# Behavioural Based Threat Modelling to Increase the Efficiency in Breach Identification and Notification

MSc Internship
Cybersecurity

Joshua Nevilraj Yuva Kumar
Student ID: X18128106

School of Computing
National College of Ireland

Supervisor: Imran Khan

## National College of Ireland

## MSc Project Submission Sheet

## School of Computing

**Student Name:** Joshua Nevilraj Yuva Kumar…………………………………………………………………

**Student ID:** X18128106

**Programme:** MSc Cybersecurity **Year:** 2018-2019

**Module:** Academic Internship

**Supervisor:** Imran Khan

**Submission Due Date:** 12 December 2019

**Project Title:** Behavioural Based Threat Modelling To Increase The Efficiency In Breach Identification and Notification.

**Word Count:** 5579 **Page Count:** 19

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:** ……………………………………………………………………………………………………………

**Date:** ……………………………………………………………………………………………………………

**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: | |
| Penalty Applied (if applicable): | |

# Behavioural Based Threat Modelling to Increase the Efficiency in Breach Identification and Notification

Joshua Nevilraj Yuva Kumar
X18128106

**Abstract**

In a rapidly changing cyber threat landscape it has become a challenging task for organisations defend against sophisticated cyber-attacks. With new regulations such as GDPR (General Data Protection Regulation) in effect, organizations had to make sure they maintain adequate measures and security controls in place. This research paper is focused on how to utilize the ability of security operations centre (SOC) effectively with the implementation of behavioural based threat model to detect the presence of adversaries in the network by identifying their behaviour. This threat model's objective goal is to save organisations from huge amount of fine and reputation loss due to cyber-attacks by APTs.

## 1 INTRODUCTION

Cyber-attacks have become everyday news in the recent years impacting Organisations of all size. These attacks have had devastating effects on individuals as well as Organizations, Financial Institutions, Critical Infrastructures, Healthcare providers, Airline services, Educational bodies and even locked down Government agencies and Local body communities. The recent ransomware attack named "WannaCry" crippled thousands and hundreds of computers and the costs to the affected Organisations is estimated around £92 million.

According to The U.S. Conference of Mayors, Ransomware attacks have hit at least 170 county, city, or state government systems since 2013, and 22 of those attacks occurred in the first half of 2019. Below listed is a sample of ransomware attacks that have hit specific cities, towns and government organizations.

- June 20, 2019: Riviera Beach, Florida, discloses ransomware attack and payment.
- May 7, 2019: City of Baltimore hit with ransomware attack.
- April 2019: Cleveland Hopkins International Airport suffered a ransomware attack.
- April 2019: Augusta, Maine, suffered a highly targeted malware attack that froze the city's entire network and forced the city centre to close.
- April 2019: Hackers stole roughly $498,000 from the city of Tallahassee.
- March 2019: Albany, New York, suffered a ransomware attack.
- March 2019: Jackson County, Georgia officials paid cybercriminals $400,000 after a cyberattack shut down the county's computer systems.
- March 2018: Atlanta, Georgia suffered a major ransomware attack.
- February 2018: Colorado Department of Transportation (CDOT) employee computers temporarily were shut down due to a SamSam ransomware virus cyberattack.

Rapid change in the cyber threat landscape and the increasing sophistication and ease of launching a cyber-attack have pushed Organisations to new security models, computer defence strategies and architectures like "Assume Breach" and "Zero Trust". There is a paradigm shift from a prevention mind-set to building agility to quickly detect and respond when there is a cyber-attack or a breach.

The cyber awakening is also due to many new regulations that have come into effect, like the General Data Protection Regulation (GDPR), that requires organisations to ensure adequate organisational measures and security controls are maintained, while also mandating timely notifications should there be a security breach.

Any Cyber-attack or intrusion attempt leaves a trace of digital footprints. In case of a breach, Organisations can follow the tracks of these digital footprints to identify and analyse the impact of the breach. To address these security risk and to meet regulatory compliance Organisations have turned to security operation centres (SOC). SOC is a facility where enterprise information systems (web sites, applications, databases, data centres and servers, networks, desktops and other endpoints) are monitored, assessed, and defended. SOCs typically are based around a security information and event management (SIEM) system.

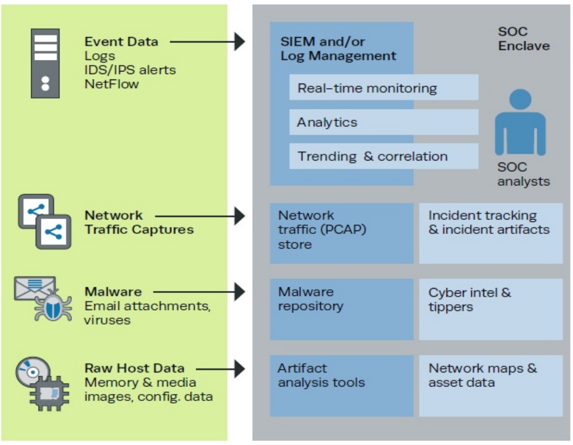


Figure - 1 SOC Architecture

Security Information and Event Management tool is an approach of combining event management and information management into a security management system. The SIEM technology creates a "single pane of glass" for the security analysts to monitor the enterprise. Real-time tracking and proactive monitoring of digital footprints can help security analysts and Organisations to prevent any possible cyber-attacks. Thus, using these indicators of compromise, the security analysts can analyse the cyber-attack and also take preventive steps to protect the system or network from similar attacks in the future.

Certain predefined criteria are used to evaluated incidents which appear to be a suspicious event in order to take further action to mitigate them and restoring the computational resources to normal state. The events that are anomalous will be automatically detected and reported by local event monitoring which is the principle source for incident discoveries. Search for sign of specific attack patterns that are already known is an alternative to local event logging because if the attack is shifted to other systems it can easily lose track of the incident and completely misses it. Hence automated tools are employed by most of the Organisations to detect and report security incidents automatically that would help them reduce the time frame between incident remediation and the discovery of attack. Sometimes external entities like Computer Emergence Response Teams (CERTs) and Internet Service Providers (ISPs) will notify data breach to the organizations. In this research we address the issues and challenges faced by the organizations in identifying the threats posed by the evolving APTs. And how to effectively utilize the abilities of SOC with the behavioural based threat model in order to identify the breach in limited time frame.

## 1.1 Breach notification in General Data Protection Regulation

The notification requirements have been extended beyond the electronic communication sector by the GDPR European legislators while considering that "a personal data breach may, if not addressed in an appropriate and timely manner, result in physical, material or non-material damage to natural persons …." (Recital 85). According to GDPR "a personal data breach" is defined as "a breach of security leading to accidental or unlawful destruction, loss, alteration, unauthorised disclosure of, or access to personal data transmitted, stored or otherwise processed".

According to General Data Protection Regulation the data breach has to be notified to the legal or supervisory authorities without undue delay not later than 72 hours after becoming aware of the breach (Recital 87). The notification should include the nature of the breach and where possible, number of data subjects that are concerned approximately, number of the personal data records that are approximately concerned, the categories, the consequence of the breach and the proposed measures that are to be taken to address the data breach along with recommendations for the data subject to mitigate the adverse effect (Recital 86). It is not required to communicate with the concerned data subject if the appropriate technological measures were implemented by the controller for the purpose of protecting the personal data and those measures were affected by the personal data breach.

## 1.2 Issues and challenges

Since the notification of data breach has become mandatory, the Organisations find it very expensive as they have to organize services for customers in occurrence of security incidents, potential fines for non-compliance with the security obligation along with the legal fees and the reputation harm which leads to a significant loss of market value. From the security management point of view there are challenges in assessing the impacts of data protection, designing the privacy requirements, to address and manage the security issues in advance, performance in design processing phase and risk assessments that can lead to possible last minute rush along with confusion on determining which risks has to be reported when the data breach occurs. Another major challenge is to determine the criteria of the data breach to characterize the events as they vary according to countries and organization depending upon the distinction and understanding of the security related events and incidents.

# 2 RELATED WORK

According to author Cláudio Toshio Kawakani who proposes a hierarchical clustering that uses historical alerts which identifies a typical strategy that were used in previous cyber-attacks and correlate them with the real time alerts which would help to reduce the time between response and security event. The proposed approach was to create two correlators where one would be offline and other would be online, so the offline correlator uses IDS alerts as the input and creates an attack model strategy in cluster which can be used in the online correlator. This method is very helpful in reducing the time between detecting the event and response but the constraint it has is that we will not be able to know the exact behaviour of the adversary which had breached the network. (Cláudio Toshio Kawakani, 2016)

Author Gustavo Gonzalez in his research has proposed two new approach on alert correlation techniques in security information and event management systems. The first approach is based on enforcing policies and defending capability model whereas the second approach is based on the indicators of compromise. So, the first approach is focused on classifying the countermeasures and implement defence mechanism by implementing the security rules that are appropriate for the classified countermeasures. The second approach is a metric-based approach that derives correlation rules from the information security indicators to evaluate the effectiveness of the Security information and event management system. These approaches are very helpful in defending the attacks against web applications (Gonzalez Granadillo, El-Barbori and Debar, 2016).

Another research done by Xindai Lu and JiaJia Han in September 2018 on Network threat detection on correlation analysis of multi-platform multi-source alert data proposes a algorithm called PMASP (Purpose-oriented Maximum Attack Sequence Pattern) for threat detection which is based on correlation of alerts. They use clustering to generate the attack sequence and then deploy the PMASP algorithm to find frequent attack sequence. Finally, they construct the rules through XML language which will help administrators to detect the attacks among the large number of false positive (Lu et al., 2018).
In 2015 Neelam Dwivedi and Aprna Tripathi did a research on event correlation for intrusion detection system. The aim of this research was to improve security by correlating similar alerts together to reduce the workload and time for the analysts who were analysing hundreds and thousands of alerts. The alerts that were generated by different IP addresses with the same name are correlated which would reduce the amount of alerts an analyst has to verify and analyse. The drawbacks of this research were erroneous design and insufficient requirements which made it hard for testing. Even though it reduced the number of alerts the analyst would have to deal with it was not sufficient (Dwivedi and Tripathi, 2015).

A research done in 2016 on Security operation centers for Information security Incident management states that Information security management system can be implemented in security operations center to rapidly detect incidents and minimize the loss and destructions to the organizations. There are number of related researches that describes the techniques and framework for planning, implementation, assessment and improvement of Information Security Incident Management Process (Killcrece G., 2019). This research mainly focuses on addressing the vulnerabilities in Internet of Things exploit. Instead of traditional security monitoring centre they have proposed a concept of security intelligence centre for responding to events occurring in Information security with regards to Internet of Things. And it also provides the inter-relation of Information Security Incident Management Process and Information Security monitoring.

Event analysis for Security Incident Management on Perimeter Access Control System research from Russia gives us a general approach to security events based on the proposed proactivity, dynamism and multidimensionality principles (V. Desnitsky and I. Kotenko, 2016) [10]. Similar to the previous research review this research also proposes a unique way to respond to events with regards to information security in Internet of Things environment. Automatic mechanism assumed by the proactivity of management uses statistical data in the events of the system (|I.V. Kotenko, V.V. Vorontsov, A.A. Chechulin, A.V. Ulanov, 2009). These data are collected from various security devices such as RFID scanners and monitoring the network activity in both user workstation and intermediate routers which combined and used for modelling the actions of the intruder. This process is carried out but by correlating various security events to increase the efficiency of the incident management system.

Dynamic approach to Cyber-Incident response research portraits to allow a low-value target to get compromised instead of defending it to learn the methods and techniques of the adversary, where the learnt information can be used defend a high-value target (K. Mepham, 2014). This research has been done by comparing various methodology used for incident response from the early stages which were a static approach and basically addressing the short coming of traditional Cyber incident response. There are four factors collaboration, sensor, information credibility and incident discrimination are used to develop a model that is processed by intelligence gathering, static impact evaluation, dynamic risk and value assessment, modelling and decision. And there-by developing a dynamic model or methodology so that the cyber-incident response key decision maker can make to defend against the incidents take place over the targeted environments.

Prerequisites for building a computer security incident response capability is research done to investigate the requirement for business prior to establishing a Computer Security Incident Response Team (CSIRT). It is a structured review of literature to understand the challenges in establishing a CSIRT. The current issues in establishing an efficient Computer Security Incident Response team are unclear mandate, revenue model selection, interacting with external parties, etc. To establish a CSIRT team the following factors are required i.e setting up the environment, forming a constituency, authority, funding and legal considerations A concept matrix was utilized to correlate the deficiency and the challenges to meet the business requirement for establishing a Computer Security Incident Response or a similar Team (R. Mooi and R. Botha, 2015).

Research on Detection and Handling of Security Incidents and Perimeter Breach based on flexible honeytoken framework suggests that peri-meter based technologies in intrusion detection can be supported by using deception systems. A framework is implemented with different type of modules such as sensing, generation, monitoring and countermeasure that is completely suited for detecting breaches at the peri-meter level (Karyda, Maria and Mitrou, 2016). Although the framework is extensible with deceptive token modules such as honeyfiles, honeylinks and honeypots, it has to enable a set of supportive mechanism such as firewalls and access control since deceptive defenses are not stand-alone solutions for security.

Security incident and event management in cloud computing infrastructure is research that provides analysis and approaches of SIEM in terms of technical requirements, logical and legal framework. And gives us a suggestion on how to deploy a security incident and event management system in a cloud environment.  Although the issues with dynamic scaling can be addressed there is a problem with multi-tenancy when we want to have security which should be written with the application for various customers. The shortcomings of this research are the interface and security issues with the shared resources.

# 3 RESEARCH METHODOLOGY

In current enterprise network it is very dubious for Organisations to have the resources and ability to defend against each method an adversary uses to gain access to the network. Even when the patching and compliance of software is up to date in an organization a zero-day exploit or a social engineering attack from an advance persistent threat can gain access to the victim's network.

For network compromise detection we have an approach that uses behavioural methodology and it is a threat-based approach which is directed by five main principles. These principles are shown in the figure 2 (Strom et al., n.d., 2017).

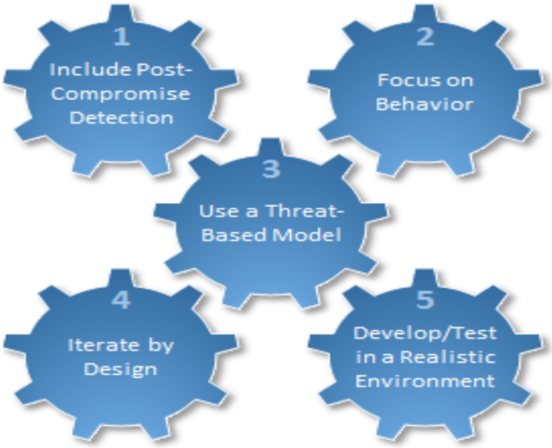

1
Include Post-Compromise Detection
2
Focus on Behavior
3
Use a Threat-Based Model
4
Iterate by Design
5
Develop/Test in a Realistic Environment

Figure - 2 *Principles of threat-based security*

*Include Post-compromise detection*:

Even the most well-defended network perimeter can be penetrated by advance persistent threats because there are no effective ways to defend zero-day exploits, no methods for instant software patching and preventing every human from exposing passwords is impossible. That's why post-compromise behaviour of the adversaries should be taken into account to minimize the damage caused.

*Focus on Behaviour*:

Most network defenders focus on the signatures and indicators of compromise (IOCs) that are observed during the presence of known malware activity. IOCs are more likely the IP addresses, file hashes and domain names but the adversaries will often change their indicators in order to hide from detection. This is why we need to focus on behaviour-based detection rather than signature-based detection because adversaries can modify the indicators to avoid signature-based detection. Many adversary groups will follow a common behaviour during an intrusion hence behaviour-based detection will be an appropriate approach in identifying the threat imposed by the adversaries.

*Use a threat model*:
To evaluate and plan the defence we use a threat model which is compiled of behaviours and actions of the adversaries. Figure 3 (Strom et al., n.d., 2017) shows how we take three post-compromise stages in the cyber-attack lifecycle and split them into 10 tactics which are commonly used by the APTs.

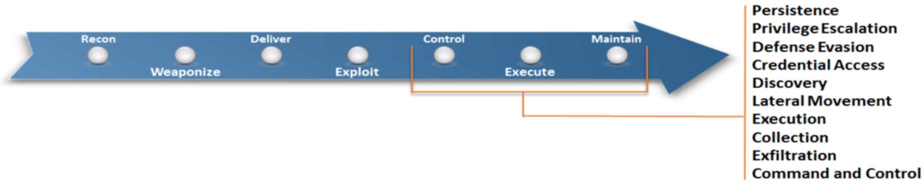


Figure – 3 *Categories of Tactics*

*Iterate by Design*:
For frequently changing threat landscape, iteration will give the ability to defend the networks. Dealing with APTs that change their behaviour to avoid detection, behavioural detection approach will be the most effective method because to bypass network defence such as blacklisting and protocol signature detection APTs will use strong encryption and web services that are legitimate but when interacting with the endpoints they have to follow a consistent way because that is how the systems operate. As the adversaries still keep changing their behaviours to avoid detection it is essential to have iterative approach for effective defending for the defensive capabilities to the latest behaviours.

*Test in realistic environment*:
In this framework it is important to develop detection capabilities and analytics that are iterative in a real-time or live production environment with real users and real background system noise. Because in a laboratory environment without real users performing their daily tasks without real background noise of the systems it would be useless to detect the behaviours of the APTs as it would be a routine of behaviours performed by system administrators or the users.

# 4 DESIGN SPECIFICATIONS

## 4.1 Post Compromise Threat Based Modelling

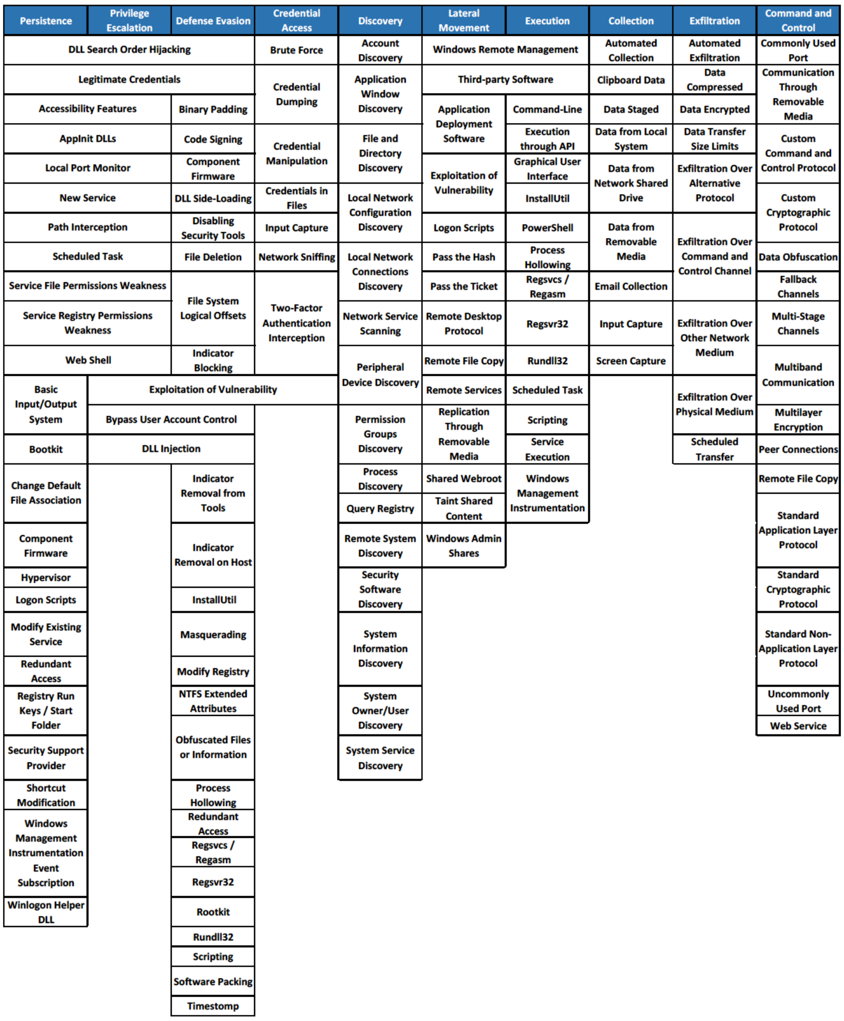
There are many public reports on the adversary groups and the behaviours of the adversaries but it does not provide us will high-level detail for example the report mentions an adversary has used lateral movement tactic but it doesn't give us the detail on how the adversary performed this tactic which would be more help for the security operators in the organization. This framework is developed mainly to address lack of knowledge on the behaviours of the adversaries, it contains the high-level details on the behaviours of the adversaries that is broken down into tactical categories which includes the techniques that were used to accomplish these tactics.

## 4.2 Tactics

They represent the tactical behaviours of the adversaries during their successful intrusion attempts. The categories of tactics are listed below

- **Persistence**: The actions or change of configuration in the systems that allows an adversary a persistent presence inside the system. This is done by loss of credentials, system restart or other failures. - ***TA0003***
- **Privilege Escalation**: This tactic includes the techniques used by the adversaries to obtain higher level permissions on a network or system. These permissions are required to perform certain operations which a normal user will not be allowed to perform. - ***TA0004***
- **Defense Evasion**: Adversaries use this tactic to avoid detection from the network defenders or evade these detections. - ***TA0005***
- **Credential Access**: Tactic which is used to gain access or control over domains, network or systems that are used within the organization. - ***TA0006***
- **Discovery**: This tactic includes techniques that are used by the adversaries to gain knowledge of the internal networks and the systems inside the organizations. - ***TA0007***
- **Lateral Movement**: The techniques that enables an adversary to control and access other systems inside the network through remote connection. - ***TA0008***
- **Execution**: The techniques used for execution of malicious code in the internal or a remote system. - ***TA0002***
- **Collection**: These techniques are used prior to exfiltration which includes gathering of information such sensitive files from the network that is targeted. - ***TA0009***
- **Exfiltration**: These are the techniques used to remove the sensitive information from the targeted network or system. Also, which location an adversary will look for this information. - ***TA0010***
- **Command and Control**: These explain how the communication is done by the adversaries with the systems that are under their control inside the targeted network. - ***TA0011***

These tactics represents the Adversary's lifecycle on how they operate from the initial access to command and control. They are indicated with unique IDs to represent them. Figure 4 (Strom et al., n.d., 2017) below will give an overview of the tactical categories that includes the techniques aligned under every category of the tactics in which they can be applied.



| Persistence | Privilege Escalation | Defense Evasion | Credential Access | Discovery | Lateral Movement | Execution | Collection | Exfiltration | Command and Control |
|---|---|---|---|---|---|---|---|---|---|
| DLL Search Order Hijacking | | | Brute Force | Account Discovery | Windows Remote Management | | Automated Collection | Automated Exfiltration | Commonly Used Port |
| Legitimate Credentials | | | Credential Dumping | Application Window Discovery | Third-party Software | | Clipboard Data | Data Compressed | Communication Through Removable Media |
| Accessibility Features | | Binary Padding | Credential Manipulation | File and Directory Discovery | Application Deployment Software | Command-Line | Data Staged | Data Encrypted | Custom Command and Control Protocol |
| AppInit DLLs | | Code Signing | Credentials in Files | Local Network Configuration Discovery | Exploitation of Vulnerability | Execution through API | Data from Local System | Data Transfer Size Limits | Custom Cryptographic Protocol |
| Local Port Monitor | | Component Firmware | Input Capture | Local Network Connections Discovery | Logon Scripts | Graphical User Interface | Data from Network Shared Drive | Exfiltration Over Alternative Protocol | Data Obfuscation |
| New Service | | DLL Side-Loading | Network Sniffing | Network Service Scanning | Pass the Hash | InstallUtil | Data from Removable Media | Exfiltration Over Command and Control Channel | Fallback Channels |
| Path Interception | | Disabling Security Tools | Two-Factor Authentication Interception | Peripheral Device Discovery | Pass the Ticket | PowerShell | Email Collection | Exfiltration Over Other Network Medium | Multi-Stage Channels |
| Scheduled Task | | File Deletion | | Permission Groups Discovery | Remote Desktop Protocol | Process Hollowing | Input Capture | Exfiltration Over Physical Medium | Multiband Communication |
| Service File Permissions Weakness | | File System Logical Offsets | | Process Discovery | Remote File Copy | Regsvcs / Regasm | Screen Capture | Scheduled Transfer | Multilayer Encryption |
| Service Registry Permissions Weakness | | Indicator Blocking | | Query Registry | Remote Services | Regsvr32 | | | Peer Connections |
| Web Shell | | | | Remote System Discovery | Replication Through Removable Media | Rundll32 | | | Remote File Copy |
| Basic Input/Output System | Exploitation of Vulnerability | | | Security Software Discovery | Shared Webroot | Scheduled Task | | | Standard Application Layer Protocol |
| Bootkit | Bypass User Account Control | | | System Information Discovery | Taint Shared Content | Scripting | | | Standard Cryptographic Protocol |
| Change Default File Association | DLL Injection | Indicator Removal from Tools | | System Owner/User Discovery | Windows Admin Shares | Service Execution | | | Standard Non-Application Layer Protocol |
| Component Firmware | | Indicator Removal on Host | | System Service Discovery | | Windows Management Instrumentation | | | Uncommonly Used Port |
| Hypervisor | | InstallUtil | | | | | | | Web Service |
| Logon Scripts | | Masquerading | | | | | | | |
| Modify Existing Service | | Modify Registry | | | | | | | |
| Redundant Access | | NTFS Extended Attributes | | | | | | | |
| Registry Run Keys / Start Folder | | Obfuscated Files or Information | | | | | | | |
| Security Support Provider | | Process Hollowing | | | | | | | |
| Shortcut Modification | | Redundant Access | | | | | | | |
| Windows Management Instrumentation Event Subscription | | Regsvcs / Regasm | | | | | | | |
| Winlogon Helper DLL | | Regsvr32 | | | | | | | |
| | | Rootkit | | | | | | | |
| | | Rundll32 | | | | | | | |
| | | Scripting | | | | | | | |
| | | Software Packing | | | | | | | |
| | | Timestomp | | | | | | | |

Figure. – 4 *Behavioural based threat model*

## 4.3 Techniques

Techniques are represented as the actions performed by the adversaries in order to obtain their tactical objectives. Like tactical IDs each technique has their IDs to represent each of them. There are number of actions categorized under the tactics to accomplish that tactical goal. Techniques are not performed in an individual occurrence normally they will follow a sequence of events or a set of behaviours. Figure 5 (Strom et al., n.d., 2017) shows how techniques are categorized and mapped with their respective tactic in the framework.

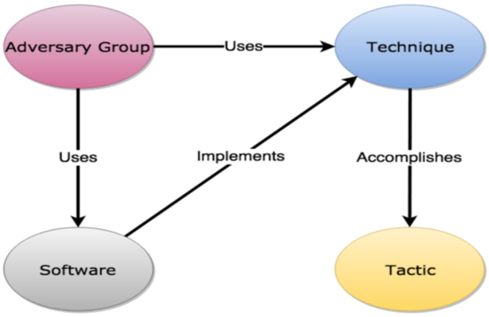


Figure – 5 *Mapping Techniques with Tactics*

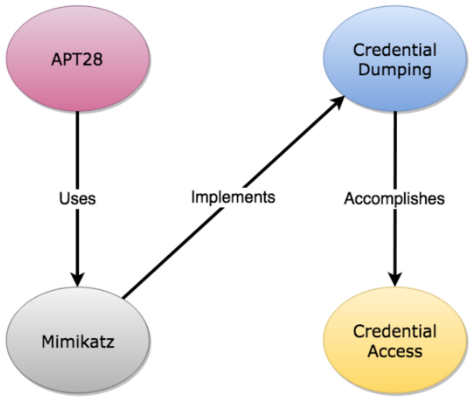


Figure – 6 *Example of Mapping Technique with Tactic*

## 4.4 Limitations

There is a major limitation in acquiring data sources like activity logs, process monitoring logs, API logs, process command line parameters, etc from a real-time (production) environment that are required for the identification of most of the techniques. This prototype is focused on techniques that uses data sources from web applications. The logs are generated manually using node.js and the techniques (suspicious activities) are identified by an application developed using JavaScript instead of a security information and event management system. This model is developed only till identification of certain techniques as we require real logs from real resources that are present in the real-time environment to portrait the accurate behaviours of the adversaries.

# 5 DATA COLLECTION

Each technique will require different types of data sources such as activity logs, process monitoring logs, API logs, process command line parameters and etc. Since there is a limitation in obtaining these types of logs from a real-time environment, simple helper function called write logs has been used to generate synthetic web-based logs.

In write logs we simply use file system (fs) module in node.js and read the previously wrote records and save that to a variable. Then the new logs are pushed and saved to the existing logs.

The attributes for these logs are:

TypeOfRequest - what type of request is made

Route - the actual route where the request is made

UserVerified - the requested user is verified.

TypeOfUser - what type of user

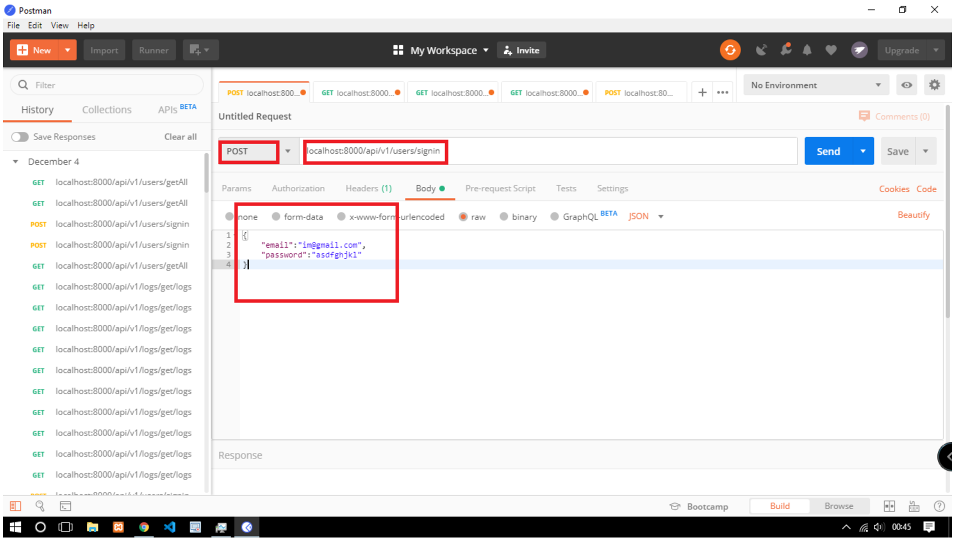
ValidToken - is the used token valid

UsersOldToken - has the user used his/her old token

CheckMadeFor - check made for the technique.

Postman is a tool used to send and receive API requests. To send the requests these three properties of the API route

- The route: eg:localhost:8000/api/v1/users/signin.
- The request type: Whether it is a GET, POST, PUT or any other method.
- The request Body: JSON object passed if the request is either a POST or PUT



December 4

GET localhost:8000/api/v1/users/getAll
GET localhost:8000/api/v1/users/getAll
POST localhost:8000/api/v1/users/signin
POST localhost:8000/api/v1/users/signin
GET localhost:8000/api/v1/users/getAll
GET localhost:8000/api/v1/logs/get/logs
GET localhost:8000/api/v1/logs/get/logs
GET localhost:8000/api/v1/logs/get/logs
GET localhost:8000/api/v1/logs/get/logs
GET localhost:8000/api/v1/logs/get/logs
GET localhost:8000/api/v1/logs/get/logs
GET localhost:8000/api/v1/logs/get/logs
GET localhost:8000/api/v1/logs/get/logs
GET localhost:8000/api/v1/logs/get/logs
GET localhost:8000/api/v1/logs/get/logs

Untitled Request

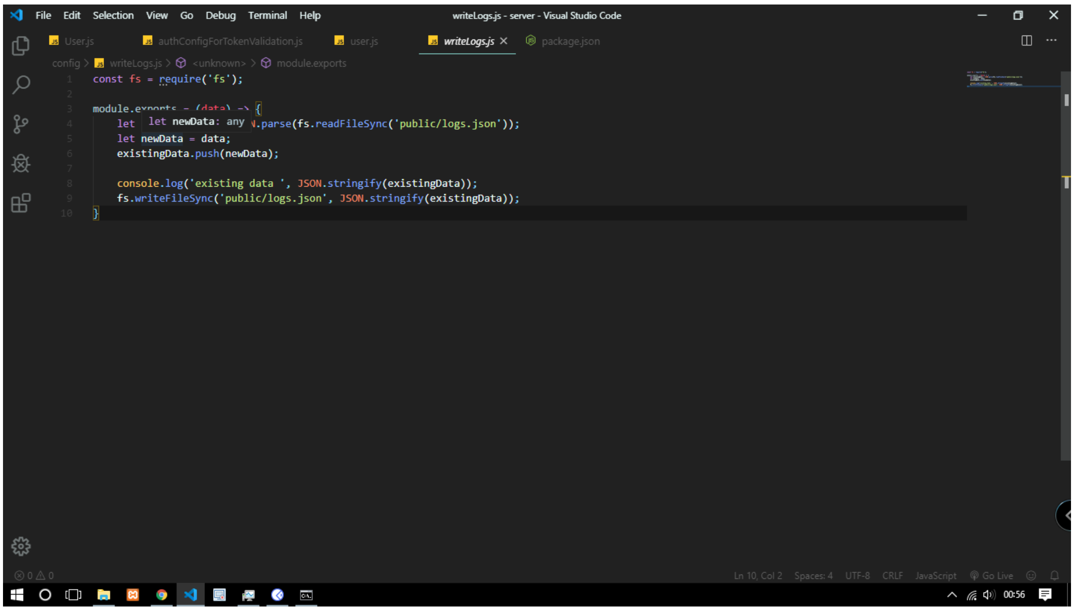
POST localhost:8000/api/v1/users/signin

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL BETA JSON

```json
{
    "email":"im@gmail.com",
    "password":"asdfghjkl"
}
```

Figure – 7 *Sending API Request*

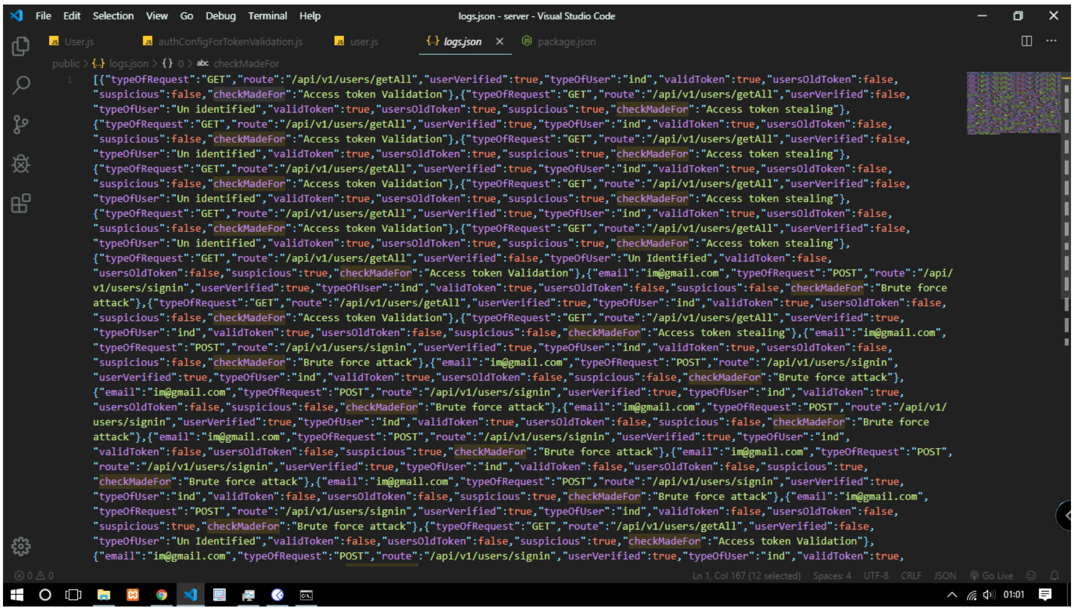
- After clicking the send button, the request is sent, and the response is received


File Edit Selection View Go Debug Terminal Help
writeLogs.js - server - Visual Studio Code
User.js
authConfigForTokenValidation.js
user.js
writeLogs.js
package.json
config > writeLogs.js > <unknown> > module.exports
Ln 10, Col 2
Spaces: 4
UTF-8
CRLF
JavaScript
Go Live

```javascript
const fs = require('fs');

module.exports = (data) => {
    let  let newData: any N.parse(fs.readFileSync('public/logs.json'));
    let newData = data;
    existingData.push(newData);

    console.log('existing data ', JSON.stringify(existingData));
    fs.writeFileSync('public/logs.json', JSON.stringify(existingData));
}
```

Figure – 8 *Writing log as files*

These logs are written as files into a folder named public and the file as logs.json public/logs.json.


File Edit Selection View Go Debug Terminal Help
logs.json - server - Visual Studio Code
User.js
authConfigForTokenValidation.js
user.js
logs.json
package.json
public > logs.json > {} 0 > checkMadeFor
Ln 1, Col 167 (12 selected) Spaces: 4 UTF-8 CRLF JSON Go Live
01:01

```
[{"typeOfRequest":"GET","route":"/api/v1/users/getAll","userVerified":true,"typeOfUser":"ind","validToken":true,"usersOldToken":false,
"suspicious":false,"checkMadeFor":"Access token Validation"},{"typeOfRequest":"GET","route":"/api/v1/users/getAll","userVerified":false,
"typeOfUser":"Un identified","validToken":true,"usersOldToken":true,"suspicious":true,"checkMadeFor":"Access token stealing"},
{"typeOfRequest":"GET","route":"/api/v1/users/getAll","userVerified":true,"typeOfUser":"ind","validToken":true,"usersOldToken":false,
"suspicious":false,"checkMadeFor":"Access token Validation"},{"typeOfRequest":"GET","route":"/api/v1/users/getAll","userVerified":false,
"typeOfUser":"Un identified","validToken":true,"usersOldToken":true,"suspicious":true,"checkMadeFor":"Access token stealing"},
{"typeOfRequest":"GET","route":"/api/v1/users/getAll","userVerified":true,"typeOfUser":"ind","validToken":true,"usersOldToken":false,
"suspicious":false,"checkMadeFor":"Access token Validation"},{"typeOfRequest":"GET","route":"/api/v1/users/getAll","userVerified":false,
"typeOfUser":"Un identified","validToken":true,"usersOldToken":true,"suspicious":true,"checkMadeFor":"Access token stealing"},
{"typeOfRequest":"GET","route":"/api/v1/users/getAll","userVerified":true,"typeOfUser":"ind","validToken":true,"usersOldToken":false,
"suspicious":false,"checkMadeFor":"Access token Validation"},{"typeOfRequest":"GET","route":"/api/v1/users/getAll","userVerified":false,
"typeOfUser":"Un identified","validToken":true,"usersOldToken":true,"suspicious":true,"checkMadeFor":"Access token stealing"},
{"typeOfRequest":"GET","route":"/api/v1/users/getAll","userVerified":false,"typeOfUser":"Un Identified","validToken":false,
"usersOldToken":false,"suspicious":true,"checkMadeFor":"Access token Validation"},{"email":"im@gmail.com","typeOfRequest":"POST","route":"/api/
v1/users/signin","userVerified":true,"typeOfUser":"ind","validToken":true,"usersOldToken":false,"suspicious":false,"checkMadeFor":"Brute force
attack"},{"typeOfRequest":"GET","route":"/api/v1/users/getAll","userVerified":true,"typeOfUser":"ind","validToken":true,"usersOldToken":false,
"suspicious":false,"checkMadeFor":"Access token Validation"},{"typeOfRequest":"GET","route":"/api/v1/users/getAll","userVerified":true,
"typeOfUser":"ind","validToken":true,"usersOldToken":false,"suspicious":false,"checkMadeFor":"Access token stealing"},{"email":"im@gmail.com",
"typeOfRequest":"POST","route":"/api/v1/users/signin","userVerified":true,"typeOfUser":"ind","validToken":true,"usersOldToken":false,
"suspicious":false,"checkMadeFor":"Brute force attack"},{"email":"im@gmail.com","typeOfRequest":"POST","route":"/api/v1/users/signin",
"userVerified":true,"typeOfUser":"ind","validToken":true,"usersOldToken":false,"suspicious":false,"checkMadeFor":"Brute force attack"},
{"email":"im@gmail.com","typeOfRequest":"POST","route":"/api/v1/users/signin","userVerified":true,"typeOfUser":"ind","validToken":true,
"usersOldToken":false,"suspicious":false,"checkMadeFor":"Brute force attack"},{"email":"im@gmail.com","typeOfRequest":"POST","route":"/api/v1/
users/signin","userVerified":true,"typeOfUser":"ind","validToken":true,"usersOldToken":false,"suspicious":false,"checkMadeFor":"Brute force
attack"},{"email":"im@gmail.com","typeOfRequest":"POST","route":"/api/v1/users/signin","userVerified":true,"typeOfUser":"ind",
"validToken":false,"usersOldToken":false,"suspicious":true,"checkMadeFor":"Brute force attack"},{"email":"im@gmail.com","typeOfRequest":"POST",
"route":"/api/v1/users/signin","userVerified":true,"typeOfUser":"ind","validToken":false,"usersOldToken":false,"suspicious":true,
"checkMadeFor":"Brute force attack"},{"email":"im@gmail.com","typeOfRequest":"POST","route":"/api/v1/users/signin","userVerified":true,
"typeOfUser":"ind","validToken":false,"usersOldToken":false,"suspicious":true,"checkMadeFor":"Brute force attack"},{"email":"im@gmail.com",
"typeOfRequest":"POST","route":"/api/v1/users/signin","userVerified":true,"typeOfUser":"ind","validToken":false,"usersOldToken":false,
"suspicious":true,"checkMadeFor":"Brute force attack"},{"typeOfRequest":"GET","route":"/api/v1/users/getAll","userVerified":false,
"typeOfUser":"Un Identified","validToken":false,"usersOldToken":false,"suspicious":true,"checkMadeFor":"Access token Validation"},
{"email":"im@gmail.com","typeOfRequest":"POST","route":"/api/v1/users/signin","userVerified":true,"typeOfUser":"ind","validToken":true,
```

Figure – 9 *Generated Logs*

# 6 IMPLEMENTATION

### Step 1:

The generated requests are sent as an API response for the route localhost:8000/api/v1/logs/get/logs. These routes are called by the client to fetch data from the server. Then they are used to display inside the data table. Each field is looped, and each entry is displayed as rows.

### Step 2:

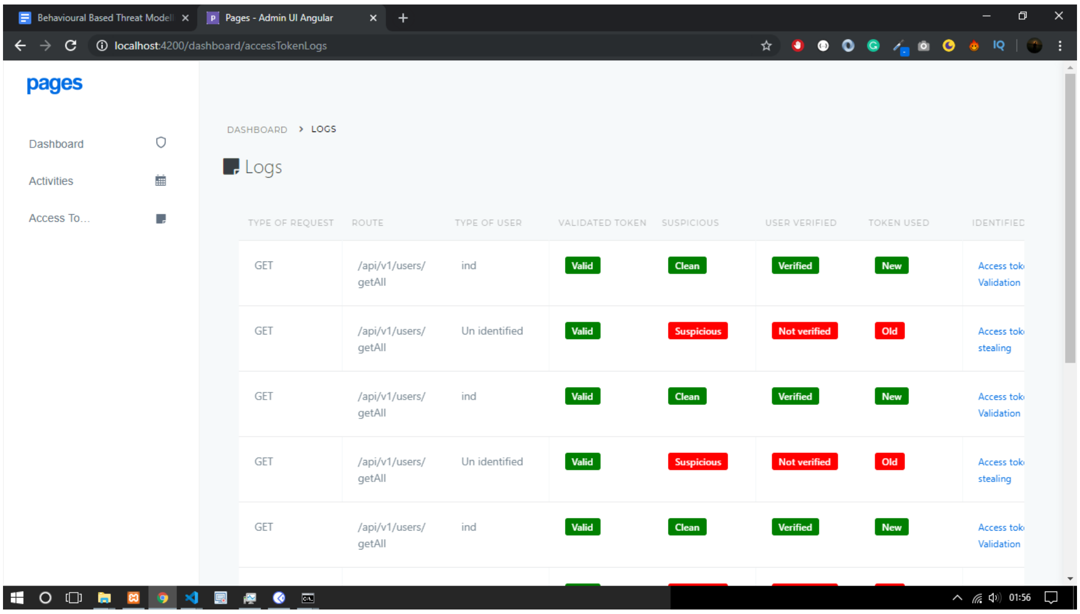
By validating parameters like userVerified, validToken and oldToken the type of activities is identified. For example, if the user uses an old token and send it as a header then it is clearly a suspicious activity and can be identified easily using queries as the new token of the user is sent to the database.

### Step 3:

The alerts are generated on the frontend by using a package called ngx-datatable. The output is displayed as a table format. the alerts are displayed using ng-class and background colours based on the condition of the statements.

{'danger': !row.validToken, 'success': row.validToken}

If it is not a valid token danger gets applied if it's a valid token success will be applied. Pages is an admin dashboard template that allow us to create dashboard UI faster and more efficiently. And for technology stack we are using angular for creating single page applications.



| TYPE OF REQUEST | ROUTE | TYPE OF USER | VALIDATED TOKEN | SUSPICIOUS | USER VERIFIED | TOKEN USED | IDENTIFIED |
|---|---|---|---|---|---|---|---|
| GET | /api/v1/users/ getAll | ind | Valid | Clean | Verified | New | Access tok Validation |
| GET | /api/v1/users/ getAll | Un identified | Valid | Suspicious | Not verified | Old | Access tok stealing |
| GET | /api/v1/users/ getAll | ind | Valid | Clean | Verified | New | Access tok Validation |
| GET | /api/v1/users/ getAll | Un identified | Valid | Suspicious | Not verified | Old | Access tok stealing |
| GET | /api/v1/users/ getAll | ind | Valid | Clean | Verified | New | Access tok Validation |

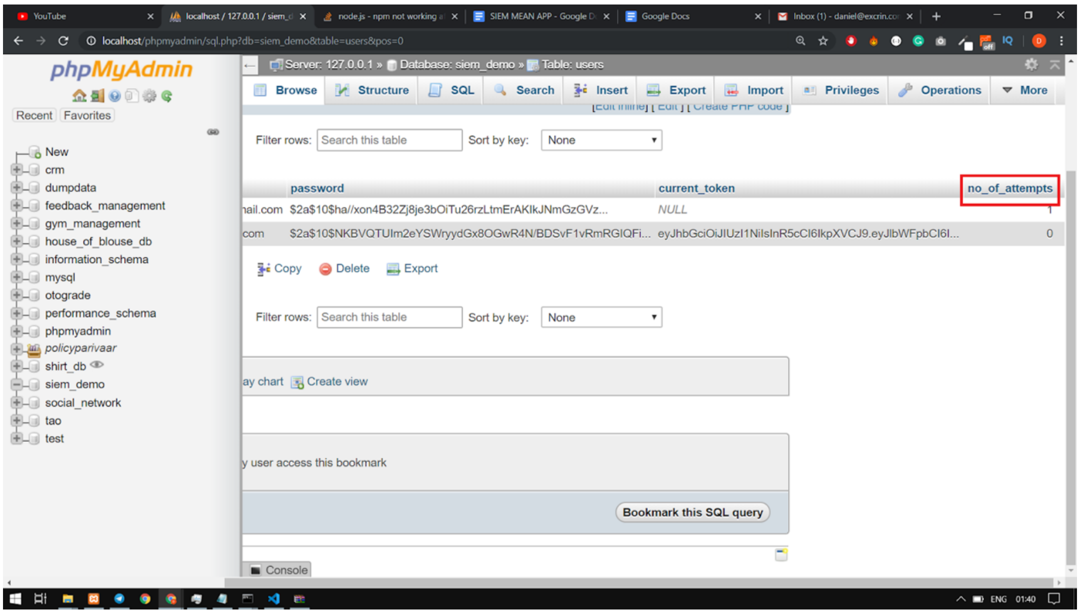
Figure – 10 *Pages Dashboard*

## 6.1 Techniques Implemented

**Brute Force** – ***T1110***:
When passwords are unknown or with the hash values of the passwords, adversaries might use this technique to get access to accounts by attempting to login with number of possible passwords.

Tactic Accomplished: Credential Access
Data Sources: *Authentication logs*.

- Field in DB called "no_of_attempts" represents the no of times the user has tried to log in to the system.
- So in case of each login failure, the "no_of_attempts" field will be incremented by one.
- If the attempts reach a maximum of 3 then it is considered as a possible brute force attack.
- If the user signs into the account within the third time this gets automatically restored to 0.



Server: 127.0.0.1 » Database: siem_demo » Table: users

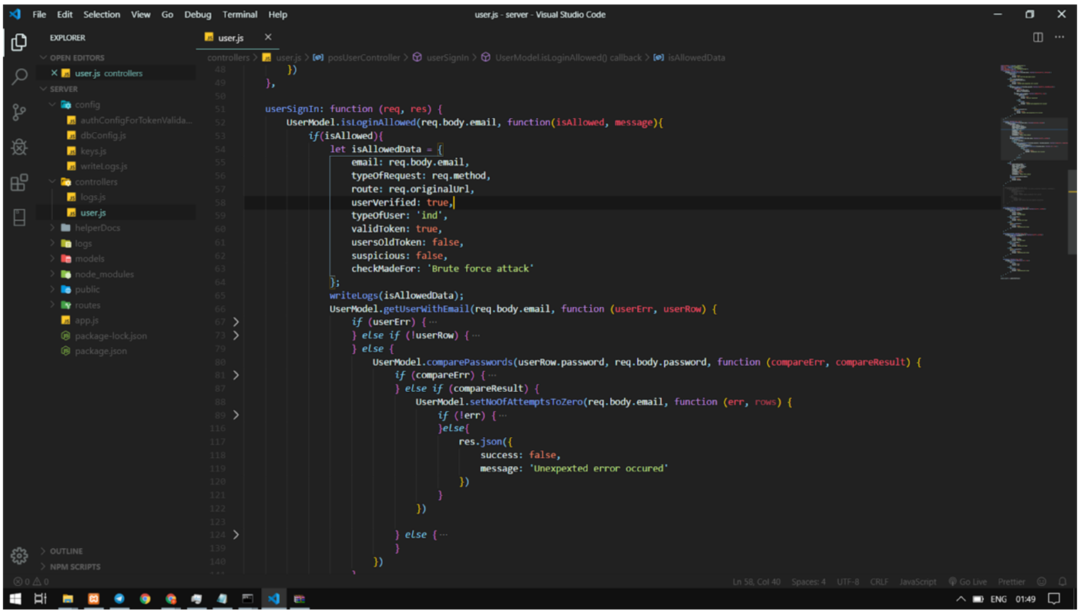
Filter rows: Search this table Sort by key: None

| | password | current_token | no_of_attempts |
|---|---|---|---|
| ail.com | $2a$10$ha//xon4B32Zj8je3bOiTu26rzLtmErAKlkJNmGzGVz... | NULL | 1 |
| com | $2a$10$NKBVQTUIm2eYSWryydGx8OGwR4N/BDSvF1vRmRGIQFi... | eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6I... | 0 |

Figure – 11 *Database with login attempts*

Steps followed for safer login

- Bcryptjs, a hashing library is used that will hash the passwords to any non-readable format there by preventing the users from stealing passwords directly from the DB. Even if they steal this is the modified order of sign-in process to our system.
    - Checking if there are maximum attempts made for that email
    - Taking the row which contains the email.
    - Matching the hashed password with the normal password.
    - If fails it will increment the count by one. And if a maximum count is reached hack login attempt is written to the log.
    - If passed, then it will reset them to zero and send them a token. Later successful attempt is written to the log.
    - This will prevent multiple attempts.



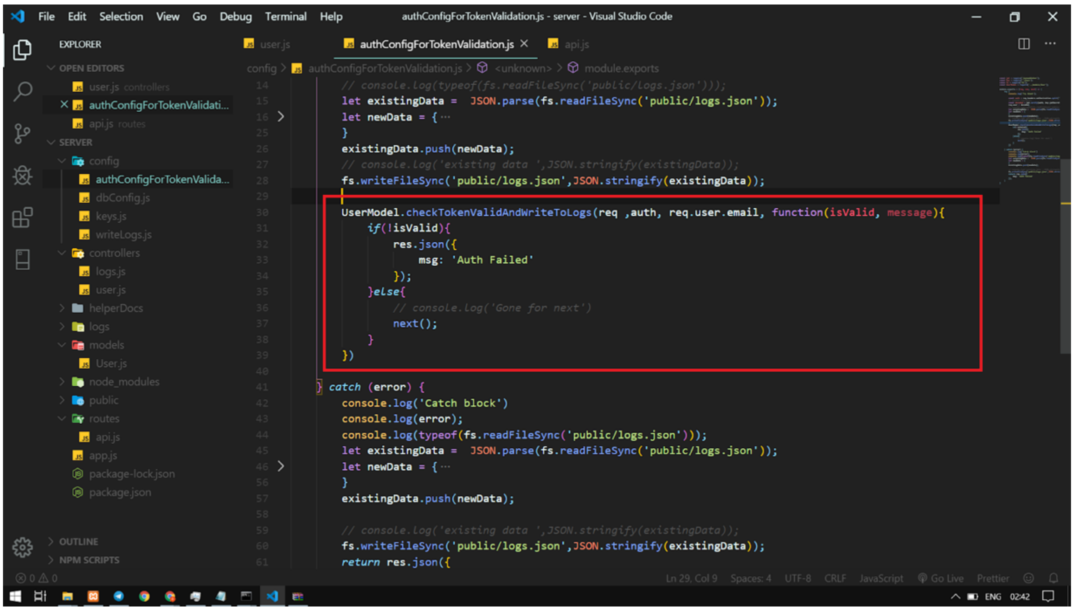
```
        })
    },

    userSignIn: function (req, res) {
        UserModel.isLoginAllowed(req.body.email, function(isAllowed, message){
            if(isAllowed){
                let isAllowedData = {
                    email: req.body.email,
                    typeOfRequest: req.method,
                    route: req.originalUrl,
                    userVerified: true,
                    typeOfUser: 'ind',
                    validToken: true,
                    usersOldToken: false,
                    suspicious: false,
                    checkMadeFor: 'Brute force attack'
                };
                writeLogs(isAllowedData);
                UserModel.getUserWithEmail(req.body.email, function (userErr, userRow) {
                    if (userErr) {···
                    } else if (!userRow) {···
                    } else {
                        UserModel.comparePasswords(userRow.password, req.body.password, function (compareErr, compareResult) {
                            if (compareErr) {···
                            } else if (compareResult) {
                                UserModel.setNoOfAttemptsToZero(req.body.email, function (err, rows) {
                                    if (!err) {···
                                    }else{
                                        res.json({
                                            success: false,
                                            message: 'Unexpexted error occured'
                                        })
                                    }
                                })

                            } else {···
                            }
                        })
```

Figure – 12 *Call Stack Code*

**Steal Application Access Token** – ***T1528***:
In order to authorize API request access tokens are used, adversaries can use these tokens to access the resource by stealing them from the user.

Tactic Accomplished: Credential Access
Data sources: *Activity logs and Audit logs.*

- In order to prevent Access token gets stolen, the current token of the user is stored in the database.
- While authenticating the user is identified by the email and set the current token of the user.
- So that we can have validations on both valid token and older token. If the user authenticates himself using the older token, we can identify him as a hacker. Since he authenticated himself with a newer token later on and the client will only store the newer token whenever it authenticates itself with the server.
- So, we can check whether the sent token is valid or not, if it's not valid then an access token error is created. But if the token is still valid and not present in the DB that is a suspicious event.



```
    // console.log(typeof(fs.readFileSync('public/logs.json')));
    let existingData =  JSON.parse(fs.readFileSync('public/logs.json'));
    let newData = {···
    }
    existingData.push(newData);
    // console.log('existing data ',JSON.stringify(existingData));
    fs.writeFileSync('public/logs.json',JSON.stringify(existingData));

    UserModel.checkTokenValidAndWriteToLogs(req ,auth, req.user.email, function(isValid, message){
        if(!isValid){
            res.json({
                msg: 'Auth Failed'
            });
        }else{
            // console.log('Gone for next')
            next();
        }
    })

} catch (error) {
    console.log('Catch block')
    console.log(error);
    console.log(typeof(fs.readFileSync('public/logs.json')));
    let existingData =  JSON.parse(fs.readFileSync('public/logs.json'));
    let newData = {···
    }
    existingData.push(newData);

    // console.log('existing data ',JSON.stringify(existingData));
    fs.writeFileSync('public/logs.json',JSON.stringify(existingData));
    return res.json({
```

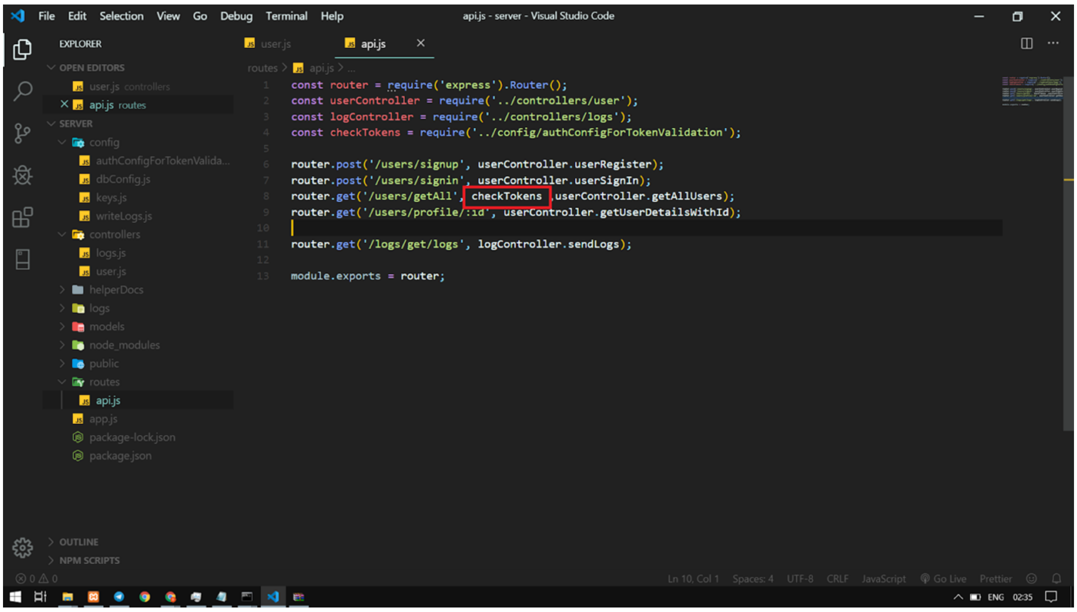
Figure – 13 *Calling Mechanism*

**Access Token Manipulation** – ***T1086***:
Adversaries will use the access token of the user who started the process to evade defensive measures to accomplish the respective tactic.

Tactic Accomplished: Defence Evasion & Privilege Escalation
Data sources: *API Monitoring, Access tokens and Process monitoring*.

- Tokens are basically generated when the users are authenticated correctly with the username and password
- JWT is used for token generation and validation. Tokens are only valid for a limited amount of time.
- The client side will use that token as a header and give the key as Authorization.

- JWT has a built-in method called jwt.decode to validate tokens by decoding it and finding errors. If the present token is not valid then it decoded.
- This module is exported and used as a middleware for all the routes in the API which has been discussed below.



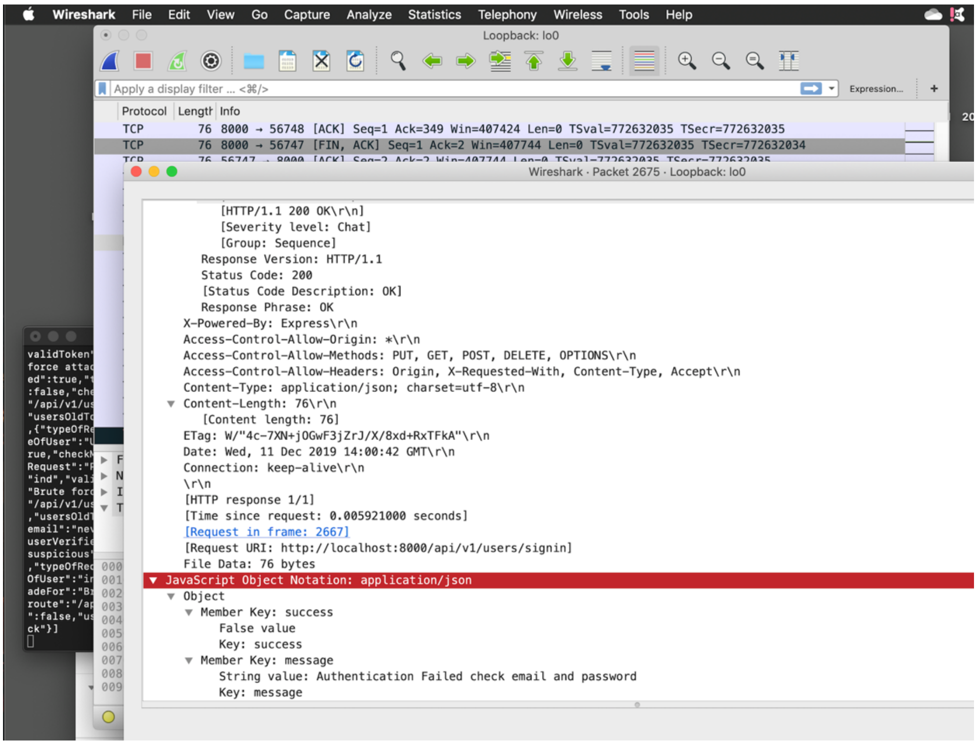
```
const router = require('express').Router();
const userController = require('../controllers/user');
const logController = require('../controllers/logs');
const checkTokens = require('../config/authConfigForTokenValidation');

router.post('/users/signup', userController.userRegister);
router.post('/users/signin', userController.userSignIn);
router.get('/users/getAll', checkTokens ,userController.getAllUsers);
router.get('/users/profile/:id', userController.getUserDetailsWithId);

router.get('/logs/get/logs', logController.sendLogs);

module.exports = router;
```

Figure – 14 *API Routes*

# 7 EVALUATION

The Evaluation of this project has high limitations as this can only be evaluated in a real-time or in a live production environment. Since we used artificial logs as data source the accuracy of the effectiveness in a real-time environment cannot be obtain. As we need real users performing their daily tasks with real background noise to evaluate the effectiveness of this framework. However, the evaluation for this prototype has been done using network sniffing tool called Wireshark. Where the packets are captured and analysed to see if the application produces the right results as it is seen.

The below image is snipped from Wireshark tool where it shows the string value as "Authentication failed check email and password" and the key value as "message". The member key success as "false" value and the key value as "success".



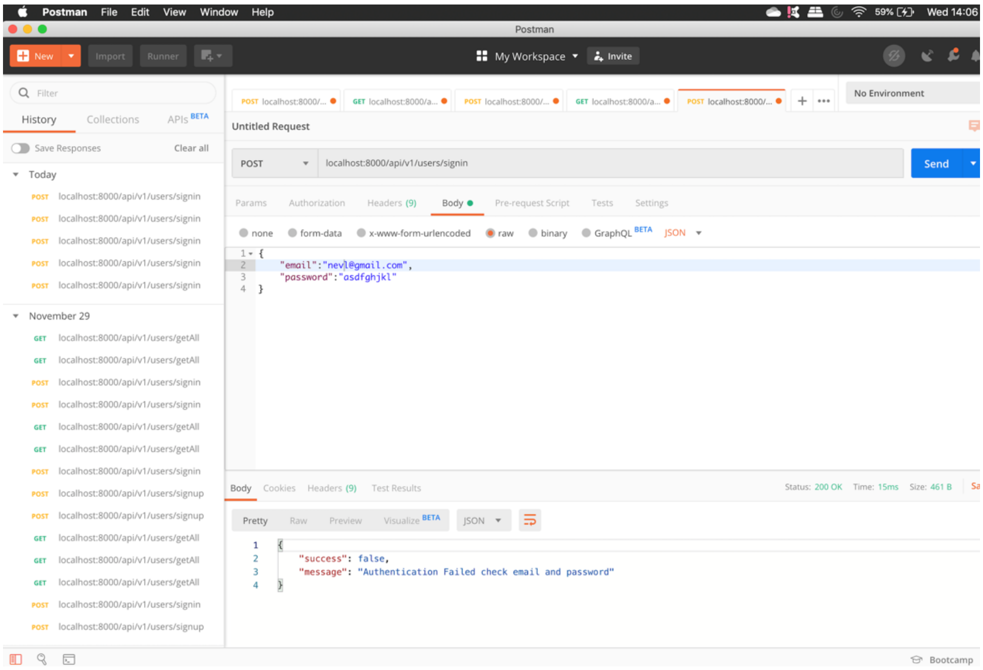
```
Protocol Length Info
TCP      76  8000 → 56748 [ACK] Seq=1 Ack=349 Win=407424 Len=0 TSval=772632035 TSecr=772632035
TCP      76  8000 → 56747 [FIN, ACK] Seq=1 Ack=2 Win=407744 Len=0 TSval=772632035 TSecr=772632034
```

Wireshark · Packet 2675 · Loopback: lo0

```
        [HTTP/1.1 200 OK\r\n]
        [Severity level: Chat]
        [Group: Sequence]
      Response Version: HTTP/1.1
      Status Code: 200
      [Status Code Description: OK]
      Response Phrase: OK
    X-Powered-By: Express\r\n
    Access-Control-Allow-Origin: *\r\n
    Access-Control-Allow-Methods: PUT, GET, POST, DELETE, OPTIONS\r\n
    Access-Control-Allow-Headers: Origin, X-Requested-With, Content-Type, Accept\r\n
    Content-Type: application/json; charset=utf-8\r\n
  ▼ Content-Length: 76\r\n
      [Content length: 76]
    ETag: W/"4c-7XN+jOGwF3jZrJ/X/8xd+RxTFkA"\r\n
    Date: Wed, 11 Dec 2019 14:00:42 GMT\r\n
    Connection: keep-alive\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.005921000 seconds]
    [Request in frame: 2667]
    [Request URI: http://localhost:8000/api/v1/users/signin]
    File Data: 76 bytes
▼ JavaScript Object Notation: application/json
  ▼ Object
    ▼ Member Key: success
        False value
        Key: success
    ▼ Member Key: message
        String value: Authentication Failed check email and password
        Key: message
```

Figure – 15 *Wireshark Packet Analysis*

The below image is snipped from the tool named Postman which is used to send/receive API requests, where it shows the error message as “Authentication failed check email and password”. And the success value is displayed as “false” since the provided credentials didn’t match the token value that was generated.



```
POST  localhost:8000/api/v1/users/signin

Body: raw  JSON
{
    "email":"nevl@gmail.com",
    "password":"asdfghjkl"
}

Status: 200 OK   Time: 15ms   Size: 461 B

{
    "success": false,
    "message": "Authentication Failed check email and password"
}
```

Figure – 16 *Output from Postman*

.

# 8 CONCLUSION AND FUTURE WORK

Identifying the breach and notifying the legal or regulatory agency within the give timeframe has become a great challenge for organizations in the current cyber threat landscape which is changing rapidly. The behavioural based threat model is effective in detecting adversaries with their behaviour when we use the right number of tools and technology. It saves an enormous amount of time for the security analysts to identify the threat and mitigate them at the earlier stage. Although SIEM is one of the best technologies for monitoring and identifying threats in effective way, the future of the incident response will be the new emerging technology called SOAR (Security Orchestration, Automation and Response). The behavioural based threat model can be implemented in an environment where SOAR and SIEM technologies are integrated for rapid and accurate response for constantly evolving cyber-threats.

# REFERENCES

Karyda, Maria and Mitrou, Lilian, "Data Breach Notification: Issues and Challenges for Security Management" (2016). *MCIS 2016 Proceedings*. 60.

X. Lu, J. Han, Q. Ren, H. Dai, J. Li and J. Ou, "Network threat detection based on correlation analysis of multi-platform multi-source alert data", *Multimedia Tools and Applications*, 2018. Available: 10.1007/s11042-018-6689-7 [Accessed 5 August 2019].

D. Fraunholz, D. Krohmer, F. Pohl and H. Schotten, "On the Detection and Handling of Security Incidents and Perimeter Breaches - A Modular and Flexible Honeytoken based Framework", *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 2018.

J. Pavlik, A. Komarek and V. Sobeslav, "Security information and event management in the cloud computing infrastructure", *2014 IEEE 15th International Symposium on Computational Intelligence and Informatics (CINTI)*, 2014.

N. Dwivedi and A. Tripathi, "Event Correlation for Intrusion Detection Systems", *2015 IEEE International Conference on Computational Intelligence & Communication Technology*, 2015. Available: 10.1109/cict.2015.111 [Accessed 5 August 2019].

Killcrece G., Kossakowski K.-P., Ruefle R., Zajicek M. Organizational Models for Computer Security Incident Response Teams. December 2003. URL: http://www.cert.org/archive/pdf/03hb001.pdf (access date 23.03.2019).

I.V. Kotenko, V.V. Vorontsov, A.A. Chechulin, A.V. Ulanov, “Proactive mechanisms of protection against network warms: approach, implementation and experimental results,” Information Technologies, no. 1, 2009. pp.37-42 (in Russian).

I.V. Kotenko, “Intelligent mechanisms for cyber-security management,” // Risc Management and Security. Proceedings of System Analysis Institute Труды Института of RAS. T.41, Moscow, URSS, 2009, pp.74- 103. (in Russian)

K. Mepham, P. Louvieris, G. Ghinea and N. Clewley, "Dynamic cyber-incident response", *2014 6th International Conference On Cyber Conflict (CyCon 2014)*, 2014.

V. Desnitsky and I. Kotenko, "Event analysis for security incident management on a perimeter access control system", *2016 XIX IEEE International Conference on Soft Computing and Measurements (SCM)*, 2016.

R. Mooi and R. Botha, "Prerequisites for building a Computer Security Incident Response capability", *2015 Information Security for South Africa (ISSA)*, 2015.

Strom, B., Battaglia, J.A., Kemmerer, M.S., Miller, D.P., Wampler, C., Whitley, S.M., Wolf, R.D., n.d. Finding Cyber Threats with ATT&CK-Based Analytics 53.