WAMPAC Framework on Blockchain

Index

1. Overview

Develop an attack-resilient Wide Area Monitoring, Protection and Control (WAMPAC) framework, with associated computational algorithms and software tools, to prevent and mitigate cyber-attacks and achieve resilience. The application is designed in accordance with CIA guidelines for information security.

Every node in the network has a unique decentralized identity (DID). A smart contract governs user enrollment and disenrollment process based on whitelisted rules.

A distributed Network Monitoring service constantly monitors the entire network for anomalies and performs a PBFT based decision making process on agreed group policies. The service detects anomalies, localizes the affected node and provides the latest committed safe state.

Node provenance is secured by Multi Party Computation (MPC) algorithm and is stored in a distributed real time database with resilience to fabrication and modification of data.

A desktop application runs on startup to manage the provenance-monitoring tool.

2. Progress Report

Requirements	Status/ Points
Application should guarantee three principles of Secure system viz. Confidentiality, Integrity and Availability of data.	Completed. Confidentiality using Secure MPC Integrity with hash-based referencing and signature schema Availability with distributed storage
If a security compromise occurs, the application should be able to detect and localize the compromise and automatically heal the network.	Completed. Two types 1. Instant heal of user system in case of following violations (a) ICMP disabling (b) Wi-Fi adapter enabling (c) USB port enabling (d) Adding local account as admin (e) Disable exes having invalid certificates 2. Additionally, a periodic (6 hours) operation is performed by distributed verifiers analyzing OS system, security and application event logs for anomalies
Application should function without any connectivity to the internet. All the nodes in this provenance system should be connected as a separate private permissioned network, with no connectivity to the internet.	Completed. Application is a private distributed network that can function without internet.
A smart contract should be deployed on the identified Blockchain, which would	Completed. Every user node backs up secure files in the blockchain ledger

be capable of storing the provenance data and retrieve as and when needed	which can be retrieved to safe state when system fails or judged unsafe.
A web UI by default on startup should run the provenance-monitoring tool listening for an incoming message.	Completed. Web UI for users as well as verifiers can be viewed at localhost:9000
Each machine should have a unique identification as per the identified blockchain in addressing scheme	Completed. Each machine has a DID Derived from system and user characteristics

3. Modules

3.a. Identity creation and enrollment

The application has 3 types of users. The first one is the server that who’s tasked with whitelisting nodes, setting roles of each node and creating, modifying and maintaining group policies. The verifiers of the network are the second type who apart from storing the provenance are also the nodes that perform all the distributed decision making in the blockchain network. The final set are the users who are normal nodes in the network. The system logs from these nodes are periodically monitored. The nodes also have peer-peer file sharing service to securely access/update files without a centralized repository.

3.a.i. Tasks

1. Network Server
   a. Whitelist new nodes to the blockchain network
   b. Set roles of newly added nodes
   c. Add/Delete/Modify group policies
2. Verifier nodes
   a. Enrollment of user nodes
   b. Log analysis for anomalies and perform a distributed decision making to determine state of machine
   c. Encrypted provenance storage
   d. Secure backup of user encrypted data for automatic healing
3. User nodes
   a. Network Monitoring Service
   b. Tokenized file sharing application
   c. Backup file encryption and recovery decryption

3.a. ii. Identity creation

Every node (Users and Verifiers) in the network has a unique network identity called Decentralized Identity (DID) that is self-created. These unique IDs are derived from a combination user and machine characteristics fused into a user chosen image in order to create unique image that is the DID of the network [1][2] (Fig.1).

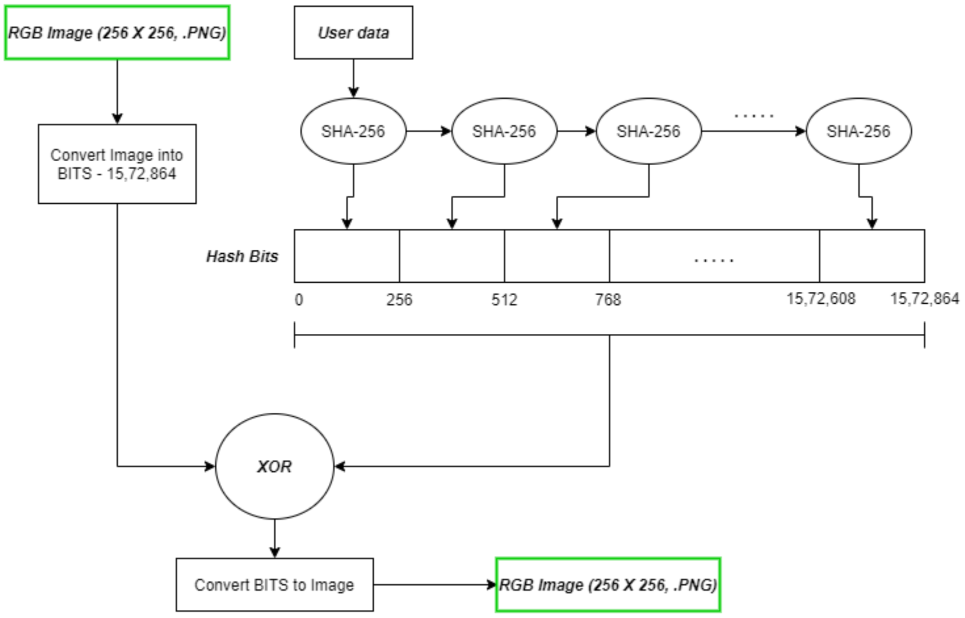


Fig. 1. DID Creation Process

3.a.iii. Share generation and key distribution

The blockchain application does not use PKI infrastructure as digital signature. Instead, relies on a secure Multi party Computation [MPC] for node authentication [3][4]. The signature mechanism follows Pedersen commitment scheme and selective bit disclosure operations to verify identity.

The DID image created is split into 2 shares: one public share and one private share [5](fig 2). The DID and public share are committed to the network by nodes after generation process on what is termed as the “Commit stage”. These values cannot be altered in network post commitment. For a user node to authenticate, it has to provide the private share proofs that corresponds to committed DID and public identities. This process is called the “reveal stage”. Proof of Knowledge are proofs of private share that are selectively disclosed at reveal stage.

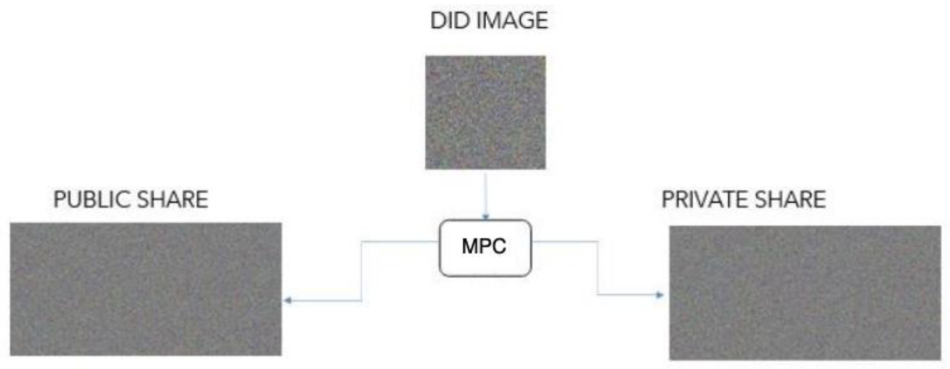


Fig. 2: Share Generation using MPC

3.a. iv. Node Enrollment process

The server node whitelists nodes based on IP Address and defines role (user or verifier). The whitelist node information is shared to all the verifier nodes.

Enrollment of nodes are of two types based on the roles

1. Verifier node enrollment
2. User node enrollment

3.a.iv.1) Verifier node enrollment

The network requires 7 verifier nodes (to follow PBFT rule) to mutually enroll each other as part of network bootstrapping. These nodes perform a challenge response-based signature schema to cross verify each other's identities. For a successful bootstrapping, all 7 nodes need to agree on the enrollment (fig 3).

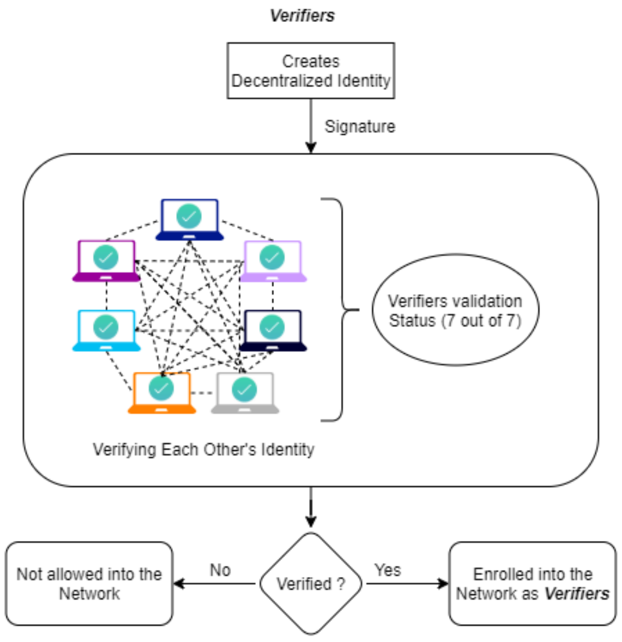

Verifiers
Creates Decentralized Identity
Signature
Verifiers validation Status (7 out of 7)
Verifying Each Other's Identity
Not allowed into the Network
No
Verified ?
Yes
Enrolled into the Network as Verifiers

Fig. 3: Verifier Node Enrollment

3.a.iv.2) User node enrollment

User nodes are added as and when they create identities in the network. The enrollment mechanism requires a consensus from minimum 5 out of 7 verifiers to validate the user. This is in accordance with the Practical Byzantine Fault Tolerance [PBFT] limit required for a distributed network [6][7] (fig 4).

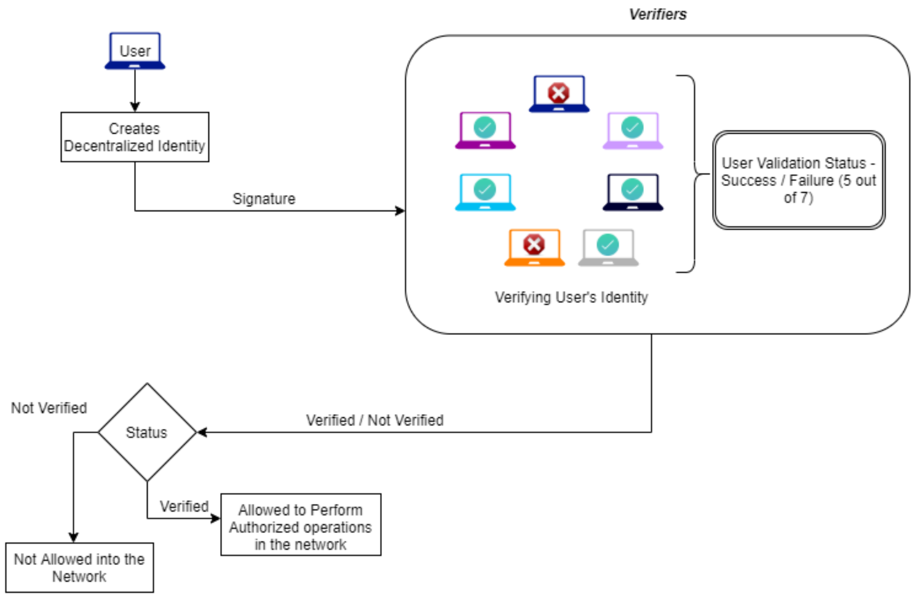

Verifiers
User
Creates Decentralized Identity
Signature
User Validation Status - Success / Failure (5 out of 7)
Verifying User's Identity
Not Verified
Status
Verified / Not Verified
Verified
Allowed to Perform Authorized operations in the network
Not Allowed into the Network

Fig. 4. User Node Enrollment

3.b. Network Monitoring and Anomaly Detection Services

This module is responsible for identifying cyber threats and maintaining a resilient network. The application is designed in such a way that confidentiality, integrity and availability of user data is maintained.

3.b.i. Network Monitoring Layer

A network monitoring layer runs locally on every node displaying the current network and memory usages. A configurable threshold can be set in order to trigger an alert to user (fig 5).

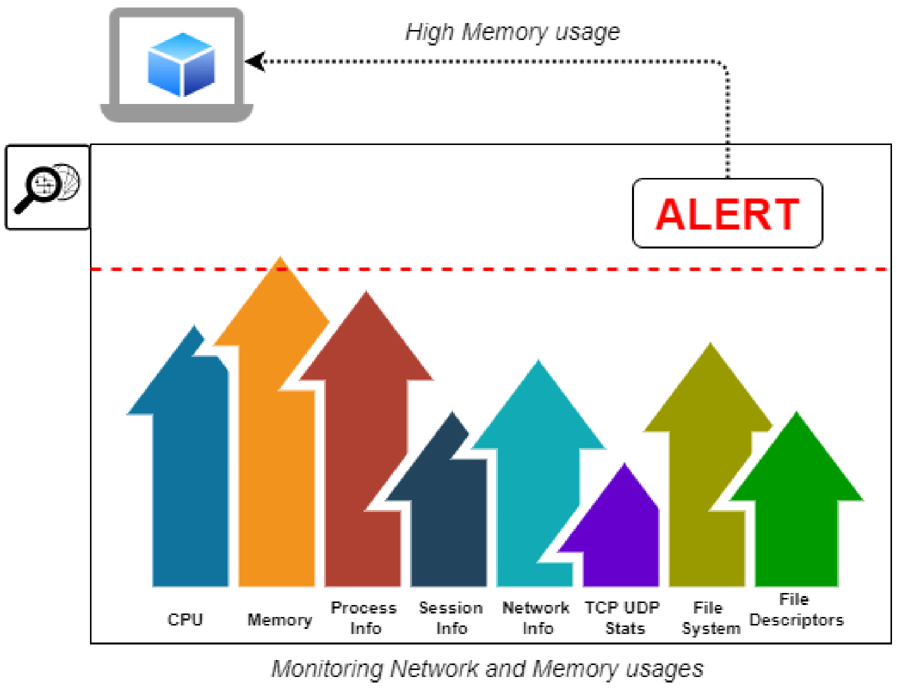


Fig. 5: Network Monitoring Service

3.b. ii. Anomaly Analysis and Detection

Periodically, logs of system, security and application related events are collected and analyzed by verifier nodes in the network. These actions are performed in a distributed manner without relying on a single server hence avoiding single point of failure. The verifier nodes independently perform analysis of user logs and arrive at a collaborative decision based on PBFT based consensus [8][9][10](fig 6).

Group policies are added, removed and modified by the server node. Some of the rules added are the expected login and logout time of user, privileged operations for every user, frequency of login failures and modification of system files.

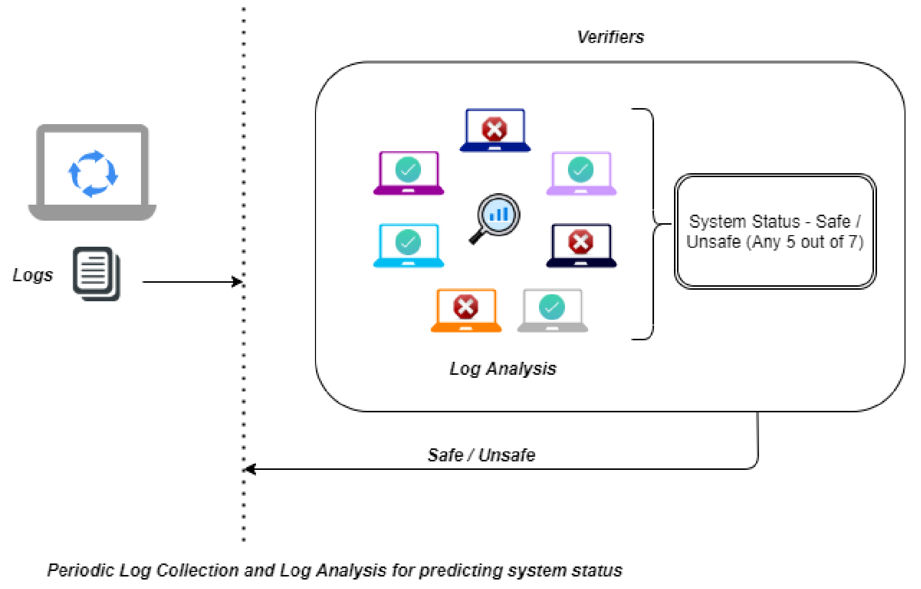

Verifiers
Logs
System Status - Safe / Unsafe (Any 5 out of 7)
Log Analysis
Safe / Unsafe
Periodic Log Collection and Log Analysis for predicting system status

Fig. 6: Log Analysis

3.b.iii. Automatic Healing

If the system is found unsafe, the verifiers advise the user to revert to the latest safe state [11][12]. The safe state stores the encrypted secure user files in the ledger. User can retrieve the encrypted files and recover the contents (fig 7).

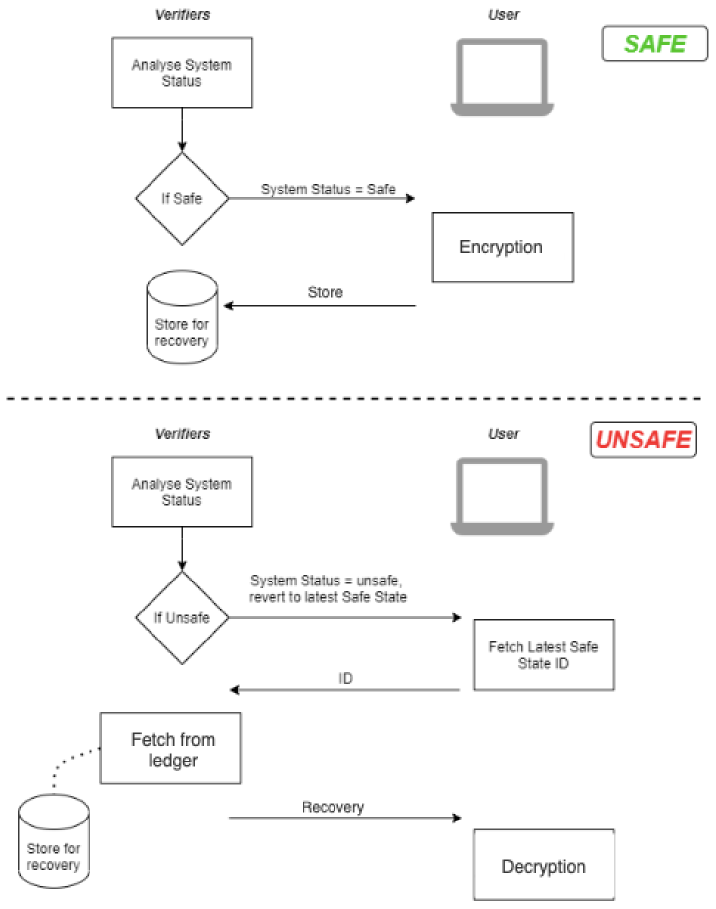


Fig. 7: Recovery Operation

3.c. File Sharing Services

File sharing module provides a platform for tokenized distributed shared file management system[13]. User nodes can use the drive to manage shared files with peer nodes. The file creator user can choose the nodes it wants to share files with and perform a peer-to-peer operation for file creation and automatic updates. The drive replicates a version control application without a centralized database (fig 8). Every file in drive is tokenized and each version update of file is linked to same token crating an immutable ledger.

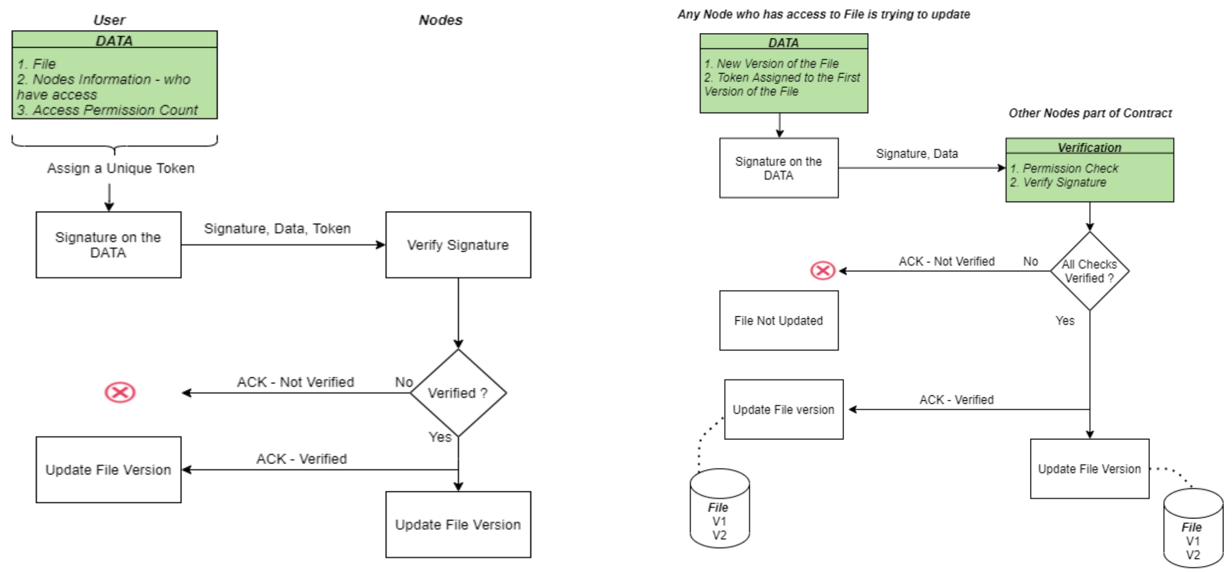


Fig. 8: Shared Drive

3.d. Network device logging and monitoring

The syslog settings on the network device (switch or router) can send logs to multiple syslog server. The Network device from where the logs are collected should be manageable either through CLI or GUI.

Device details:

1. Cisco SG300, 28 port Gigabit managed switch
2. Ubuntu 20.04 LTS *(Since windows machines does not have rsyslog, we have to setup Ubuntu machines in the network which have syslog server setup to log system information and events of the switch/router)*

The steps to configure the syslog server and the network device is shared in the User Manual (User Manual section, 3.b)

3.e. Requirements & Limitations of Network

1. The application requires 7 systems as verifiers. This is to satisfy fault tolerance required in a distributed network (following PBFT rule).
2. Limited asynchronous handling of requests to same service
3. Limited backup file size to 2MB.

4. Disenrollment of system is a manual operation and require resynchronization of network.

4. Appendix

Appendix A

Decentralized Identity

Every node joining the blockchain platform creates a unique network Decentralized Identity (DID). DID is a 256*256 PNG image that is self-created but verified by peers in the network.

DID creation take 2 input parameters, first a 256*256 PNG image of users' choice. Second seed is a SHA3-256 hash H, that is generated from user and machine characteristics. Decentralized Identifier needs to satisfy two properties; Uniqueness and Randomness. Uniqueness is required to distinctly refer each node and randomness property is essential for share generation phase. The PNG image provides randomness since its chosen by user whereas the hash H contributes towards achieving uniqueness.

Consensus protocol

Consensus involves agreement between multiple parties in a distributed network. This can be achieved by running a single chain with blocks created one after the other as an agreement to the previous blocks on the chain. However, such chains never reach a state of finality. Plus, the scalability factors for the all nodes to synchronize with other peers in the network makes these models inoperable. In the designed application, consensus protocol is run for each transaction independently. This way each transaction can be independently verified reducing forks. The value 7 is chosen based on Practical Byzantine Fault Tolerance (PBFT) algorithm. Hence, 5 out of 7 votes from the Verifier nodes is required for a successful consensus.

Provenance Storage

Multi party distributed storage scheme is used during this storage module at the Sender after successful Consensus. In the deployed network the backup files are encrypted and stored in different verifier nodes to avoid single point of failure[14].

Interplanetary File System

Every node who joins the network will be part of private IPFS and therefore they are not connected to the external IPFS network and communicate only to those nodes connected to this private IPFS Swarm. All data in the private network will only be accessible to the known peers on the private network[15].

IPFS properties

1. Immutable objects represent files
2. Objects are content addressed by cryptographic hash
3. DHT to help locate data requested by any node and to announce added data to the network
4. Removes redundant files in the network and does version control
5. Content addressing of the data stored in IPFS, every file name is unique if there is even a single character change in the content of the file
6. Private IPFS that can only be accessed by certain entities
7. Committing of the data - avoids double spending

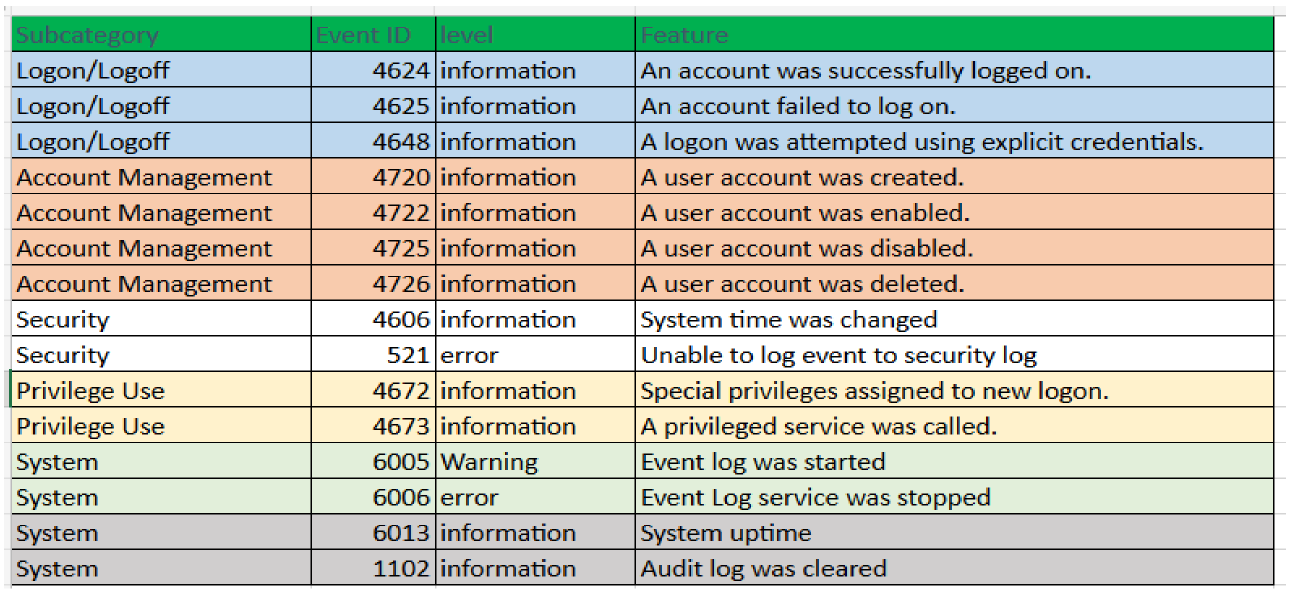
Windows Event Logs

The Windows event log contains logs from the operating system and applications.

This application logs and analyzes three major windows event logs

- Application – Information logged by applications hosted on the local machine.
- Security – Information related to login attempts (success and failure), elevated privileges, and other audited events.
- System – Messages generated by the Windows operating system.

Table1 describe the various event IDs tracked and analyzed to determine suspicious activities or misbehaviors in the system. Apart from these special event IDs, a threshold is chosen for number of errors and warnings after which the system will be deemed unsafe.



Subcategory	Event ID	level	Feature
Logon/Logoff	4624	information	An account was successfully logged on.
Logon/Logoff	4625	information	An account failed to log on.
Logon/Logoff	4648	information	A logon was attempted using explicit credentials.
Account Management	4720	information	A user account was created.
Account Management	4722	information	A user account was enabled.
Account Management	4725	information	A user account was disabled.
Account Management	4726	information	A user account was deleted.
Security	4606	information	System time was changed
Security	521	error	Unable to log event to security log
Privilege Use	4672	information	Special privileges assigned to new logon.
Privilege Use	4673	information	A privileged service was called.
System	6005	Warning	Event log was started
System	6006	error	Event Log service was stopped
System	6013	information	System uptime
System	1102	information	Audit log was cleared

Table 1: Monitored Event IDs

Group Policies

The following group policies are managed and controlled by distributed verifiers in the network.

1. ICMP should be enabled
2. Local Users from built in Admin group should be removed.
3. USB storage should be disabled
4. Exes checking
5. Certificate warning
6. Wi-Fi should be disabled

Appendix B

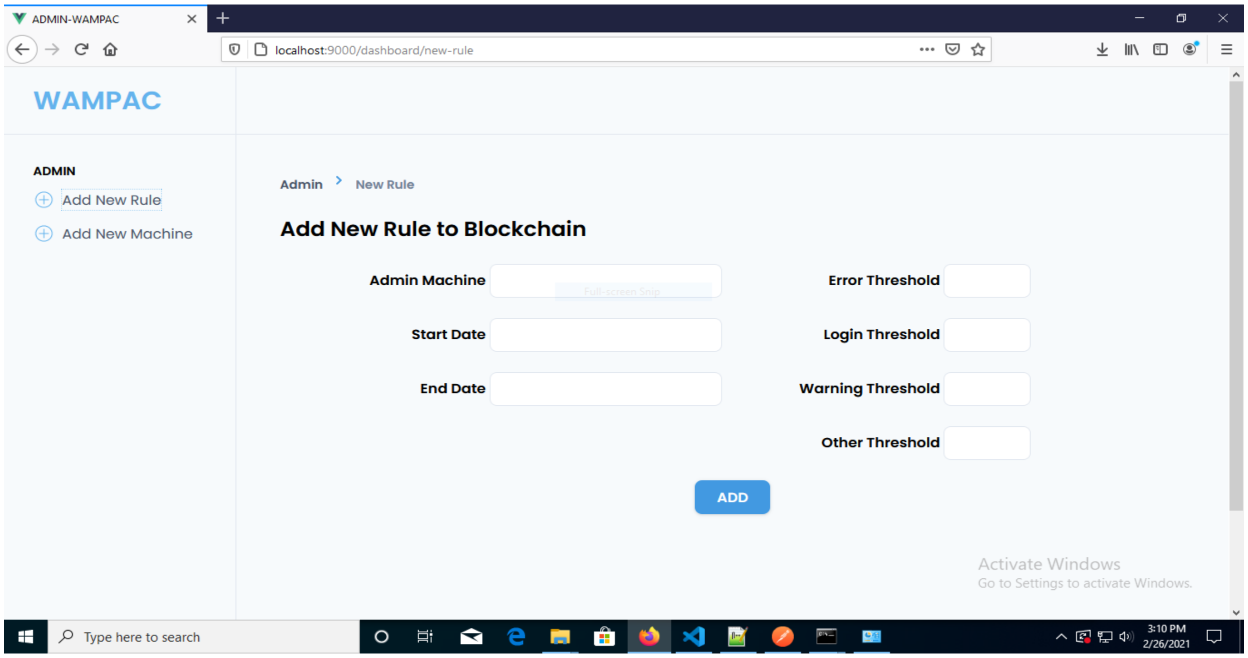
There are 3 types of nodes in the network

1. Server or management node
2. Verifier node
3. User node

This section will explain the application flow with respect to each of these nodes.

1. Server node

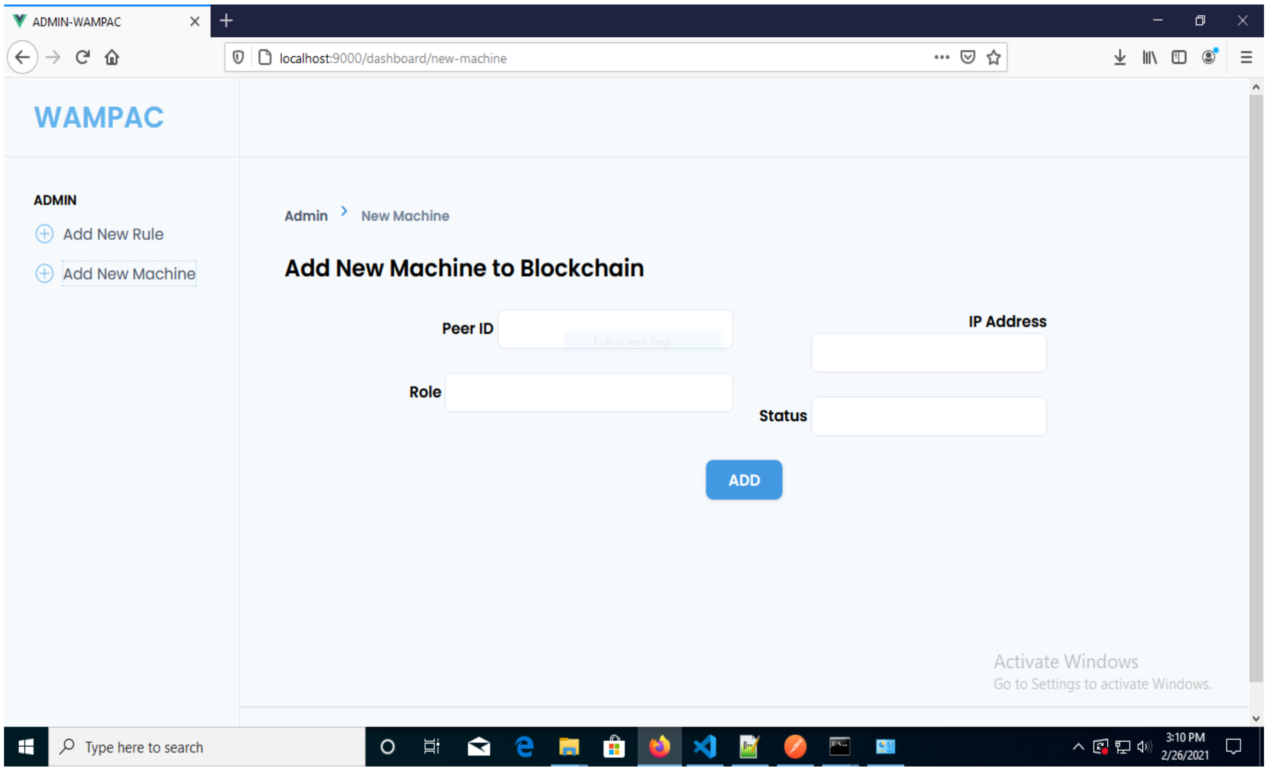
a) Creating rules to the network



Add New Rule to Blockchain

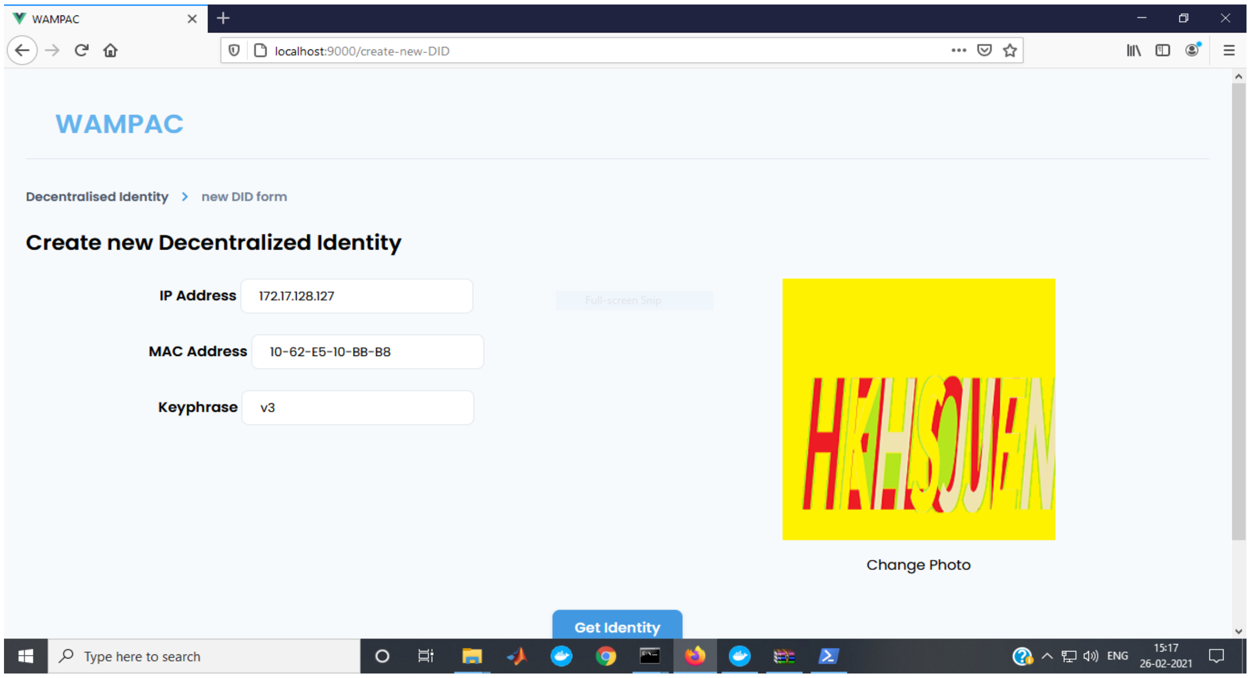
Admin Machine		Error Threshold	
Start Date		Login Threshold	
End Date		Warning Threshold	
		Other Threshold	

b) Whitelisting network nodes with role (Verifiers and Users)



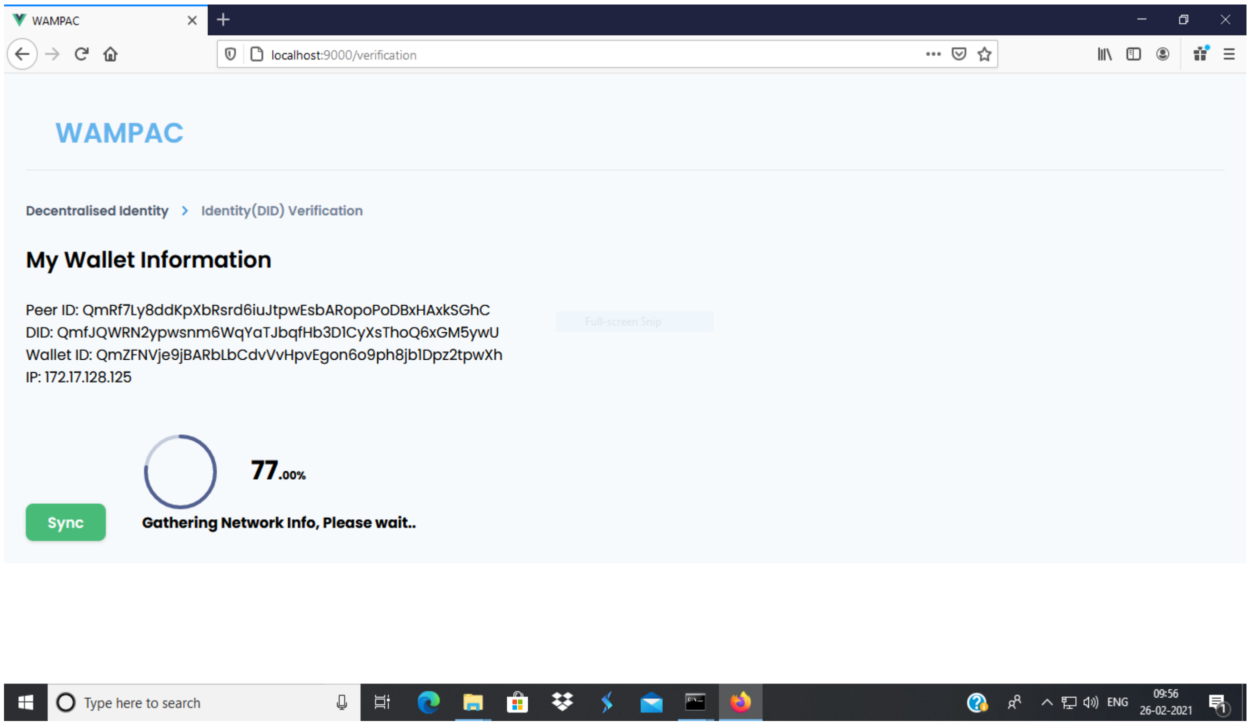
2. Node Enrollment Process

a) Creating Decentralized Identity


WAMPAC
localhost:9000/create-new-DID
WAMPAC
Decentralised Identity > new DID form
Create new Decentralized Identity
Change Photo
Get Identity
Type here to search
ENG
15:17
26-02-2021

IP Address	172.17.128.127
MAC Address	10-62-E5-10-BB-B8
Keyphrase	v3

b) Synchronizing to the network


WAMPAC

Decentralised Identity > Identity(DID) Verification

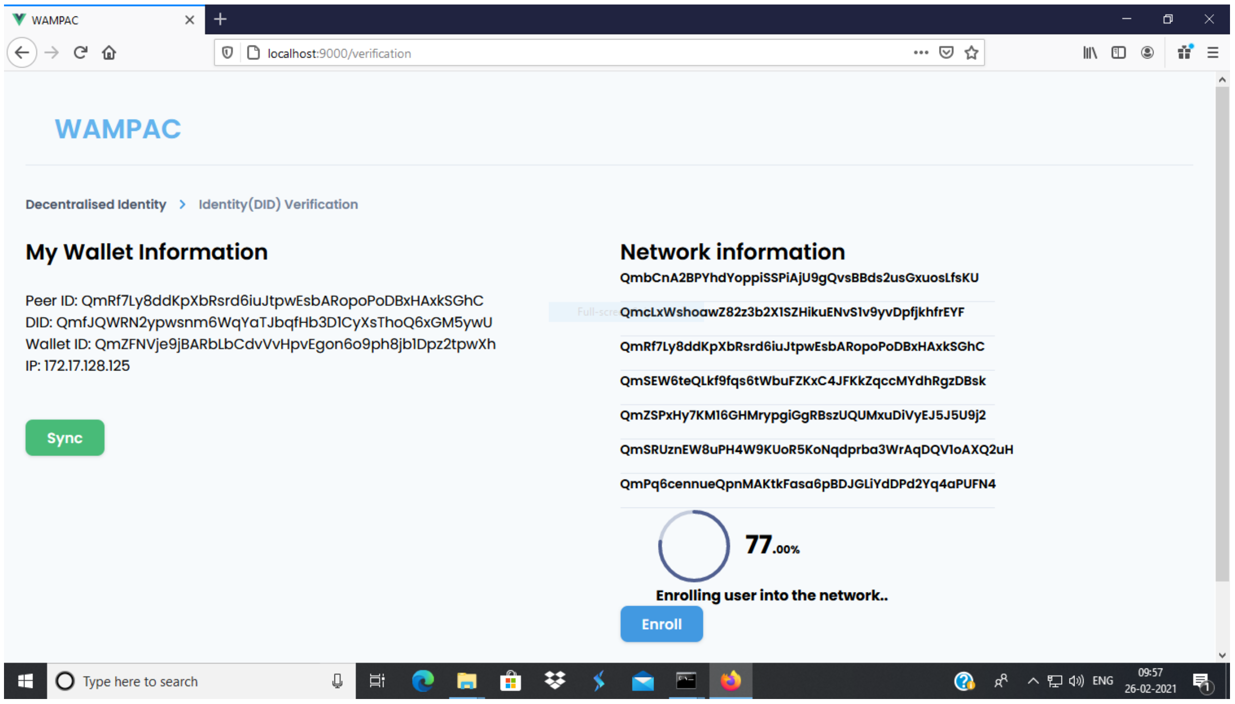
My Wallet Information

Peer ID: QmRf7Ly8ddKpXbRsrd6iuJtpwEsbARopoPoDBxHAxkSGhC
DID: QmfJQWRN2ypwsnm6WqYaTJbqfHb3DlCyXsThoQ6xGM5ywU
Wallet ID: QmZFNVje9jBARbLbCdvVvHpvEgon6o9ph8jblDpz2tpwXh
IP: 172.17.128.125

77.00%

Gathering Network Info, Please wait..

c) Enrollment (Signature verification)


WAMPAC

Decentralised Identity > Identity(DID) Verification

My Wallet Information

Peer ID: QmRf7Ly8ddKpXbRsrd6iuJtpwEsbARopoPoDBxHAxkSGhC
DID: QmfJQWRN2ypwsnm6WqYaTJbqfHb3DlCyXsThoQ6xGM5ywU
Wallet ID: QmZFNVje9jBARbLbCdvVvHpvEgon6o9ph8jblDpz2tpwXh
IP: 172.17.128.125

Network information

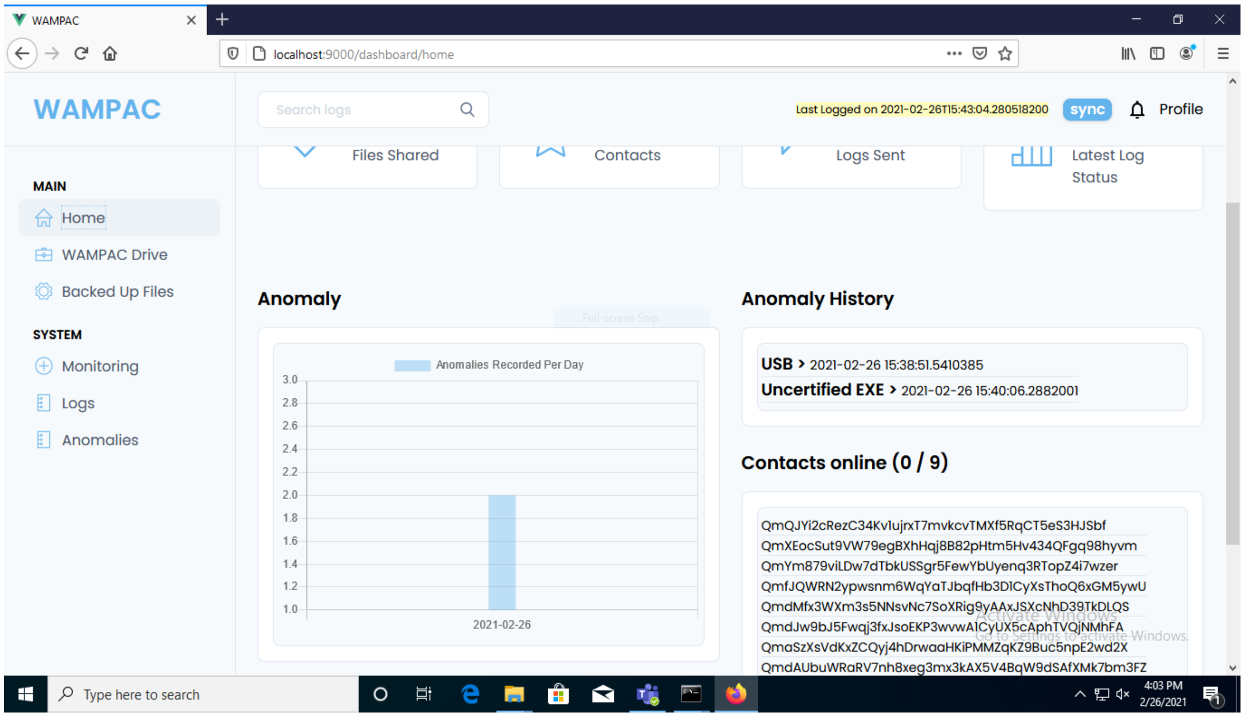
QmbCnA2BPYhdYoppiSSPiAjU9gQvsBBds2usGxuosLfsKU
QmcLxWshoawZ82z3b2X1SZHikuENvS1v9yvDpfjkhfrEYF
QmRf7Ly8ddKpXbRsrd6iuJtpwEsbARopoPoDBxHAxkSGhC
QmSEW6teQLkf9fqs6tWbuFZKxC4JFKkZqccMYdhRgzDBsk
QmZSPxHy7KM16GHMrypgiGgRBszUQUMxuDiVyEJ5J5U9j2
QmSRUznEW8uPH4W9KUoR5KoNqdprba3WrAqDQV1oAXQ2uH
QmPq6cennueQpnMAKtkFasa6pBDJGLiYdDPd2Yq4aPUFN4

77.00%

Enrolling user into the network..

3. User Node - Interface

a) Main dashboard

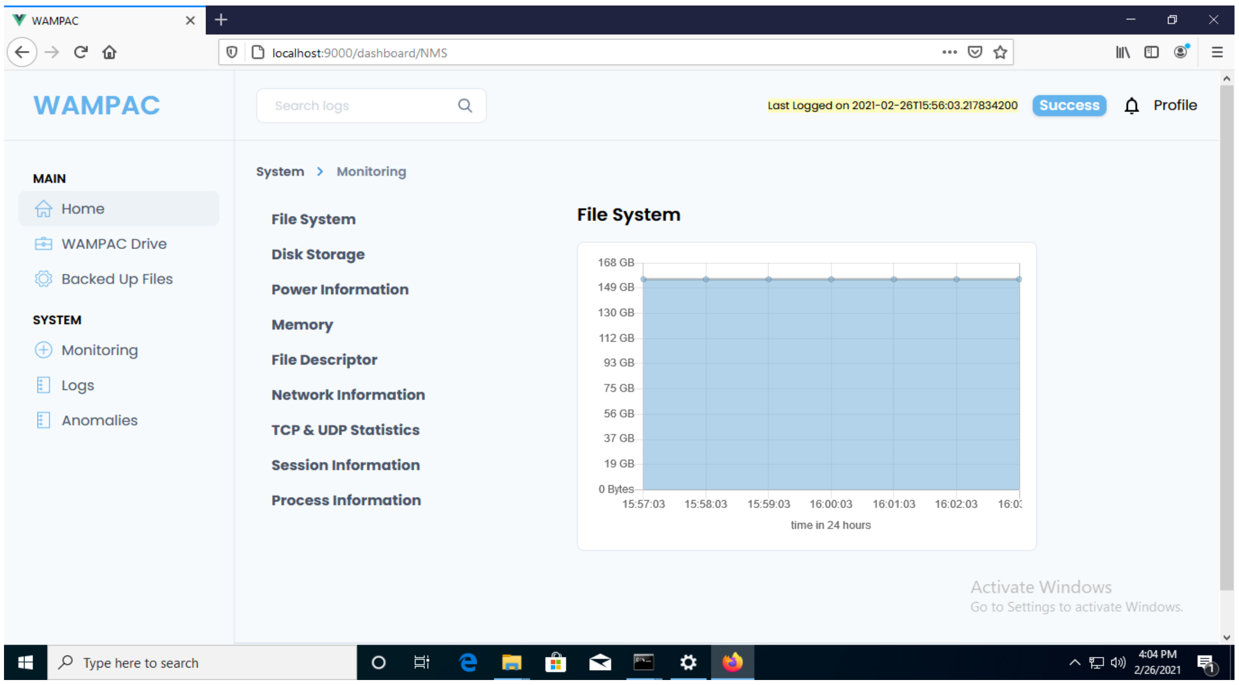


USB > 2021-02-26 15:38:51.5410385
Uncertified EXE > 2021-02-26 15:40:06.2882001

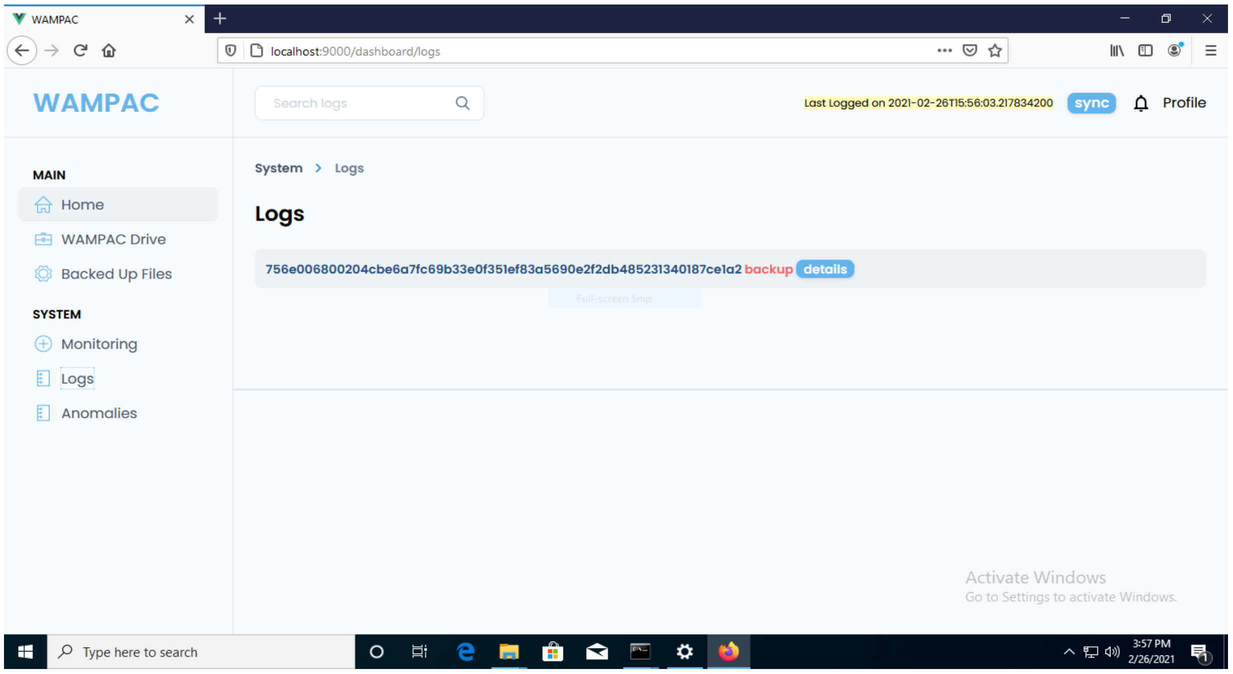
Contacts online (0 / 9)

QmQJYi2cRezC34Kv1ujrxT7mvkcvTMXf5RqCT5eS3HJSbf
QmXEocSut9VW79egBXhHqj8B82pHtm5Hv434QFgq98hyvm
QmYm879viLDw7dTbkUSSgr5FewYbUyenq3RTopZ4i7wzer
QmfJQWRN2ypwsnm6WqYaTJbqfHb3D1CyXsThoQ6xGM5ywU
QmdMfx3WXm3s5NNsvNc7SoXRig9yAAxJSXcNhD39TkDLQS
QmdJw9bJ5Fwqj3fxJsoEKP3wvwA1CyUX5cAphTVQjNMhFA
QmaSzXsVdKxZCQyj4hDrwaaHKiPMMZqKZ9Buc5npE2wd2X
QmdAUbuWRaRV7nh8xeg3mx3kAX5V4BqW9dSAfXMk7bm3FZ

b) Monitoring service

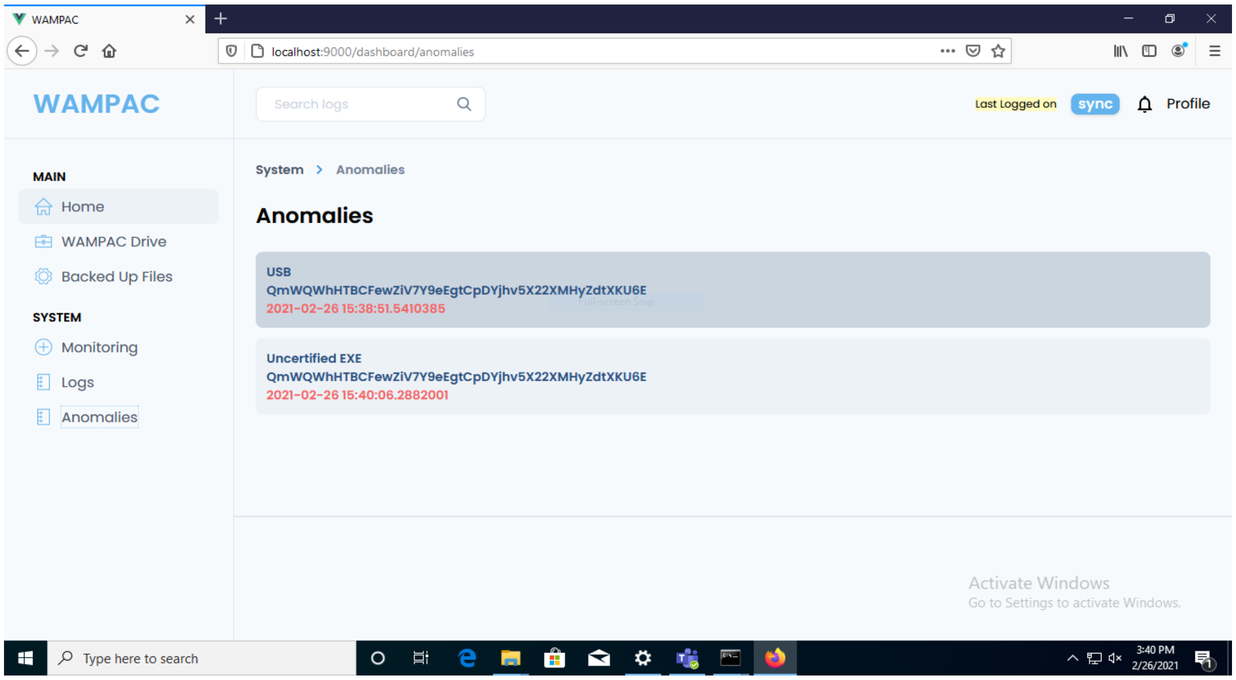


c) Event Logs



```
756e006800204cbe6a7fc69b33e0f351ef83a5690e2f2db485231340187ce1a2 backup details
```

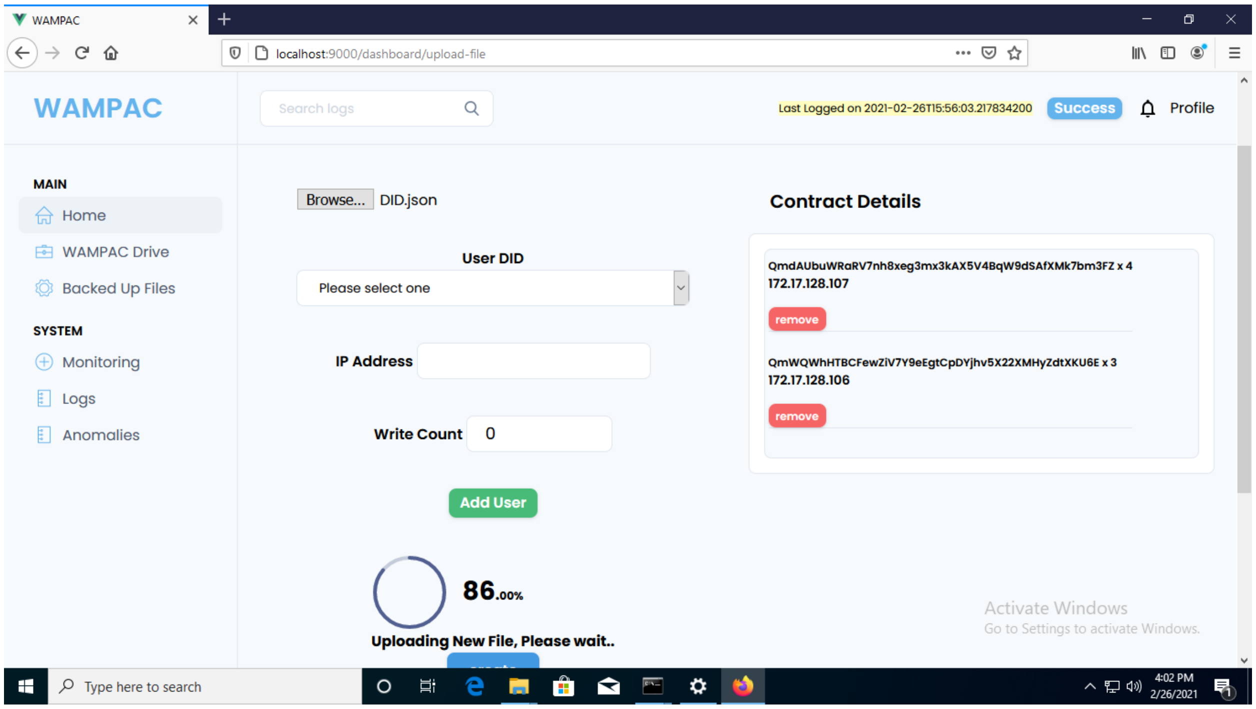
d) Anomalies



```
USB
QmWQWhHTBCFewZiV7Y9eEgtCpDYjhv5X22XMHyZdtXKU6E
2021-02-26 15:38:51.5410385

Uncertified EXE
QmWQWhHTBCFewZiV7Y9eEgtCpDYjhv5X22XMHyZdtXKU6E
2021-02-26 15:40:06.2882001
```

e) WAMPAC drive



Browse... DID.json

User DID
Please select one

IP Address

Write Count 0

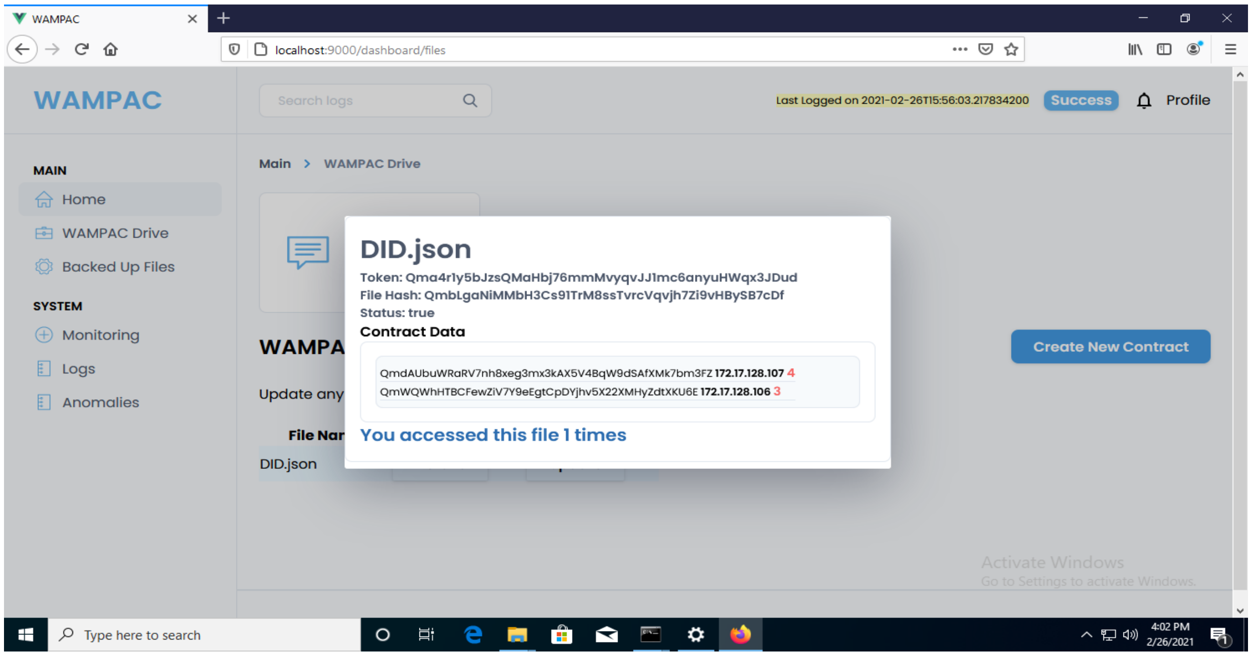
86.00%

Uploading New File, Please wait..

Contract Details

QmdAUbuWRaRV7nh8xeg3mx3kAX5V4BqW9dSAfXMk7bm3FZ x 4
172.17.128.107
remove

QmWQWhHTBCFewZiV7Y9eEgtCpDYjhv5X22XMHyZdtXKU6E x 3
172.17.128.106
remove



DID.json

Token: Qma4r1y5bJzsQMaHbj76mmMvyqvJJ1mc6anyuHWqx3JDud
File Hash: QmbLgaNiMMbH3Cs91TrM8ssTvrcVqvjh7Zi9vHBySB7cDf
Status: true

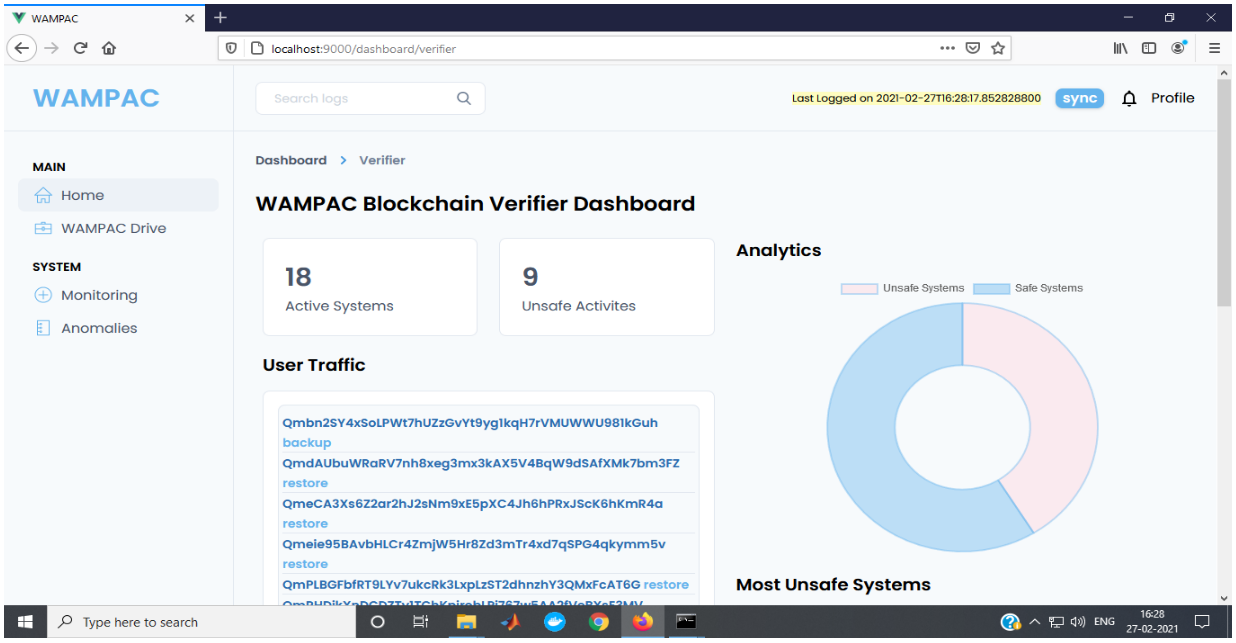
Contract Data

QmdAUbuWRaRV7nh8xeg3mx3kAX5V4BqW9dSAfXMk7bm3FZ **172.17.128.107** 4
QmWQWhHTBCFewZiV7Y9eEgtCpDYjhv5X22XMHyZdtXKU6E **172.17.128.106** 3

You accessed this file 1 times

4. Admin Node – Interface

a) Main dashboard



Last Logged on 2021-02-27T16:28:17.852828800

WAMPAC Blockchain Verifier Dashboard

18 Active Systems

9 Unsafe Activites

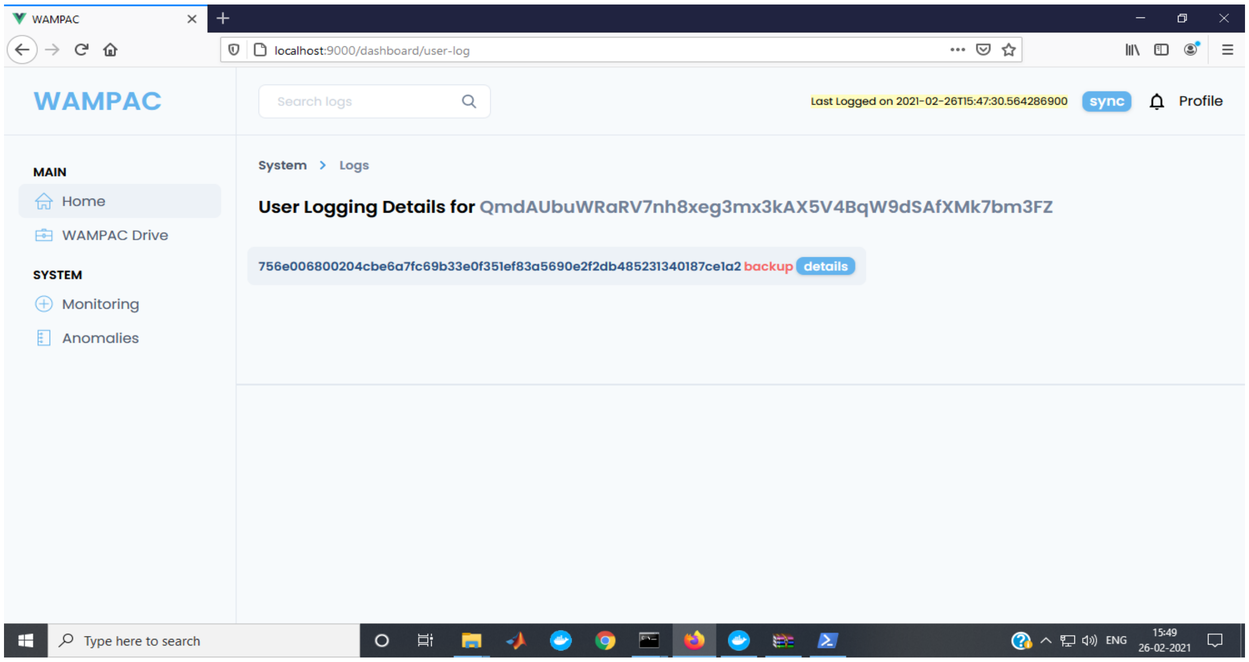
Analytics

User Traffic

```
Qmbn2SY4xSoLPWt7hUZzGvYt9yg1kqH7rVMUWWU981kGuh backup
QmdAUbuWRaRV7nh8xeg3mx3kAX5V4BqW9dSAfXMk7bm3FZ restore
QmeCA3Xs6Z2ar2hJ2sNm9xE5pXC4Jh6hPRxJScK6hKmR4a restore
Qmeie95BAvbHLCr4ZmjW5Hr8Zd3mTr4xd7qSPG4qkymm5v restore
QmPLBGFbfRT9LYv7ukcRk3LxpLzST2dhnzhY3QMxFcAT6G restore
```

Most Unsafe Systems

b) User Log details with status

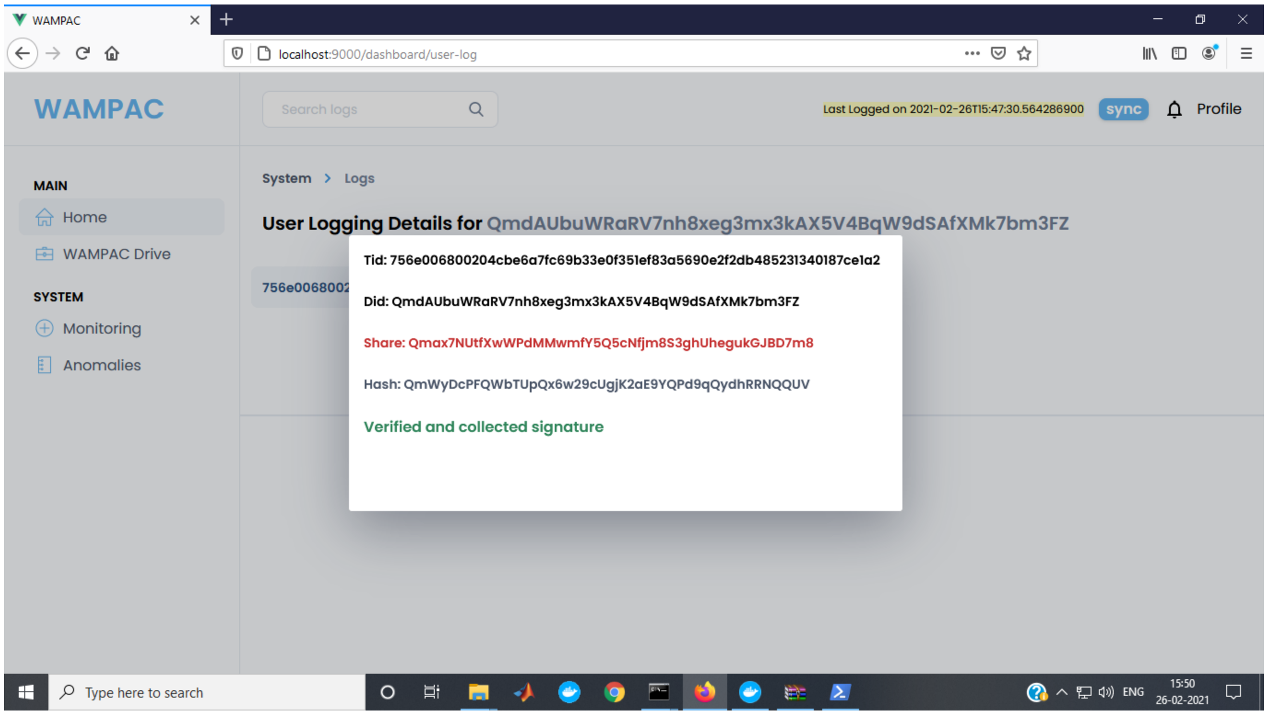


Last Logged on 2021-02-26T15:47:30.564286900

User Logging Details for QmdAUbuWRaRV7nh8xeg3mx3kAX5V4BqW9dSAfXMk7bm3FZ

```
756e006800204cbe6a7fc69b33e0f351ef83a5690e2f2db485231340187ce1a2 backup details
```

c) Log transaction details



Last Logged on 2021-02-26T15:47:30.564286900

User Logging Details for QmdAUbuWRaRV7nh8xeg3mx3kAX5V4BqW9dSAfXMk7bm3FZ

Tid: 756e006800204cbe6a7fc69b33e0f351ef83a5690e2f2db485231340187ce1a2

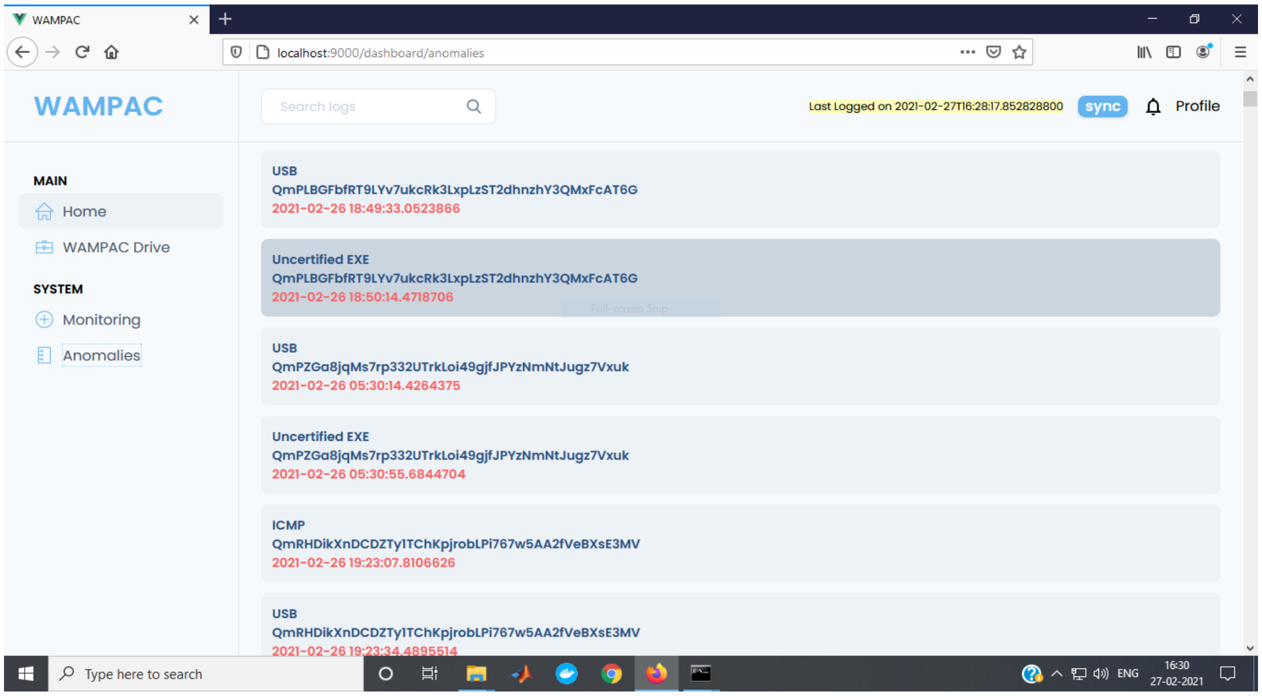
Did: QmdAUbuWRaRV7nh8xeg3mx3kAX5V4BqW9dSAfXMk7bm3FZ

Share: Qmax7NUtfXwWPdMMwmfY5Q5cNfjm8S3ghUhegukGJBD7m8

Hash: QmWyDcPFQWbTUpQx6w29cUgjK2aE9YQPd9qQydhRRNQQUV

Verified and collected signature

d) User anomalies



Last Logged on 2021-02-27T16:28:17.852828800

USB
QmPLBGFbfRT9LYv7ukcRk3LxpLzST2dhnzhY3QMxFcAT6G
2021-02-26 18:49:33.0523866

Uncertified EXE
QmPLBGFbfRT9LYv7ukcRk3LxpLzST2dhnzhY3QMxFcAT6G
2021-02-26 18:50:14.4718706

USB
QmPZGa8jqMs7rp332UTrkLoi49gjfJPYzNmNtJugz7Vxuk
2021-02-26 05:30:14.4264375

Uncertified EXE
QmPZGa8jqMs7rp332UTrkLoi49gjfJPYzNmNtJugz7Vxuk
2021-02-26 05:30:55.6844704

ICMP
QmRHDikXnDCDZTy1TChKpjrobLPi767w5AA2fVeBXsE3MV
2021-02-26 19:23:07.8106626

USB
QmRHDikXnDCDZTy1TChKpjrobLPi767w5AA2fVeBXsE3MV
2021-02-26 19:23:34.4895514

5. References

1. Zyskind, Guy, and Oz Nathan. "Decentralizing privacy: Using blockchain to protect personal data." 2015 IEEE Security and Privacy Workshops. IEEE, 2015.
2. Baars, D. S. Towards self-sovereign identity using blockchain technology. MS thesis. University of Twente, 2016.
3. Goldreich, Oded. "Secure multi-party computation." Manuscript. Preliminary version 78 (1998).
4. Cramer, Ronald, Ivan Damgård, and Ueli Maurer. "General secure multi-party computation from any linear secret-sharing scheme." International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, 2000.
5. Praveen, K 2019, 'Novel approaches for constructing efficient visual cryptographic schemes', PhD thesis, Amrita Vishwa Vidyapeetham, Coimbatore, India
6. Sukhwani, Harish, et al. "Performance modeling of PBFT consensus process for permissioned blockchain network (Hyperledger fabric)." 2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS). IEEE, 2017.
7. Castro, Miguel, and Barbara Liskov. "Practical byzantine fault tolerance." *OSDI*. Vol. 99. No. 1999. 1999.
8. Apap, Frank, et al. "Detecting malicious software by monitoring anomalous windows registry accesses." International Workshop on Recent Advances in Intrusion Detection. Springer, Berlin, Heidelberg, 2002.
9. Dwyer, John, and Traian Marius Truta. "Finding anomalies in windows event logs using standard deviation." 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing. IEEE, 2013.
10. Ligh, Michael Hale, et al. The art of memory forensics: detecting malware and threats in windows, linux, and Mac memory. John Wiley & Sons, 2014.
11. Meng, Weizhi, et al. "When intrusion detection meets blockchain technology: a review." Ieee Access 6 (2018): 10179-10188.
12. Alexopoulos, Nikolaos, et al. "Towards blockchain-based collaborative intrusion detection systems." International Conference on Critical Information Infrastructures Security. Springer, Cham, 2017.

13. Wang, Shangping, Yinglong Zhang, and Yaling Zhang. "A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems." Ieee Access 6 (2018): 38437-38450.
14. Shafagh, Hossein, et al. "Towards blockchain-based auditable storage and sharing of IoT data." Proceedings of the 2017 on Cloud Computing Security Workshop. 2017.
15. Benet, Juan. "Ipfs-content addressed, versioned, p2p file system." arXiv preprint arXiv:1407.3561 (2014).