agent and the target [31]. We use the inward communication measures in Jason to imitate the interactivity between the BDI model and the target agent [31].

In [33] suggested a penetration test automation method based on BDI agent model; BDI (Belief – Desire - Intention) agent could be a typical design in astutely professional programming field. Many PRS (Procedure Reasoning System) were designed based on BDI agent programming including ASL (Agent Speak Language) [34]. The ontology based BDI agent can contribute of penetration experts understanding, which is very beneficial, moreover, this understanding can be regenerated for its immaterial to specific systems under penetration testing [34].The perspective is based on Ontology – based BDI agent and Reinforce Learning based Q-learning method [34].The agent of reinforcement learning has good educating and designing potential, but absence in the reasoning potential of the BDI agent, Q-learning can guide BDI model [34].

Chapter 3

3. METHODS

In this chapter the block diagram and design aspect of independent modules are considered. Block diagram is shown in Fig 3.1.



Fig 3.1 Ontology for Penetration Testing

The ontology is the understanding base in an official design that can be collaborate between intelligent AI and human [35]. As shown in figure 3.1, our suggested ontology based BDI agent with augmentation learning planning, is started with a penetration test ontology model design [34]. We set up our ontology model based on specific penetration test domain knowledge [34].

By penetration testing in use [36], the attacks comprise of the following categories: - information gathering attack, configuration attack, buffer overflow attack, password attack, web attack, sniffer attack, social engineering attack, and denial of service attack.

3.1 Information Gathering Attack

Information gathering is the most condemning stage in penetration testing [33]. Moreover, we need to accumulate data about the target including IP address, opened port, application information, OS information, human or organisation information, network information, defence mechanism, configuration information, vulnerability information physical environment information and so on [33].

3.2 Configuration Attack

Such charges are normally based on a controller's inappropriate configuration of the system [33]. For example, the robot.txt file generally reveal the structure data of the website, or the inventory that allows users to transfer files has terminatable permission so that assailant can upload a hostile file and execute it [33].

3.3 Buffer Overflow Attack

A buffer overflow is a representative software coding inaccuracy that an attacker could utilize to gain access to the target system [33]. When a scheme, while writing

information to a buffer, invades the buffers borderline and initialise adjoining memory locations [33] .

3.4 Password Attack

Password attack is a crucial part of penetration testing [33]. Mostly, an attacker can obtain some certain authorization from the target system if a passcode attack is successful [33]. The most certain passed down password attack is centred on a dictionary, which compose of all practicable passwords [33].

3.5 Web Attack

WEB attack is a charge against web applications [33]. The most typical attacks incorporate injection, cross-site scripting (XSS) and cross-site request forgery (CSRF) [33].

3.6 Sniffer Attack

If there is no vulnerability or the object system is safeguarded, a human encountered penetration tester would considerably attempt to perform a sniffer attack [33]. Moreover, they burgle other systems which are beneath similar sub-network as the original target, and the researcher can observe and examine all network flow to gain hypersensitive data such as password [33].

3.7 Social Engineer Attack

Social engineer attacks are charges against humans, including administrators who have weak security

perception [33]. Social engineering is the phrase used for a wide range of malevolent movements obtained via human communications [33]. In distant penetration testing, typically present spear-phishing attack by email or link, website forge attack – spoofing attack [33].

3.8 Denial of Service Attack

A denial-of-service (DoS) is a charge where the hackers strive to counter authorized users from gaining access to the services [33]. In a DoS attack, the hacker commonly dispatch's immoderate information flow to the network or server to drain target resources [33]. In penetration testing, a DoS attack is not common and considerably tends to reinstall the target system for some motive [33]. These types of attacks includes SYN flood, TCP/UDP attack, SMTP attack, ICMP attack, etc [33]. If the charge target comes from a non-identical device, we name it a distributed denial-of-service attack (DDoS attack) [33].

In BDI agent, we build the value of exemption as none primitively and the original target is root privilege [31]. We prearrange information gathering organize to probe OS type, opened port, service and vulnerability information from the target agent [31].

CHAPTER 4

4. RESULT

4.1 Experimental Results

The project “Agent-Based BDI model for Automated Penetration Testing” is designed to improve the adaptability and contingency of success for automated penetration testing using agent-based programming language Jason.

4.2 Report Generation

Our model would document the charge action name and track to reveal the procedure of the whole penetration testing and output it as described if a plan is accomplished successfully [26]. We will use the central action of Jason to attain this purpose [26].

Figure 4.1 shows a representation of the Jason code based on the BDI model for penetration testing [26]. Firstly, we structure the commencing belief and inceptive goal of our BDI agent as IP address and root privilege [26]. Then we user define the basic data gathering plan to investigate the target opened port, OS type and application services data by Nmap [26].Following the basic data gathering stage, we execute the vulnerability data gathering by OpenVAS and execute the buffer overflow attack using Metasploit or an SQL injection attack by SQL map if the prerequisite is composed [26].Each of these tasks will be reported by Jason’s internal action named print function [26].

```

1  ip_address(192.168.0.10). // initial belief
2  !privilege(root). //initial goal
3
4  // plans
5  @basic information gathering
6  +! get(port) : true
7  <- !Nmap(ip_address).
8  .print(Namp ip_address).
9
10 +! get(os) : true
11 <- !Nmap(ip_address);
12 .print(Namp os).
13
14 +!get(service) : true
15 <- !Nmap(ip_address);
16 .print(Namp service).
17
18 @vulnerability information gathering
19 + get(vulnerability) : true
20 <- !openvas(ip_address);
21 .print(openvas ip_address).
22
23 @ Buffer overflow attack
24 +get(vulnerability)
25 : get(vulnerability) != null
26 <- !Metasploit(cve no.);
27 .print(metasploit ip_address).
28
29 @Sql injection attack
30 +get(port == 80)
31 : get(service == apache or nginx)
32 <- !SQLmap(ip_address);
33 .print(SQLmap ip_address).

```

Fig 4.1 A part of Jason code for Penetration Testing

4.3 Experiment

Our model would run on a PC with an intel i7 CPU at 2.0 GHz and 4GB of RAM. As we can see in figure 4.2, the virtualized experiment comprises of two agents to display

the BDI agent and the target [26] . Consequently, to clarify the procedure of penetration testing in the simulated environment, we use the central transfer actions in Jason to imitate the interconnection between the BDI model and the target agent [26].

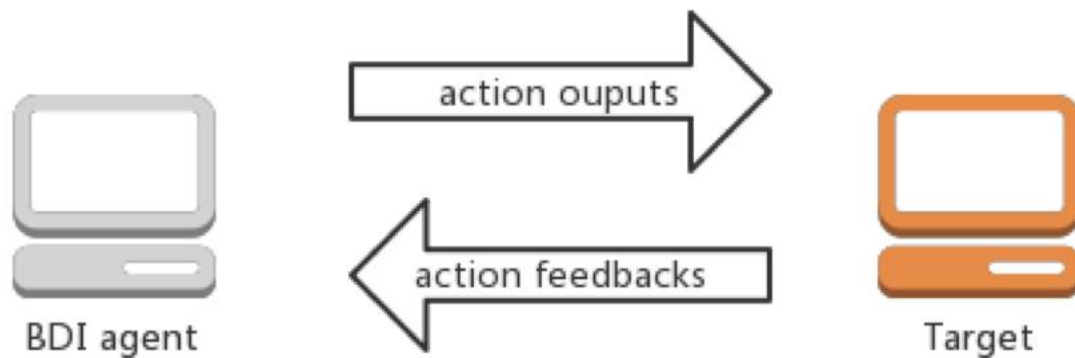


Fig 4.2 The interaction between BDI agent and Target

4.4 Target Agent

OS	Port	Services	Vulnerability	Password
Linux	80, 22, 3306	Nginx, SSH, MySQL	CVE-remote CVE-local	SSH:456

Table 4.1. Target Information

We have to establish fundamental information concerning the object comprising the system type , opened port, service , vulnerability and the SSH password in the commencing belief set to imitate a target server as shown in Table 1 [26].To make the framework unresolved, we use

arrangement and set 0.8 as the threshold to regulate if the SSH password attack is accomplished by achieving a random number and correlating it with the threshold [26]. Concerning the remote or local buffer inundation attack achieving rate, we set thresholds as 0.5 and 0.3, respectively [26].

4.5 BDI Agent

In BDI agent, we establish the worth of advantage as none originally and the initial goal is root privilege [26]. We predetermine data gathering plans to inquire OS type, opened port, service and vulnerability data from the target agent [26]. To clarify the procedure of penetration testing by the BDI agent, we predetermine the password attack and buffer overflow attack to target [26].

4.6 Reasoning Process between BDI agent and Target agent

We conduct two simulations to prove how our BDI agent can accomplish in multiple conditions:

1. Simulation 1

We can obtain from the output of the operation of the BDI agent in figure 4.3, the BDI agent examined all data about the target in the belief set but was unsuccessful to execute the password attack because the rate of the password attack has not compassed the specified 0.8 threshold [26]. Therefore, the BDI agent would not enact the buffer

overflow attack as well because we determine the prerequisites of it as successful password attack [26].

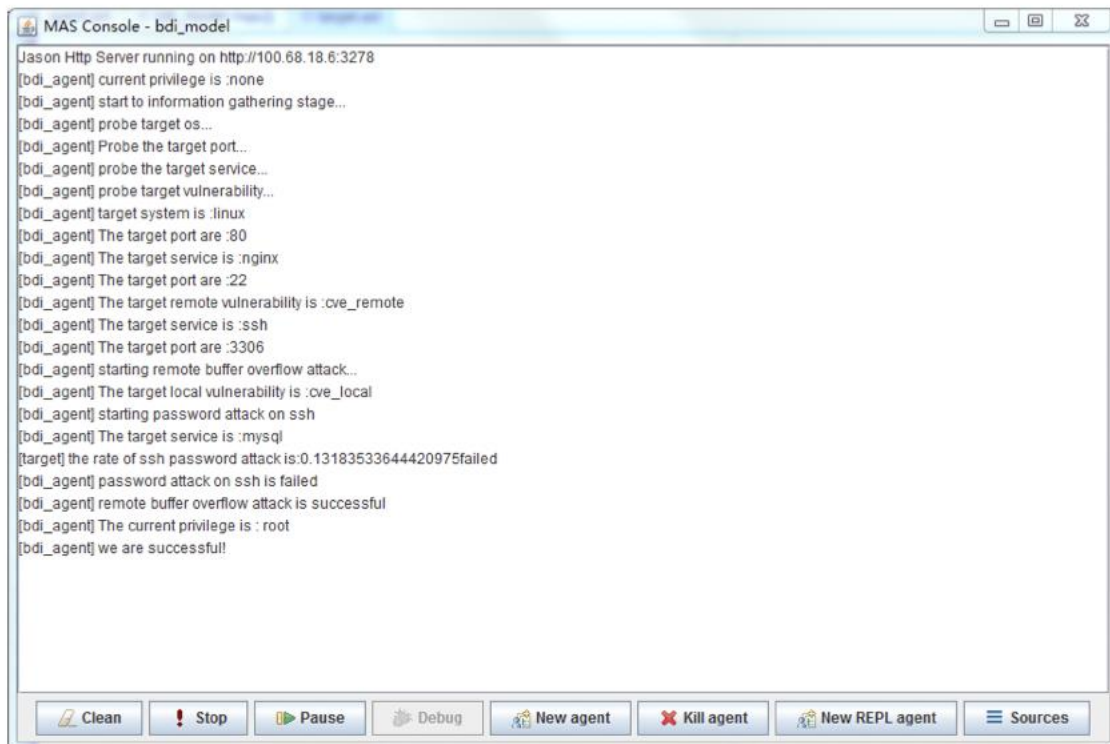


Figure 4.3 BDI agent result in simulation 1

Moreover, the isolated buffer overflow attack was successful, and the current concession has changed to root [26]. We can verify the rationality of the procedure in figure 4.4.

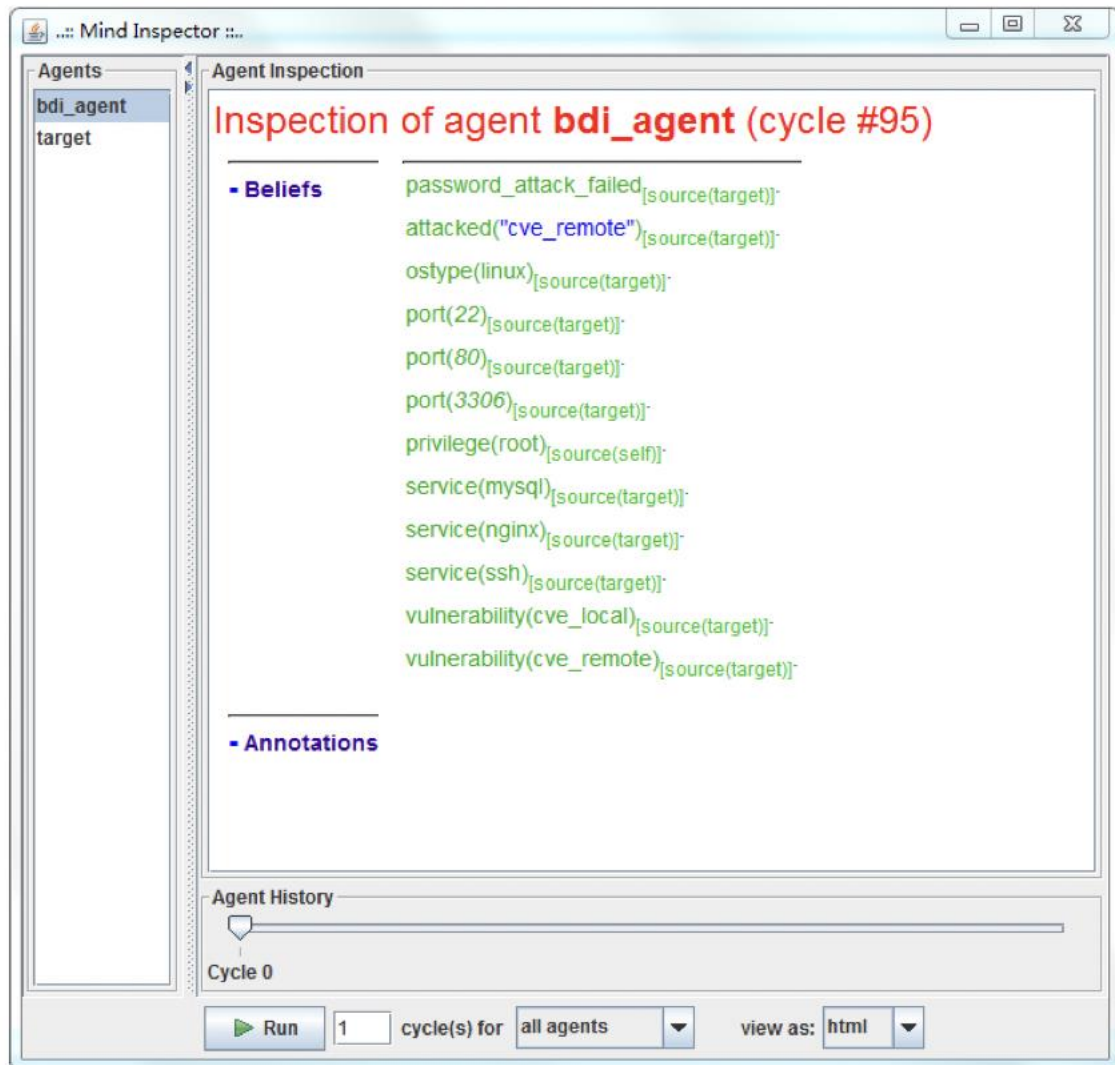


Figure 4.4 BDI agent Belief set in simulation 1

2. Simulation 2

this simulation, the BDI agent examined all the data of the charges and successfully disrupted the SSH password since the rate of the SSH password attack was set to 0.9 which is greater than the 0.8 threshold [26]. Thereby, the BDI agent carried out successfully in both the local and the isolated buffer overflow attacks [26] . We can see the allowance change from none to the user then reduced to root in figure 4.5.

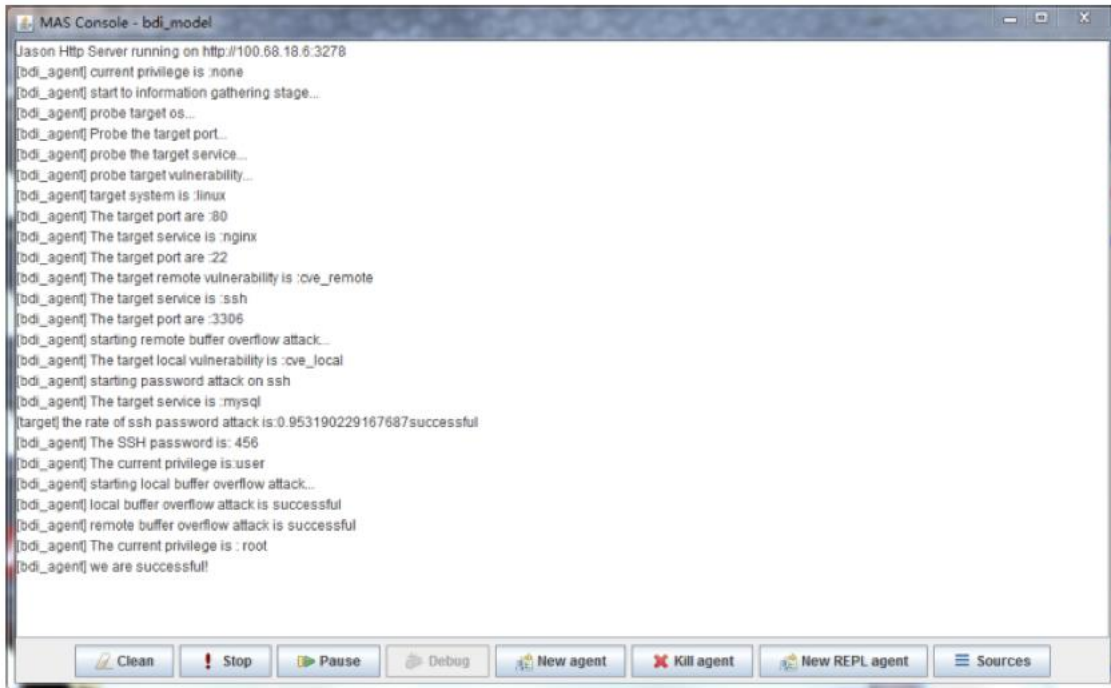


Figure 4.5 BDI agent result in simulation 2

The result of a successful penetration testing simulation is the recognition of vulnerabilities, in which characteristics will be handled and finally be reported to the owner of the object with data to improve future security [21].

The NIST SP-800-155's special publication [37] outline several securities testings calculates and supports guidelines for organizations on arranging and managing several kinds of security assessments [38]. Amidst all, it explains a penetration testing methodology model based on the following four stages, repeated iteratively: Planning, Discovery, Attack, Reporting [38]. Security aim and penetration testing objective are discovered during the planning phase [38]. The discovery phase contains information gathering, scanning and vulnerability analysis [38]. The attack phase is the main activity in which the previously recognized potential vulnerabilities are established and validated [38]. The reporting phase takes

place at the same time with the other three phases of the penetration test [38]. The penetration testing procedure we adapted supports IoT-based systems and authorize personnel with restricted computer security skills to recognize and signify working attacks [38].

The BDI agent based programming states penetration planning better in terms of intelligence and flexibility [34]. With the purpose of penetration goal [34]. It can make designs according to its assumption which came from the ontology knowledge base and the reinforcement learning process [34]. With an quintessential algorithm, the BDI agent can make the most systematic plan [34]. The construction of this research paper can be easily elongated [34]. More network scan tools, and exploit tool systems can add to this structure, such as Immunity Canvas, etc [34].

This project proposal showcases an agent-based belief-Desire-Intension (BDI) modelling for the automation of penetration testing, which authorize interactivity between dynamic and uncertain targets [31]. Penetration testing measures are described as a series of BDI plans while the BDI reasoning cycle is acclimated to represent the penetration testing process [31]. The substitute shows the BDI agent behaviour and reasoning procedure to authenticate the modelling [31].

Penetration testing is a compendious procedure to recognize the vulnerabilities in a network [7]. There are three types of penetration testing i.e., network, application and social engineering [7]. It provides benefits such as avoidance of financial loss; concurrence to industry regulators, customers and shareholders; conserving corporate image; dynamic removal of identifiable risks [7].

CHAPTER 5

5. EVALUATION

In this project proposal, we proposed an agent based BDI model for automated penetration testing approach. As stated by the proposed penetration testing attack taxonomy, we developed an ontology by protégé. In order to make conclusions and deliver new knowledge, we put together attacks and methodology steps to generate a penetration testing knowledge base as well as to attain logical functions. Moreover, we make use of agent based (BDI) structure and target agent to approach the procedure of automated penetration testing. To authenticate our proposal, we use the internal action of Jason to implement an attack framework in a computer-generated environment. The ontology and BDI agent are flexible, while we are going to find ways to expand it impulsively soon [35].

Therefore, the dissimilarity is that attack details, and penetration testing understanding is reserved in the ontology [35]. In our perspective, attack details are reserved as occurrence under attack classes with information characteristics such as IP address, port, OS, application, configuration, vulnerabilities and current permission [35]. The target characteristic is same subnet constituting the connection between objects [35]. From the prospects of hackers, target objectives are saved as occurrences under the hacker techniques class with data characteristics such as actions, precondition and post-condition [35].

The purpose of the SWRL rules is to determine the target scope and their weaknesses according to target data, in the interim, upgrading the information or association in the ontology [35]. After the ontology regulates a specific target action, the BDI plan will carry out the attack [35].

At this moment, the belief set in the BDI agent is accustomed to hoard new details achieved during the attack [35]. Therefore, the ontology will be upgraded, comprising the target characteristic exploit-By and the information attribute, present characteristic after the attack is finished in the BDI agent by Owlready2 [35]. The procedure of our ontology-based automation of penetration testing is introduced [35].

This research proposal narrates a three-stage methodology comprising of test preparation, test, and test analysis phase [39]. The test phase is accomplished in three stages: information gathering, vulnerability analysis and vulnerability exploit [39]. This stage can be accomplished manually or by utilizing automated tools [39]. This penetration testing procedure was demonstrated on the web application, BOG, and the Tune store [39].

The penetration testers need to follow a general design to present the test result [39]. One of the greatly used principal roles of the test analysis phase is the construction of decontamination which is inclusive of all required restorative proceedings for the identified vulnerabilities [39].

Intrusion detection on the internet of things network is discovered to counter the increasing number of impingements on most of the websites and networks [6].

Over the coming years, safeguarding the network security in internet of things has turned out to be very strenuous because there has been a substantial increase in the number of networks being connected and the considerably high sophisticated nature of attack technologies being used [6]. This project proposal showcases an outline of intrusion detection in internet of things as well as detail

understanding of several threats, vulnerabilities, attacks and available methods of observing an intruder in our internet of things network [6]. Agents execute parallel threads of user defined tactical programming language (TPL) that controls their behaviour.

The proposed proposal passes the three phases of penetration testing 1. Pre-engagement, which defines its scope, documentation, and rules. 2. Engagement, where in application and network layers along with segmentation is evaluated. 3. post-engagement, finds and insights which are applicable and useful.

Penetration testing is a perspective that authenticate the web application security at the conclusion and only from a attacker's viewpoint [10].An attacker is not anxious because of time, but a software release is [10].Moreover, software defects, detection and removal of vulnerabilities in the SDCL is costly, and a basic approach throughout the SDLC will capitulate a more secure application with a high level of reliance [10]. The use of automation tools along with the human interactions can prove to be both successful and accurate [10].

REFERENCES

- [1] B. Fraser, *Site Security Handbook*, .
- [2] Anonymous "Benefits of Penetration Testing ," *iVolution Security Technologies*, 2011. Available: http://www.ivolutionsecurity.com/pen_testing/benefits.php.
- [3] Mohamed Abomhara and Geir M.Koien, "Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks" , " *Department of Information and Communication Technology, University of Agder, Norway*, 2015.
- [4] ↑ Shahid Raza a, Linus Wallgren a and b. Thiemo Voigt a, "SVELTE: Real-time intrusion detection in the Internet of Things," *Department of Information Technology, Uppsala University, Sweden*, pp. 2662-2674, 2013. Available: <http://www2.cs.uh.edu/~acl/cs6397/Presentation/2013-Elsevier-SVELTE%20realtime%20ID%20in%20the%20IoT.pdf>.
- [5] Kevin Henry, *Penetration Testing: Protecting Networks and Systems*, Revised, .
- [6] O. Okpe, O. John and S. Emmanuel, "Intrusion Detection In Internet Of Things (IoT)," *International*

Journal of Advanced Research in Computer Science, vol. 9, (1), pp. 504-509, 2018.

[7] A. G. Bacudio *et al*, "An Overview of Penetration Testing," *International Journal of Network Security & its Applications*, vol. 3, (6), pp. 19-38, 2011.

[8] Gary McGraw, *Software Security: Building Security In*, .

[9] The Canadian Institute of Chartered Accountants, "Using an Ethical Hacking Technique to Assess Information Security Risk ," *Information Technology Advisory Committee*, 2003.

[10] 2017.

[11] Anonymous *Technical Guide to Information Security Testing and Assessment* , .

[12] Michael Kyei Kissi and P. Michael Asante, "Penetration Testing of IEEE 802.11 Encryption Protocols using Kali Linux Hacking Tools," *International Journal of Computer Applications* (0975 – 8887), vol. 176, (32), pp. 26-33, 2020. Available: www.ijcaonline.org.

[13] Fotios Chantzis *et al*, *Practical IoT Hacking*, .

[14] Aaron Guzman and Aditya Gupta, *IoT Penetration Testing Cookbook : Identify Vulnerabilities and Secure Your Smart Devices* , .

[15] Ralph Ankele *et al*, "Requirements and recommendations for IoT/IIoT models to automate security assurance through threat modelling, security analysis and penetration testing," in 2019, .

- [16] Geeta Yadav *et al*, "IoT-PEN : An E2E Penetration Testing Framework for IoT," *Information Processing Society of Japan*, vol. 28, pp. 633-642, 2020. Available: https://www.jstage.jst.go.jp/article/ipsjjip/28/0/28_633/pdf/-char/en. DOI: 10.2197/ipsjjip.28.633.
- [17] Anonymous "NIST: Common vulnerabilities and exposures," Available: <https://cve.mitre.org>.
- [18] Anonymous "NVD: Official common platform enumeration (CPE) dictionary," 2007. Available: <https://nvd.nist.gov/products/cpe>.
- [19] M. Denis, C. Zena and T. Hayajneh, "Penetration testing: Concepts, attack methods, and defense strategies," in *Systems, Applications and Technology Conference (LISAT)*, 2016, pp. 1-6.
- [20] Anonymous "IoT Attack Surface Areas - OWASP," Available: <https://www.owasp.org/index.php/IoT> Attack Surface Areas.
- [21] Ge Chu and Alexei Lisitsa, "Penetration testing for internet of things and its automation," in *IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th Intl. Conference on Data Science and Systems*, 2018, pp. 1479-1484.
- [22] Ge Chu, "Automation of Penetration Testing." , The University of Liverpool (United Kingdom), 2021.
- [23] M. Swan, "Sensor mania! the internet of things, wearable computing, objective metrics, and the quantified

self 2.0," *Journal of Sensor and Actuator Networks*, pp. 217-253, 2012.

[24] I. Markit, *The Top Trends of 2019: Powered by Transformative Technologies*, .

[25] Xue Qiu *et al*, "An automated method of penetration testing," (*IEEE Comput.Commun.IT Appl. Conf.*), pp. 211-216, 2014.

[26] Ge Chu and Alexei Lisitsa, "Agent-based (BDI) modeling for automation of penetration testing ," *Department of Computer Science, University of Liverpool*, pp. 1-6, 2019. Available: <http://arxiv.org>.

[27] C.Phillips and L.P.Swiler, "A Graph-based System for Network-Vulnerability Analysis," (*Work.New Secur. Paradig.*), pp. 71-79, 1998.

[28] Peshraw Ahmed Abdalla and Cihan Varol, "Testing IoT Security: The Case Study of an IP Camera," *Ieee*, 2020. Available: <https://ieeexplore-ieee-org.ezproxy.uwtsd.ac.uk/stamp/stamp.jsp?tp=&arnumber=9116392>.

[29] S. Angel and S. Sarala, "A study on Penetration Testing," *International Journal of Advanced Research in Computer Science*, vol. 2, (5), 2011.

[30] R. H. Bordini, J. F. Hu ¨bner and M. Wooldridge, "Programming multi-agent systems in Agent Speak using Jason," *John Wiley & Sons*, vol. 8, 2007.

[31] Ge Chu and Alexei Lisitsa, "Poster: Agent-based (BDI) modeling for automation of penetration testing," in 2018, .

- [32] R. H. Bordini, J. F. Hu ¨bner and M. Wooldridge, "Program- ming multi-agent systems in AgentSpeak using Jason," *John Wiley & Sons*, vol. 8, 2007.
- [33] G. Chu and A. Lisitsa, "Ontology-based automation of penetration testing." in *Icissp*, 2020, pp. 713–720.
- [34] Kexiang Qian *et al*, "Ontology and reinforcement learning based intelligent agent automatic penetration test," in *IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, 2021, pp. 556-561.
- [35] G. Jian-bo *et al*, "Ontology-Based Model of Network and Computer Attacks for Security Assessment," *Journal of Shanghai Jiaotong University(Science)*, vol. 18, (05), pp. 554–562, 2013. . DOI: 10/ghgs7m.
- [36] Broad *et al*, "Hacking with Kali : Practical Penetration testing techniques", *Newnes*, 2013.
- [37] A. C. Karen Scarfone, Murugiah Souppaya and A. Orebaugh, "Technical guide to information security testing and assessment," *NIST Special Publication 800-115*, 2008. Available:
<https://csrc.nist.gov/publications/detail/sp/800-115/final>.
- [38] Massimiliano Rak, Giovanni Salzillo and Claudia Romeo, "Systematic IoT Penetration Testing: Alexa Case Study*," *University of Campania Luigi Vanvitelli, Department of Engineering, Via Roma 29, Aversa (CE), Italy*, vol. 2597, (Creative Commons License Attribution

4.0 International (CC BY 4.0).), pp. 1-11, 2020. Available: <http://ceur-ws.org/Vol-2597/paper-17.pdf>.

[39] G. Kaur and G. Kaur, "Penetration Testing: Attacking Oneself to Enhance Security," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 5, (4), pp. 574-577, 2016.

ANNEXURE

Project Completion Status with respect to Gantt Chart

A Gantt chart, commonly used in project management, is one of the most popular and useful ways to showing activities (tasks or events) displayed against time. On the left of the chart is a list of the activities and along the top is a suitable time scale. Each activity is represented by a bar; the position and length of the bar reflects the start date, duration and end date of the activity. This allows you to see briefly different project activity. Table 1 indicate the task sheet of the different project activities.

