National College of Ireland

CONFIGURATION MANUAL

MSc Research Project
MSc in Cybersecurity

RYAN CHERIAN ROY
Student ID: x23277220

School of Computing
National College of Ireland

Supervisor: Kamil Mahajan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: RYAN CHERIAN ROY

Student ID: X23277220

Programme: MSc in Cybersecurity **Year:** 2024-2025

Module: MSc (Research) Practicum/Internship

Supervisor: Kamil Mahajan

Submission Due Date: 01/09/2025

Project Title: Blockchain-Integrated Detection and Prevention System with Hash-Based Validation and AI under Zero Trust Security

Word Count: 1156 **Page Count : 7**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: RYAN CHERIAN ROY

Date: 31/08/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	

A Blockchain-Integrated Detection and Prevention System with Hash-Based Validation and AI under Zero Trust Security

Ryan Cherian Roy
Student ID: x23277220

1 Introduction

This manual guides you through setting up the Blockchain-Enabled Intrusion Detection and Prevention System (BIDPS) on Windows.
It integrates Sysmon for event logging, PowerShell for file hash generation, a private Ethereum blockchain for hash verification, and the Random Forest anomaly detection model for identifying advanced threats.

2 Prerequisites

- Windows 10/11 system (Admin rights required)
- Internet connection
- Node.js and npm installed
- MetaMask browser extension for blockchain interaction
- Ganache (for local Ethereum blockchain testing)
- Familiarity with Command Prompt, PowerShell, and running basic scripts

3 System Installation

A. System Preparation

- Ensure you have administrator rights.
- Create a working directory, e.g., C:\BIDPS.
- Ensure your firewall allows local Ganache and Next.js communication.

B. Sysmon Installation & Configuration

Option 1: Download Community Config
Download the SwiftOnSecurity Sysmon config:

- Direct download: sysmonconfig-export.xml

Option 2: Create "Log Everything" Config
For testing, create a file named logall.xml with

```
<Sysmon schemaversion="4.50">
  <HashAlgorithms>sha256</HashAlgorithms>
  <EventFiltering>
    <ProcessCreate onmatch="exclude"/>
    <FileCreateTime onmatch="exclude"/>
    <NetworkConnect onmatch="exclude"/>
    <ServiceStateChange onmatch="exclude"/>
    <ProcessTerminate onmatch="exclude"/>
    <DriverLoad onmatch="exclude"/>
    <ImageLoad onmatch="exclude"/>
    <CreateRemoteThread onmatch="exclude"/>
    <RawAccessRead onmatch="exclude"/>
    <ProcessAccess onmatch="exclude"/>
    <FileCreate onmatch="exclude"/>
    <RegistryEvent onmatch="exclude"/>
    <FileCreateStreamHash onmatch="exclude"/>
    <PipeEvent onmatch="exclude"/>
    <WmiEvent onmatch="exclude"/>
    <DnsQuery onmatch="exclude"/>
    <FileDelete onmatch="exclude"/>
    <ClipboardChange onmatch="exclude"/>
    <ProcessTampering onmatch="exclude"/>
  </EventFiltering>
</Sysmon>
```

C. Install/Update Sysmon with the Config

1. Open Command Prompt as Administrator
2. Navigate to the folder containing sysmon.exe and your config file
3. To install fresh:
4.

```
sysmon.exe -i sysmonconfig-export.xml
```

   To update an existing install:

```
sysmon.exe -c sysmonconfig-export.xml
```

5. Verify installation:
   You should see “Sysmon successfully installed/updated.”

D. Verify Sysmon is Logging All Events

- Open Event Viewer (eventvwr.msc)
- Go to:
  Applications and Services Logs → Microsoft → Windows → Sysmon → Operational

E. Generate System Integrity Hash

A. PowerShell Hash Script

1. Open PowerShell as Administrator
2. Create and run this script (save as hash.ps1):

```powershell
$fileHashes = @()
$hashesPath = "C:\integrity\system_hashes.csv"
$finalHashPath = "C:\integrity\final_hash.txt"
$excludedDirs = @("C:\AdminDocs")

# Create folder if not exist
if (!(Test-Path "C:\integrity")) {
    New-Item -ItemType Directory -Path "C:\integrity" | Out-Null
}

# Get all logical drives
$drives = Get-PSDrive -PSProvider FileSystem | Where-Object { $_.Free -gt 0 }

foreach ($drive in $drives) {
    Get-ChildItem -Path $drive.Root -Recurse -File -ErrorAction SilentlyContinue |
    Where-Object {
        $_.Attributes -notmatch "ReparsePoint" -and
        ($excludedDirs -notcontains $_.DirectoryName) -and
        $_.FullName -notmatch '\\(WinSxS|Temp|AppData|ProgramData|System Volume Information|Recovery|\$Recycle.Bin)\'
    } | ForEach-Object {
        try {
            if (Test-Path $_.FullName -PathType Leaf) {
                $hash = Get-FileHash -Path $_.FullName -Algorithm SHA256 -ErrorAction Stop
                $fileHashes += $hash
            }
        } catch {}
    }
}

# Export hashes and generate final integrity hash
$fileHashes | Export-Csv $hashesPath -NoTypeInformation
$finalHash = Get-FileHash -Path $hashesPath -Algorithm SHA256
Set-Content -Path $finalHashPath -Value $finalHash.Hash -Encoding ascii

```

The final hash is in system_integrity_hash.txt.

F. Blockchain Logging Setup

A. Smart Contract Deployment

1. Install Ganache (for local blockchain):
    - Download here
    - Start Ganache (leave running)
2. Create a Solidity contract file (e.g., IntegrityChecker.sol):

solidity

```
pragma solidity ^0.8.0;

contract IntegrityChecker {
    address public owner;
    bytes32 public systemHash;

    constructor() {
        owner = msg.sender;
    }

    function setSystemHash(bytes32 _hash) public {
        require(msg.sender == owner, "Only owner can set hash");
        systemHash = _hash;
    }

    function checkHash(bytes32 _hash) public view returns (bool) {
        return (_hash == systemHash);
    }
}
```

3. Compile & Deploy using Remix (for quick start) or Hardhat/Truffle (for CLI workflow)
    - Connect Remix to your Ganache network (custom HTTP provider)
    - Deploy the contract and copy its address

B. Next.js 15 Frontend Setup

1. **Initialize Next.js Project:**

```
npx create-next-app@latest integrity-check-app
cd integrity-check-app
```

2. **Install Dependencies:**
3.
```
npm install ethers
```

4. **Basic Frontend Code (pages/index.js):**
    - (Here's a simple UI for setting and checking the hash)
5. **Run Your Next.js App:**

```
npm run dev
```

6. **Visit** http://localhost:3000 in your browser.

G. Usage Workflow

1. Generate a system hash with PowerShell (hash.ps1).
2. Open the Next.js frontend.
3. Paste the hash into the form and save to blockchain.
4. For periodic checks:
    - Generate new hash.
    - Paste into form and use "Check Integrity"
    - If the hash matches, integrity is OK; if not, you'll get an alert.
5. Sysmon is always logging ALL events. If you ever get a mismatch, review Sysmon logs for forensic details (what was changed, when, by whom).

F. Random Forest Anomaly Detection Setup

1. Training Pipeline

- Dataset Used: filtered_dataset_v2.csv containing 1,975,096 entries and 94 features.
- Feature Engineering:
    - Time-based features: SystemTime_day, SystemTime_hour, SystemTime_week_*, etc.
    - Event-based features: EventID_*, ProcessId, Execution_ProcessID
    - Host identifiers: Computer_*
    - Network indicators: DestinationPortName_, SourceIsIpv6_, Initiated_*
- Model: RandomForestClassifier(random_state=42)
- Training Split: 80/20 stratified split
- Top Features:
    - EventRecordID, SystemTime_month_6, SystemTime_week_25, Computer_LAPTOP-ROPR18AK, EventID_10, ProcessId, etc.
- Model Export: Saved as rf_model.pkl using joblib

2. Prediction Interface

- Script: windows fileless event attack detection prediction.html

  Train RF model on Sysmon logs dataset:

```
import joblib
model = joblib.load('rf_model.pkl')
```

- Deploy model in backend for anomaly detection.
- Integrate prediction output with dashboard.

3. Event ID creator Script:

```
Get-WinEvent -LogName "Microsoft-Windows-Sysmon/Operational" -MaxEvents 100 |
ject Id, TimeCreated,
/entRecordID';Expression={$_.Id}},
/stemTime';Expression={$_.TimeCreated.ToString('o')}},
ocessId';Expression={$_.Properties[3].Value}},
mputer';Expression={$_.MachineName}},
/entID';Expression={$_.Id}},
ocessName';Expression={$_.Properties[4].Value}}

ConvertTo-Json -Depth 5 | Out-File "C:\Users\jisha\OneDrive\Documents\sysmon intergity checking - new\sysmon intergity checking\client\sysmon_events.json"
```

- Run this script in powershell as an admin and load the converted event ID json file into the anomaly detection interface.

4 Instruction:

Step 1: Prepare Environment

- Ensure Python environment with scikit-learn, pandas, joblib, matplotlib, seaborn installed.
- Place rf_model.pkl in the working directory.

Step 2: Convert Sysmon Logs

- Run the PowerShell script to generate sysmon_events.json.
- Optionally preprocess or map fields to match expected model input.

Step 3: Run Prediction

- Open the prediction interface.
- Manually input values from JSON (this process can be automated)
- Review prediction result and probability scores.

5 Usage Workflow

1. Sysmon runs continuously to log system events.
2. PowerShell script generates hash → compared locally.
3. Unverified hash sent to blockchain via frontend.
4. Random Forest evaluates Sysmon logs for anomalies.
5. Dashboard displays:
    - Integrity status
    - Anomaly alerts
    - Blockchain validation results

6 Troubleshooting/Blockers

- **Only Event 1 visible in Sysmon** → Use full config XML. Sometimes, the Sysmon configuration will act out weirdly and thus only giving Event ID 1. In that case, instead of the modified Sysmon config XML file, use the full config, run the Sysmon, make sure all the event IDs are listed and then finally make changes to the config file to get out desired Event ID outputs.
- **Blockchain verification fails** → Check contract address & Ganache RPC. Any mismatch in the contract address in the code and the Ganache interface will result in error.
- **RF model errors** → Ensure rf_model.pkl matches training features. The dataset needs to be prepared in such a way that the training features should match.

References

Sarhan, M., Lo, W. W., Layeghy, S. & Portmann, M. (2022). HBFL: A Hierarchical Blockchain-based Federated Learning Framework for a Collaborative IoT Intrusion Detection. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0045790622005961

Albuquerque, R. et al. (2020). Monitoring File Integrity Using Blockchain and Smart Contracts. Available at: https://www.academia.edu/download/93913013/09246586.pdf

Peelam, M. S., Chamola, V. & Chaurasia, B. K. (2025). Blockchain-Enabled Intrusion Detection Systems for Real-Time Vehicle Monitoring. Available at: https://www.sciencedirect.com/science/article/abs/pii/S2214209625000889

Achmad, Nariswari, Pratomo, Studiawan. (2025) Sysmon event logs for machine learning-based malware detection. Available at: https://www.sciencedirect.com/science/article/pii/S277291842500027X