Cybersecurity through Crowdsourcing

Deloitte

Use Case One
Vulnerability & Penetration Testing Platform For Cloud Applications

April 30, 2018

Group Two:
Yiqing(Alice) Guo
Steve Cardozo
Li Chen
Pan Deng
Mengwen Zhou

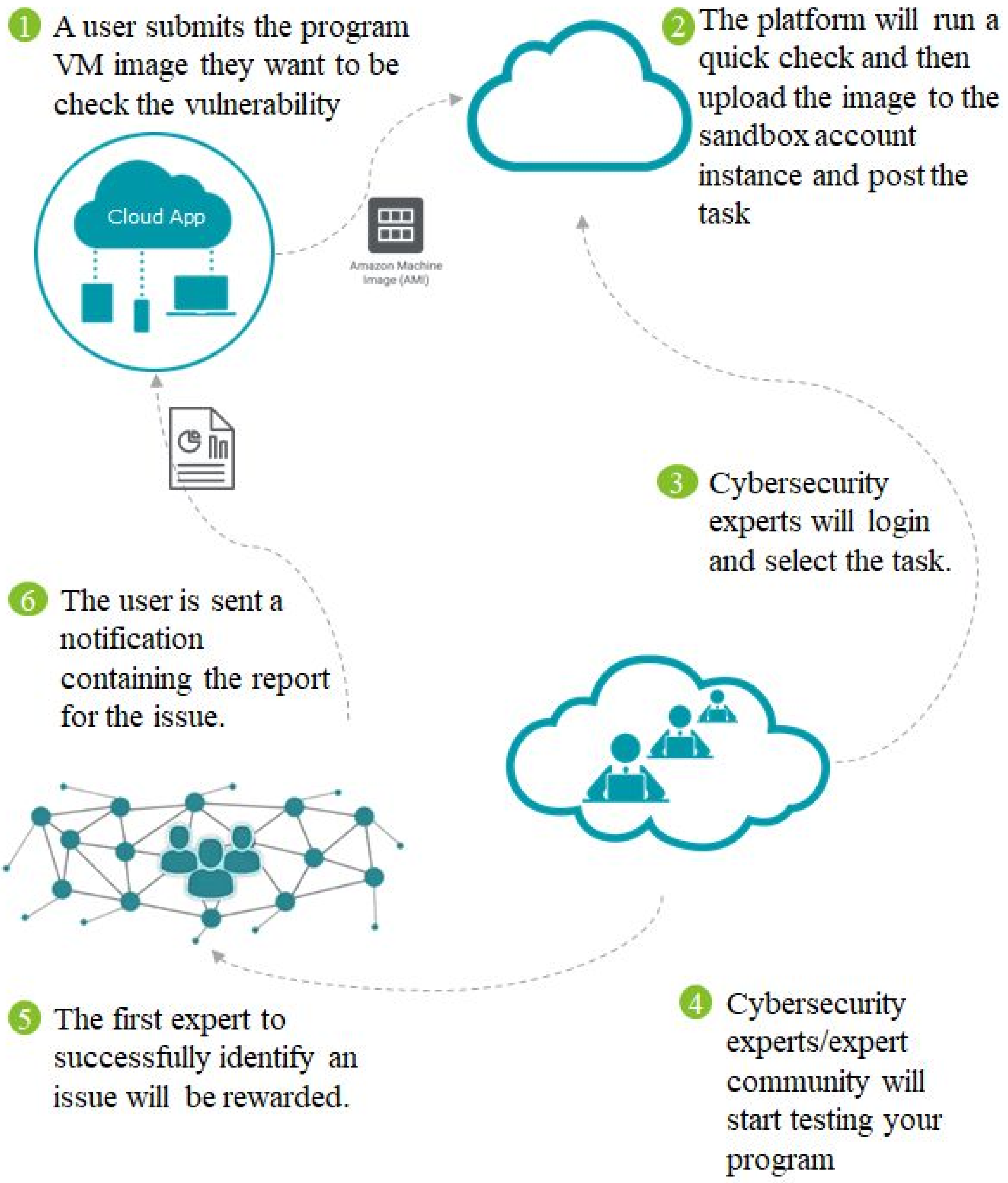
Information Technology Master's Capstone
ITWS-6800

Prof. Richard M. Plotka

Use Case One

Table of Content

Introduction



1 A user submits the program VM image they want to be check the vulnerability

2 The platform will run a quick check and then upload the image to the sandbox account instance and post the task

3 Cybersecurity experts will login and select the task.

4 Cybersecurity experts/expert community will start testing your program

5 The first expert to successfully identify an issue will be rewarded.

6 The user is sent a notification containing the report for the issue.

Targeted Issue

In this day and age, code and software are released quickly and frequently. The traditional way of testing does not meet the security testing needs of modern cloud based applications. Companies are eager to implement new technologies into their websites and systems on the cloud, whether it's to increase their operational efficiency or to be able to provide their users with the best user experience possible. Companies also face the pressure of matching the most recent capabilities offered by competitors. While focusing on further developing their systems with the goal of trying to roll out new features, a company's IT department may possibly overlook some subtle bugs or loopholes in the newly updated codebase. It is essential to test cloud applications prior to deployment to ensure security and optimal performance. Cloud application relies on remote servers for processing logic that is accessed through a web browser with a continual internet connection. Cloud applications must be tested to ensure processing logic is error-free.

The following points are some possible results if these weaknesses are exploited:

- Information loss
- Monetary loss
- Damage to reputation
- Customer dissatisfaction
- Loss of customers

Solution Purpose

Even with having a dedicated IT team, it is still possible for a company to have some small bugs or errors go unnoticed in their code. This solution allows a company to have its code analyzed by experienced developers around the world. With more eyes looking over the code to check for any cybersecurity issues, a company can feel more reassured that its system is safe and secure.

The Vulnerability & Penetration Testing Platform For Cloud Applications will transform traditional security testing to a data-driven vulnerability management tool. The proposed solution will provide a complete and actionable vulnerability report including an executive summary, identified issues ordered by how critical they are, and remediation recommendations to the client.

The platform will use the Amazon Web Services (AWS) Organizations service, which offers policy-based management for multiple AWS accounts[1]. With this service, platform

[1] "AWS Organizations – Policy-Based Management for Multiple Accounts - AWS." Amazon Web Services, Inc., aws.amazon.com/organizations/.

administrators will be able to create different groups of accounts and assign different policies to each group to control the use of the AWS services. Clients' software will only be uploaded to accounts under the sandbox group, which only get access to EC2, RDS, CloudFront, API Gateway and Lambda services from Amazon. AWS Organizations APIs will also be used to automate the creation and management of new AWS accounts for each client to upload their application. When the test duration is reached, the accounts for those instances will be deleted.

This platform will be built on Amazon Web Server. It will be able to automatically deploy client-provided virtual machine (VM) images through a user friendly interface. The Machine Image provides the information required to launch an instance, which is a virtual server in the cloud.[2] An Amazon Machine Image (AMI) is a VM image specifically for Amazon Elastic Compute Cloud (EC2). It includes a template for the root volume for the instance; for example, an operating system, an application server, and applications, launch permissions that control which AWS accounts can use the AMI to launch instances and a block device mapping that specifies the volumes to attach to the instance when it's launched. If the client is already using AWS, then the client will be able to simply copy their AMI to the platform's AWS account through a user friendly interface, and grant the platform's account permission to run this image. If the client is using a different cloud service, the virtual machine image can still be imported to the platform's AWS account as long as it meets the requirements of AWS. All of the images provided by clients will be encrypted before they are launched on the platform. Clients can also choose to encrypt their AMI themselves before copying it to the platform. It is possible to copy an unencrypted snapshot to yield an encrypted snapshot, and not available to copy an encrypted snapshot to yield an unencrypted one. Following table shows different scenario of copy AMI[3].

Scenario	Supported
Unencrypted-to-unencrypted	Yes
Encrypted-to-encrypted	Yes
Unencrypted-to-encrypted	Yes
Encrypted-to-unencrypted	No

Clients are required to agree to the Terms and Conditions of Use in order to use the platform.

[2] "Amazon Machine Images (AMI)." Amazon Machine Images (AMI) - Amazon Elastic Compute Cloud, docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/AMIs.html.

[3] "Copying an AMI." Copying an AMI - Amazon Elastic Compute Cloud, docs.aws.amazon.com/AWSEC2/latest/UserGuide/CopyingAMIs.html.

To ensure the security and transparency of all activity on the instances, a record will be made on the blockchain. The blockchain is a distributed ledger that serves as a permanent and shared record of every transaction associated with an asset, which ultimately creates an unbroken chain of trust. Each record will be time-stamped and appended to the event before it[4]. Instead of having the platform check every action performed, a blockchain-based record system will be used. It uses a large number of networked units that perform authentication, rather than an intermediary that verifies entries[5]. Clients will be able to access all data records regarding their software and track all activity on their software. Due to the decentralization, the blockchain record system is not controlled by a single person or the platform. It is controlled by the network, and the transactions are authenticated through the network so that the original data, such as the instance information and attack information, cannot be changed. This prevents both the clients and testers from violating the rules of the platform, If there is any violation on the record, a fine will be charged to the violator and this user will be added to the platform's blacklist. Moreover, the blockchain record system ensures that the reward is always given to the expert who finds an issue first. The solution offers a platform to harness the power of all talented cybersecurity experts to identify vulnerabilities of clients' software - out of place.

Crowdsourcing Approach

In his book, *Crowdsourcing*[6], Daren C. Brabham proposed a problem-based typology of crowdsourcing approaches: knowledge discovery and management, distributed human intelligence tasking, broadcast search, and peer-vetted creative production. The team uses two of the approaches, knowledge discovery and management and distributed human intelligence tasking, to find vulnerabilities in clients' programs. The proposed crowdsourcing system will allow a company to post a task for a large base of certified cybersecurity experts to work on for a certain duration. Posting a task requires submitting a machine image, whether it is their entire codebase or just a small portion of code that creating to the machine image, along with a description of the work to be completed and reward amount. The crowd of cybersecurity experts will help to identify any potential security issues in the clients' software so they can fix them at an early stage.

Potential Clients

This solution provides three different types of accounts: regular, premium, and enterprise accounts. For regular and premium accounts, the potential clients are small companies that don't have an experienced IT team. The platform's user-friendly and easy-to-use interface will attract

[4] Trust in Trade, 10 Feb. 2017, www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=GBE03771USEN.

[5] "CREDITS." Blockchain Technology and Smart Contracts in the Online Games Industry, credits.com/zh/Home/Case/29.

[6] Brabham, Daren C. Crowdsourcing. MIT, 2013.

this type of customer. They don’t need to go through the process of creating an environment for testing or worry about security issues during testing. This platform will handle this for the client. Since the platform also allows for small cases, meaning the user can choose to only have a small portion of their code analyzed, this makes it more accessible to smaller companies. The difference between regular and premium accounts is that premium accounts have the ability to view the rankings of the experts and search for specific experts or groups to test their program. Regular accounts will only be allowed to use the EC2 service to test their cloud application. Because performing penetration testing on the AWS platform requires a form to be submitted which the platform will handle for our client and take some time for Amazon to review it, the platform offers priority to premium accounts; the application process will only take up to two days for them while it may take even longer for regular accounts.

Enterprise accounts are reserved for big corporations that want to post testing software which are only accessible to their employees. Companies are given full control over their accounts and can create different user groups each with their own Identity Access Management (IAM) policies.

List of Requirements for the Crowd

Testers should pass proficiency tests before participating in any testing assignments.
All users must accept a confidentiality agreement when accepting tasks.

Potential Issues

When developing this solution, the team came up with four major potential issues. The first issue is ensuring the privacy of the software that clients upload to the system. A major concern of clients is the ability of the platform to ensure that the source code of their testing software will not be stolen. The second issue is coming up with a way for the platform to identify if the uploaded program is malware. Chances are that some users might try to upload malware to crash the platform. The third issue is that AWS has strict policy for running vulnerability and penetration testing, so the team needs to ensure that both the clients and testers adhere to the rules. Finally, the solution must address the issue of ensuring a safe transaction in the reward process.

Possible Solutions

In order to protect client software from being stolen, the team decided to utilize AWS Amazon Machine Image (AMI) to let clients upload their cloud based applications. AMI allows clients to have control over the access rights/permissions and APIs to open for testing. Once a client uploads software to the platform, the platform will go through an automatic process to create an instance for this software. Access control will then be set up through the AWS Organizations service and IAM to allow for different user groups and policies. For example, access can be granted for a specific range of dates, different AWS service APIs, and specific resources. There is also the option to enable multi-factor authentication for highly privileged users[7]. The team will use this to control the API actions and resources the application can use for different user groups, thus ensuring the source code or application would not be stolen by a tester or a third party. Also, the team will use SSL/TLS for uploading the program and encrypting image on the server-side. Also there will be a record on all the action is made to the instance on the blockchain, for those who does not follow the policy will be add to blacklist and will not be able to access this platform again.

To address the possibility of malicious code being uploaded to the platform, for every AMI and VM image copy or upload to the sandbox accounts, an automated test will be run to scan the software. If malware is found, the client will be send a warring and the instance will be delete. Because the platform is using AWS Organizations service, the sandbox accounts is only for testing, those accounts will not be able to access the platform database and source code. When each client uploaded the image, a new AWS account will be create and also a new instance, so there will not be infect on other's application that testing on the platform. Moreover, since AWS is built to meet the requirements of the most security-sensitive organizations, the malware will not be able to pass though AWS security.

The platform will be strictly follow the AWS policy, and every client and expert will need to sign agreement when they are creating user account on the platform. Also to become an expert on the platform they will need to be certified and there will be question asking prohibited testing activities on the platform. The type of testing cannot be done are as follow but not be limited to:

- perform Denial-of-Service (DoS) attacks
- simulations of such against any AWS asset
- Protocol flooding (eg. SYN flooding, ICMP flooding, UDP flooding)
- Resource request flooding (eg. HTTP request flooding, Login request flooding, API request flooding)

If any of those above activities are detect the expert will be charge fine, and may add to the blacklist depend on the conditions.

The team will use distributed ledger technology or blockchain technology to support our platform, to ensure that the first person to find an issue will receive the reward. Hyperledger is an open source blockchain technology, supporting private transactions and confidential contracts. The blockchain provides transparency to the client. It will record every action taken on clients' uploaded programs, so clients are fully aware of what happens during this process, and this

[7] "Identity and Access Management (IAM) - Amazon Web Services (AWS)." Amazon Web Services, Inc., aws.amazon.com/iam/?nc1=h_ls.

process is private transaction, which only the client who own the program and platform administrator will be able to access the record. Also the issue report will be saved to the blockchain so that no one can change the record. This will protect the tester from others trying to claim their work. The blockchain technology is also used to ensure the safety of the transaction used through smart contracts for the reward process.

Business Model

Key Partners :

- Owners of applications and web services able to embed their services into this platform.
- All Firewall and Network Security Company.
- Insurance company have service in cyber attack coverage.

The main source of the platform income:

- The fees received from transactions within the system
- The fees received for tested program
- The fee for Premium account and Enterprise account

Payment options

VPTP stands for Vulnerability & Penetration Testing Platform. VPTP is a blockchain- based platform that supports the VPTP community by implementing a rewarding system that utilizes VPTP Token, or V Token(VT), which is a cryptocurrency designed to incentivize community members to perform actions that create value within the community. VPTP enables users to find issues on the software and users get rewarded for doing so.

VPTP where users can securely and reliably trade with each other in a decentralized environment. The platform encourages users to test program on the platform. The VPTP Rewarding System guarantees that user who find problem is rewarded by V Token(VT).

Use of Token

V Token is used to:

- Submit the cloud base application wants to be tested
- Upgrade an account for extra services to choose experts, and longer test duration
- Record the activity on your software
- Pay for experts who find the issues

Types of accounts

Account Type	**Regular account**	**Premium account**	**Enterprise account**

vulnerability scan by our scanner	No	Yes	Yes
Code Access for Testing	Black Box	Black Box/Gray Box	Black Box/Gray Box/White Box
Vulnerability Report	Only when the testing duration is over	Every day as well as a final report when the testing duration is over	Every day as well as a final report when the testing duration is over
Testing Service	EC2	EC2, RDS, CloudFront, API Gateway and Lambda	EC2, RDS, CloudFront, Aurora, Lightsail, API Gateway and Lambda
Instance Types	Burstable Performance Instances small only (t2.small)	All Instances	All Instances
Image type support	AMI only	AMI, VM image	
Permission wait time	2 days and up	Up to 2 days	Up to 2 days
Max Testing duration	Up to 3 days	Up to 90 days	Up to 90 days
Multiple users	N/A	N/A	Yes (For enterprise account, the admin user can create multiple users under this account)
Search and choose experts	No	Yes	Yes
IAM Setup	N/A	N/A	Yes

Operation and Fee

Each action taken in VPTP is rewarded by regulated fee in V Token. The fees are regulated by economy observers without stopping or restarting the network. The fees can be used across a wide spectrum: from covering the cost of account to rewarding for the experts.

In order to keep this platform running and kept the record on the blockchain a base fee of upload application will be charge, also fee will be apply depend on different type of instance and test duration.

Test Server Type	Regular (Token/day)	Premium (Token/day)
T2 (General Purpose)	2	1
M5 (General Purpose)	2	1
M4 (General Purpose)	3	2
C5 (Compute Optimized)	N/A	1
C4 (Compute Optimized)	N/A	1
P2 (GPU Instances)	N/A	2
G3 (GPU Instances)	N/A	3
X1 (Memory Optimized)	N/A	6
R4 (Memory Optimized)	N/A	3
I3 (Storage Optimized)	N/A	1
H1 (Storage Optimized)	N/A	1
D2 (Storage Optimized)	N/A	3
LightSail	N/A	1 (2 days)

* All Price are basic price, price can be various depend on CPU, Memory, and Storage.

Use Case

Use Case Name: **Vulnerability & Penetration Testing Platform For Cloud Applications**
Goal: Identify cybersecurity issues in the client's software through crowdsourcing using blockchain.

***Use Case Descriptions*:**

Scenario 1:

Ben, a lead company technician, is trying to find security problems in his new cloud base application hosted on AWS. It is scheduled to be released in two weeks. Since it would be difficult for his IT team to find all of the potential issues by themselves, Ben decides to use a crowdsourcing system with certified cybersecurity experts. He signs up for a regular account, creates an AMI, copies it to the given sandbox account on the crowdsourcing platform, and supplies it with all of the necessary information. The platform automatically deploys the software and runs a basic scanner on the software. He chooses to pay 6 V Tokens to have the system tested on the platform for three days. Ben needs to wait two or three days for his request to be reviewed. An email is sent to him when it is approved and testing has started. The reward he set up is as follows: high priority level issues will be rewarded 3 coins, medium level issues will receive 2 coins, and low level issues will get 1 coin. Now Ben only needs to wait for the results. Once an issue is detected, the platform will classify the urgency of the problem and Ben will receive an email containing all of the details of the problem and how serious the issue is. After Ben confirms the identified issue, Ben will send the appropriate reward to the expert. After three days of testing, Ben gets a complete report with test data visualized, including an executive summary, a listing of risk ratings, and recommended firewall and insurance plans from the platform's partners.

Scenario 2:

Alex, a software developer, has just developed a new program by himself. As a premium user on the Vulnerability & Penetration Testing Platform, he decides to get a report for his new program. He is an Azure user, and he create VM image from Azure, then uploads his image on the platform, he decides to search for testers with five star ranks. He decide to test for 2 weeks and pays for 14 Tokens. Since Alex has a premium account, before his program gets posted to the crowd, the platform will run a vulnerability scanner, generate a report, and send it to Alex so those known vulnerabilities issues will not be rewarded and reported to him again. Then he wait for a day, the testing is being approved. Also, he will receive a report every day since the test start, so he can fix any issues as soon as possible. After a week, Alex decides to stop the testing process, and 7 Token is get back to him from the platform. A few minutes after ending the testing, a final report is send to him.

Scenario 3:

Jane works in a global company as the chief of technology. She decides to use the Vulnerability Assessment Platform to test newly developed software. Instead of only having the testing team analyze the software, she creates accounts for all the programmers in the company around the world, and set up a group for read only access to the code. A notice will be sent through the company email asking those programmers to test this new software. The development team get reports every day and after two weeks the report shows that there are no new issues found, so Jane decides to stop the testing. A report of full test results is sent to Jane and the development team.

Actors:

Primary actor: User

Users - Users post their tasks by deploying the system or app onto the platform along with a description and reward amount. Users can choose to only have a small portion of their code analyzed; it doesn’t have to be their whole system.

Secondary actors: Cyber security experts

Cybersecurity experts - Experts who have been certified by the platform by passing background checks and completing certain tests upon registering. They work to complete tasks and compete for the rewards. Once they meet the requirements of the task, they submit their results to the platform and wait for them to evaluated. After the evaluation of their solutions, they are assigned with the corresponding rewards.

Preconditions:

The user have account on the platform and have a copy of image ready

Trigger:

User upload/copy the VM image to the platform.

Basic Flow:

1. User uploads image for testing onto the platform
2. Select testing duration
3. Baice scanner to check the application
4. If the account is a premium or enterprise account:
   a. User chooses the test type (Black Box[8]/Gray Box[9]/White Box[10])
   b. Choose testing service (can be multiple)
   c. The uploaded software will be scanned through our vulnerability scanner
   d. A scanner report will be sent to the User.
5. Waiting for approval email
6. Certified cybersecurity experts begin to test the system
7. If a problem is identified by the experts:
   a. The crowdsourcing system sends an email to the User containing the details of the identified problem
   b. User confirms the identified problem
   c. New transaction records (representing the problem) to the blockchain and validate a new block by the consensus protocol
   d. The crowdsourcing system sends a notification to the expert who first identified the problem so he/she may receive the reward.

Post Conditions:

1. When the testing duration is over, a full report with visualization, an executive summary, remediation recommendations will be sent to the user on all the issues that have been identified with their corresponding levels of urgency
2. The platform assigns the reward to the first one(s) who found the distinct problem(s)
3. Predetermined amount of transaction fee is awarded to the miner

Alternate Flow:

If AWS detects malicious code in the uploaded codebase:

1. The platform will reject the submitted codebase
2. The user will be notified and penalized appropriately

[8] black box --testing without access to source code

[9] gray box -- with some information (like configuration files) but without complete access to source code)

[10] white box -- accompanied by source code

If no problem is found in the specified testing duration:

1. The platform will send an email to notify the user that no problems could be found in the system

If the test is not approved by AWS:

1. An email notice will be send, with explanation on the situation.
2. Providing option if user want to stop testing or send request again.

Operating Systems requirement on testing application[11]*:*

Windows

- Microsoft Windows Server 2003 (Standard, Datacenter, Enterprise) with Service Pack 1 (SP1) or later (32- and 64-bit)
- Microsoft Windows Server 2003 R2 (Standard, Datacenter, Enterprise) (32- and 64-bit)
- Microsoft Windows Server 2008 (Standard, Datacenter, Enterprise) (32- and 64-bit)
- Microsoft Windows Server 2008 R2 (Standard, Datacenter, Enterprise) (64-bit only)
- Microsoft Windows Server 2012 (Standard, Datacenter) (64-bit only)
- Microsoft Windows Server 2012 R2 (Standard, Datacenter) (64-bit only) (Nano Server installation not supported)
- Microsoft Windows Server 2016 (Standard, Datacenter) (64-bit only)
- Microsoft Windows 7 (Professional, Enterprise, Ultimate) (US English) (32- and 64-bit)
- Microsoft Windows 8 (Professional, Enterprise) (US English) (32- and 64-bit)
- Microsoft Windows 8.1 (Professional, Enterprise) (US English) (64-bit only)
- Microsoft Windows 10 (Professional, Enterprise, Education) (US English) (64-bit only)

Linux/Unix (64-bit)

- Ubuntu 12.04, 12.10, 13.04, 13.10, 14.04, 14.10, 15.04, 16.04, 16.10
- Red Hat Enterprise Linux (RHEL) 5.1-5.11, 6.1-6.9, 7.0-7.3 (6.0 lacks required drivers)
- SUSE Linux Enterprise Server 11 with Service Pack 1 and kernel 2.6.32.12-0.7
- SUSE Linux Enterprise Server 11 with Service Pack 2 and kernel 3.0.13-0.27
- SUSE Linux Enterprise Server 11 with Service Pack 3 and kernel 3.0.76-0.11, 3.0.101-0.8, or 3.0.101-0.15
- SUSE Linux Enterprise Server 11 with Service Pack 4 and kernel 3.0.101-63
- SUSE Linux Enterprise Server 12 with kernel 3.12.28-4
- SUSE Linux Enterprise Server 12 with Service Pack 1 and kernel 3.12.49-11
- CentOS 5.1-5.11, 6.1-6.6, 7.0-7.4 (6.0 lacks required drivers)
- Debian 6.0.0-6.0.8, 7.0.0-7.8.0, 8.0.0
- Oracle Linux 6.1-6.6, 7.0-7.1

[11] "VM Import/Export Requirements." VM Import/Export Requirements - VM Import/Export, docs.aws.amazon.com/vm-import/latest/userguide/vmie_prereqs.html.

- Fedora Server 19-21

Import from VM[12]*:*

Windows:

- Enable Remote Desktop (RDP) for remote access.
- Make sure host firewall (Windows firewall or similar), if configured, allows access to RDP. Otherwise, will not be able to access instance after the import is complete.
- Make sure that the administrator account and all other user accounts use secure passwords. All accounts must have passwords or the importation might fail.
- Install the appropriate version of .NET Framework on the VM. Note that .NET Framework 4.5 or later will be installed automatically on VM if required.

Linux:

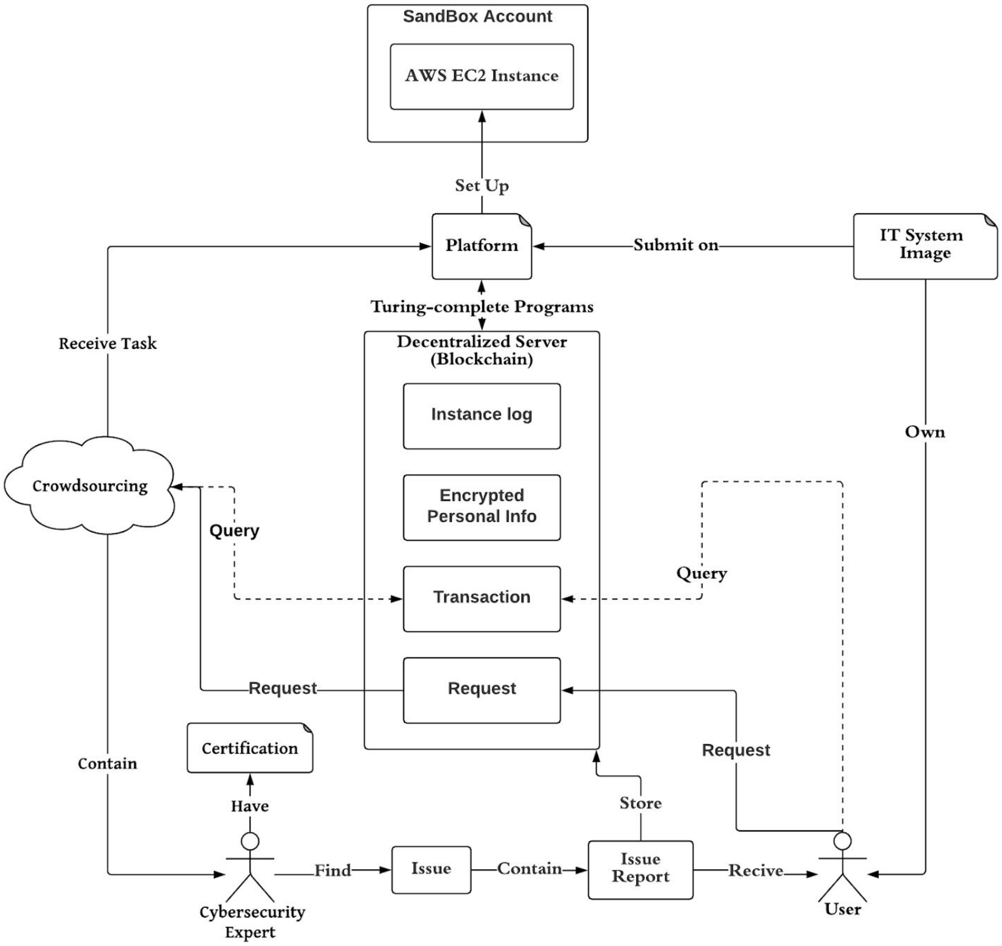
- Enable Secure Shell (SSH) for remote access.
- Make sure that host firewall (such as Linux iptables) allows access to SSH. Otherwise, won't be able to access instance after the import is complete.
- Make sure that have configured a non-root user to use public key-based SSH to access instance after it is imported. The use of password-based SSH and root login over SSH are both possible, but not recommended. The use of public keys and a non-root user is recommended because it is more secure. VM Import will not configure an ec2-user account as part of the import process.
- Make sure that Linux VM uses GRUB (GRUB legacy) or GRUB 2 as its bootloader.
- Make sure that Linux VM uses one of the following for the root file system: EXT2, EXT3, EXT4, Btrfs, JFS, or XFS.

[12] "VM Import/Export Requirements." VM Import/Export Requirements - VM Import/Export, docs.aws.amazon.com/vm-import/latest/userguide/vmie_prereqs.html.

Conceptual Design / Prototype

System diagram

In this use case, the team integrated the concept of crowdsourcing and blockchain technology into the process of vulnerability identification. In this solution, the client is able to identify vulnerabilities in their IT product, such as an application or program, in an easy and fast way. The platform allows users to submit their application system image and program will automatically create an instance to run the application on Sandbox account. Our cybersecurity expert community can see the new application and start to analyze it for vulnerabilities. Once an issue is found, a record will be stored on the blockchain for it and the client can reward the expert using the blockchain. Also all the instance activities will store in the blockchain. The conceptual design of the system is shown in the diagram below.



Conceptual Design Diagram

System Requirements

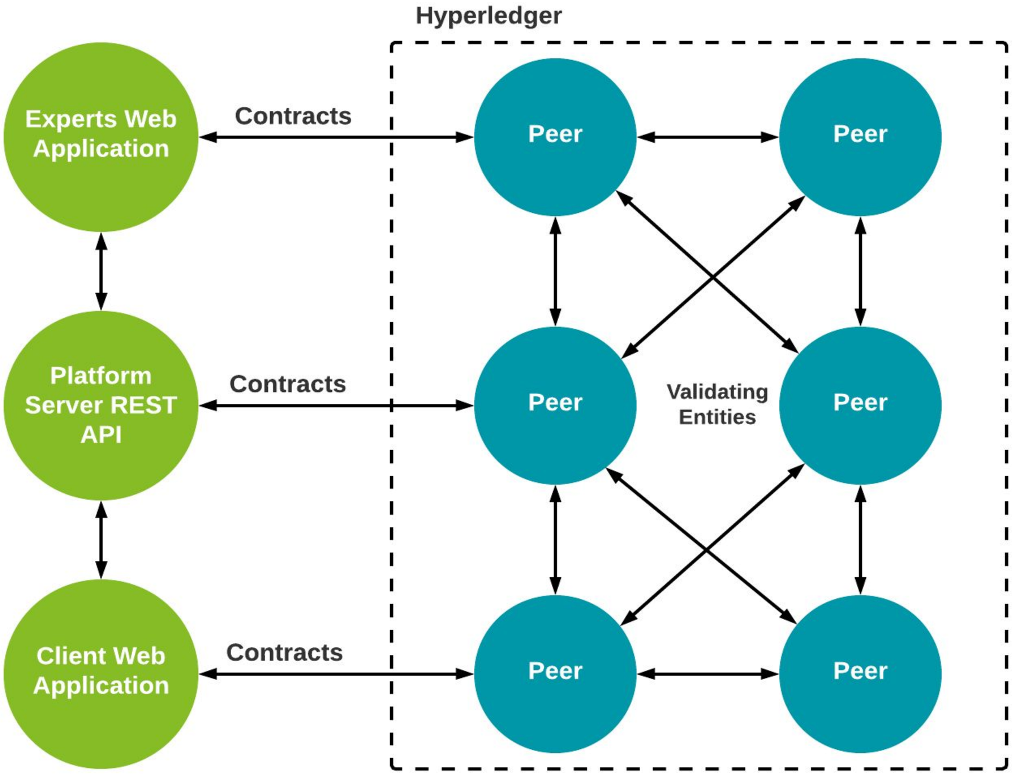
The interface of the platform will be implemented with HTML5, CSS3, Javascript, and PHP. The server side will be implemented with Java and run on AWS. The team decided AWS Organizations service to run the user application under sandbox group accounts that can only access limited service. The interface of this platform will support different browsers and different operating systems.

Processing and storing data in VPTP will be provided by VPTP REST API. This solution allows access to data over the network, regardless of device type (through a web application. Applications will communicate with REST API over HTTPS, using JSON as message format.

Minimum requirements for tester and client:

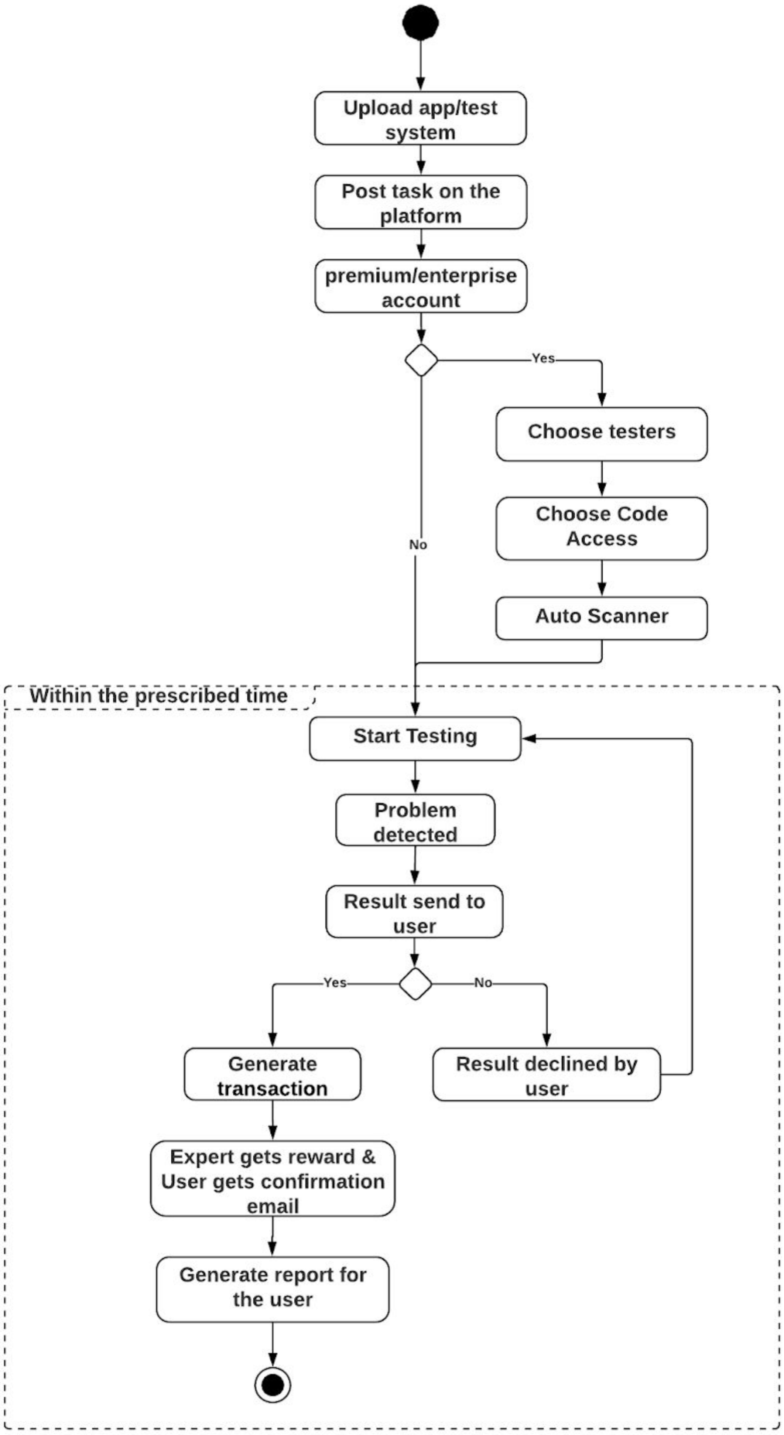
- Computer with a browser

Architecture Diagram

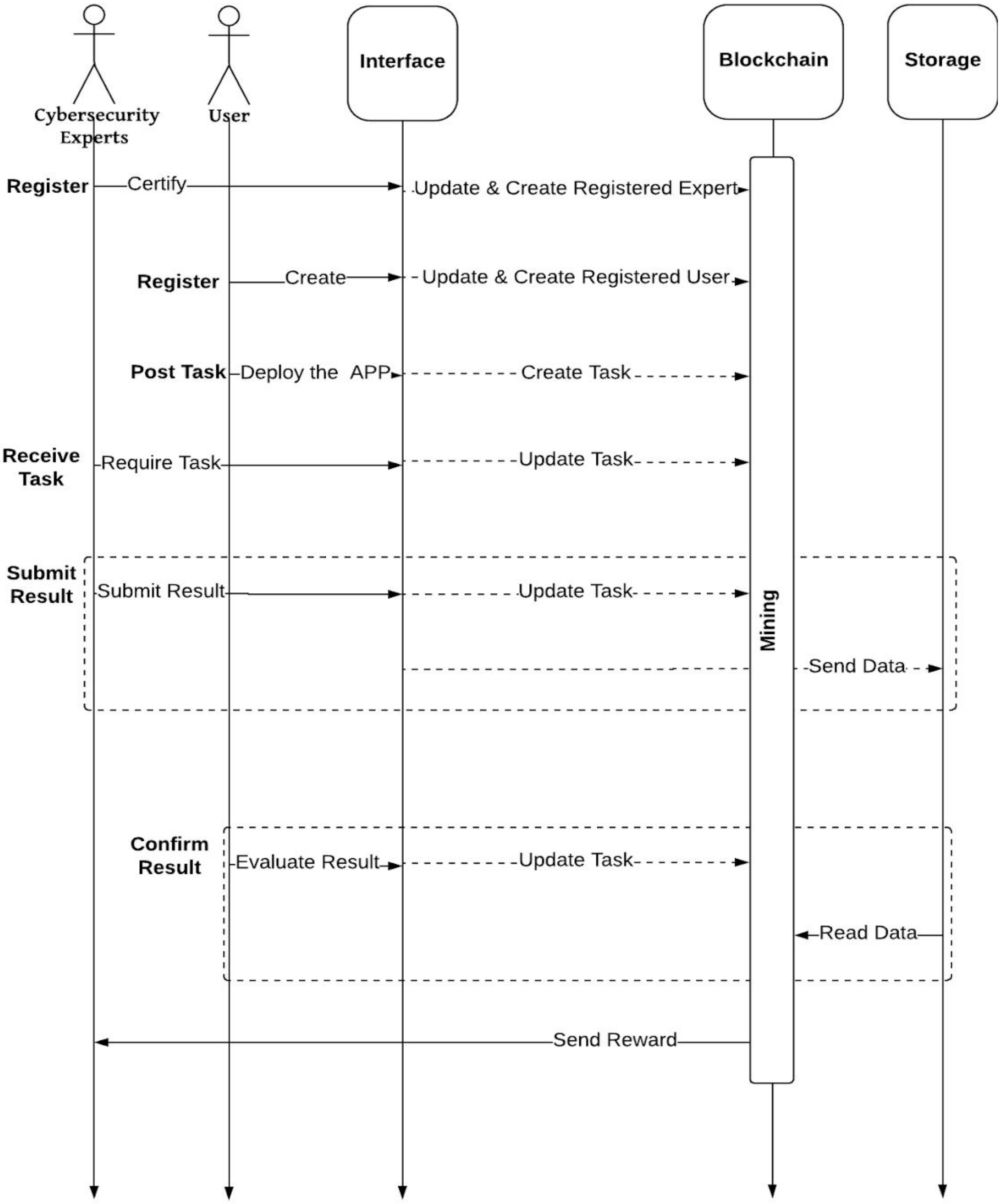


Architecture Diagram.

Activity Diagram

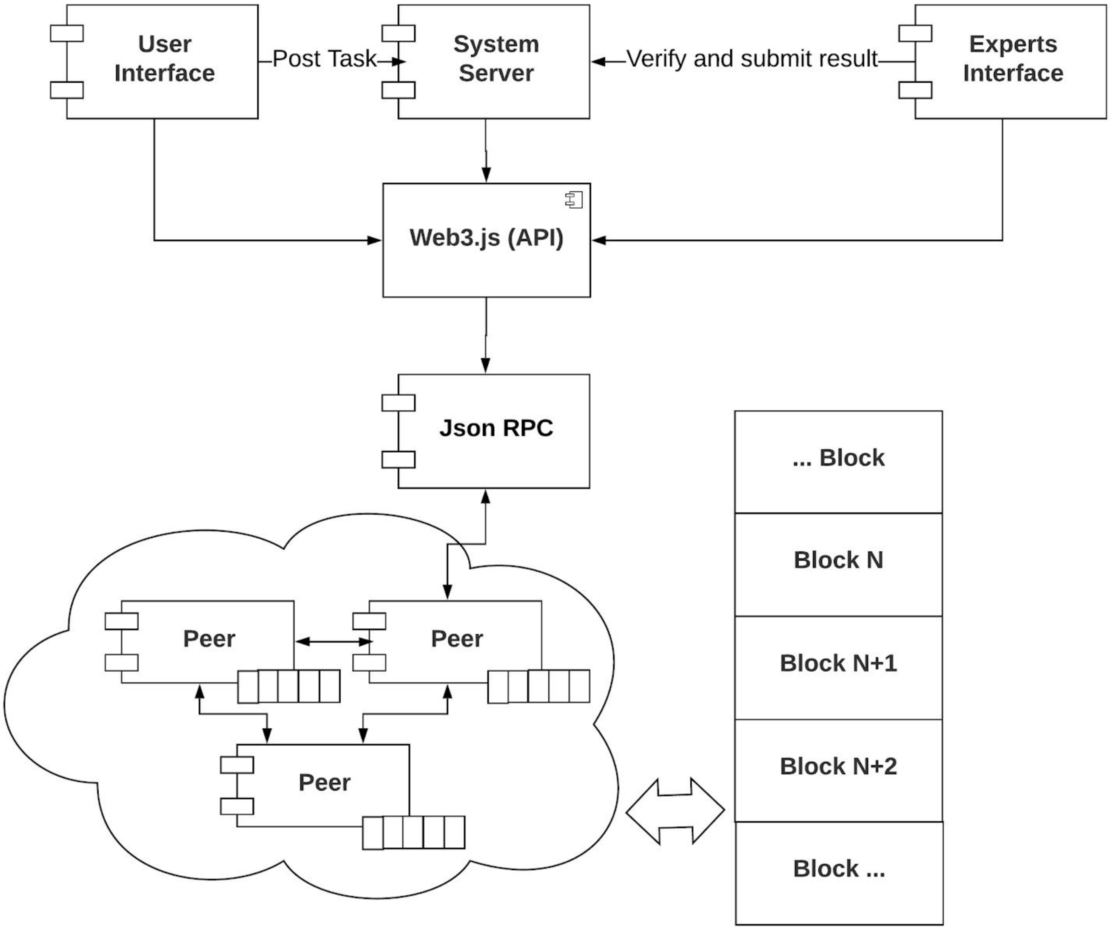


Sequence Diagram



Sequence Diagram

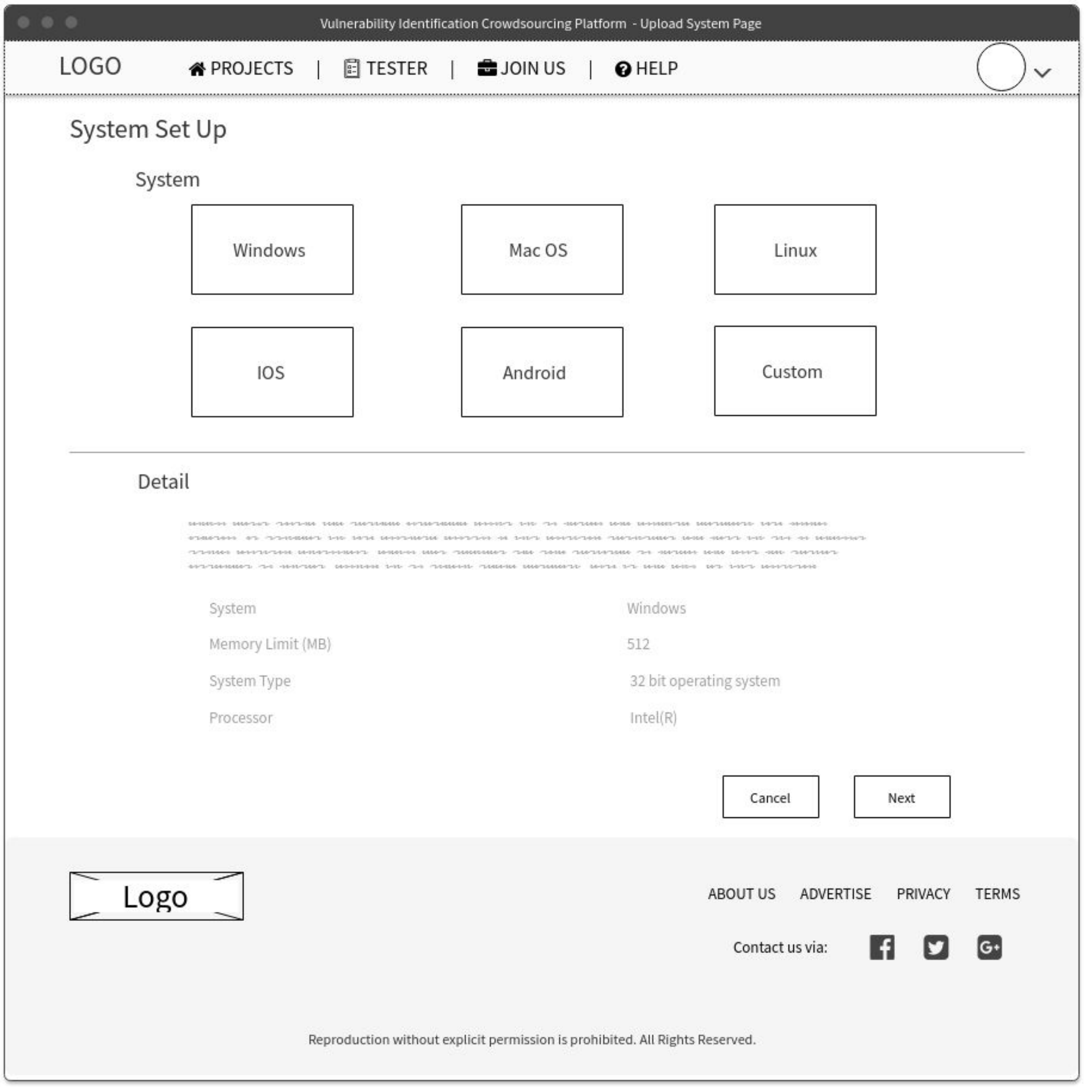
Component Diagram



Component Diagram

Human Computer Interaction

The client interface is targeted towards customers looking to test their IT system. Clients have the ability to post tasks for testers to complete and request testing for a specified duration. Posting a task involves uploading a simulation of their IT solution to the server provided by the Cybersecurity Through Crowdsourcing platform along with a description of the task to be completed and a fee that will serve as a reward for the testers. Clients can also view tester profiles, and if they feel impressed by certain testers' ratings, they can even send requests to those testers to ask them to analyze their system. Also, the service provides clients with a clear and easy understandable report for the vulnerability issues.

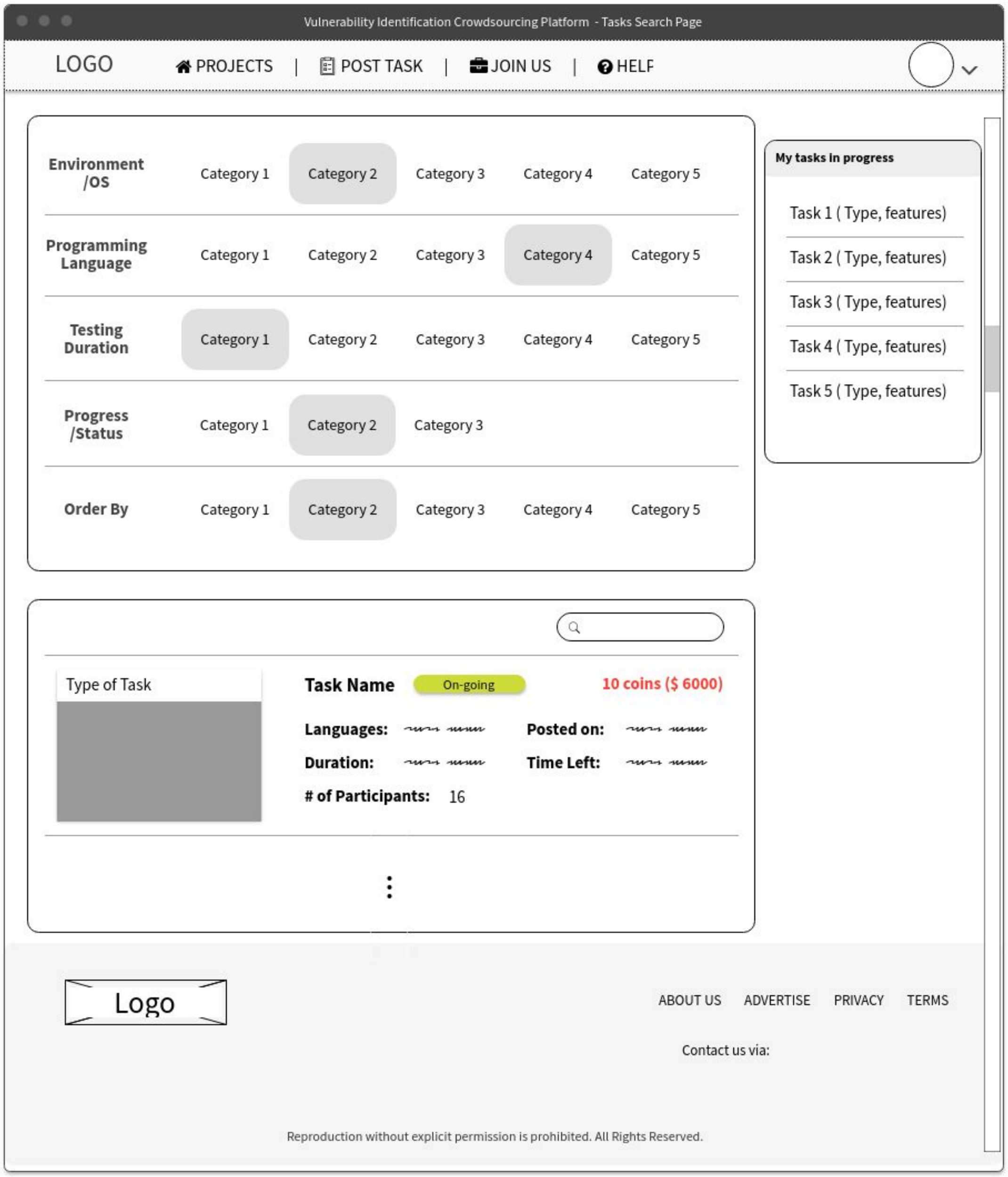


System	Windows
Memory Limit (MB)	512
System Type	32 bit operating system
Processor	Intel(R)

Reproduction without explicit permission is prohibited. All Rights Reserved.

Client Interface Wireframe.

The tester interface is intended for cybersecurity experts who wish to take part in this system's crowdsourcing efforts to find software vulnerabilities. Experts can apply to register as a tester, which will require them to fill out their profile with the skills they know and take a test so the system can certify them. After being certified, testers can view tasks and select certain tasks to compete against others for a reward. Each tester has a profile containing the skills they have and all of the tasks they have completed. Completing tasks will add value to their profile and boost their reputation.



Tester Interface Wireframe.

The administrator interface provides for enterprise account with access to various administrative actions. The most important functions are account management, tester selection, and reviewing tester solutions.