Swinburne University of Technology

School of Science, Computing, and Engineering Technologies

ASSIGNMENT AND PROJECT COVER SHEET

Unit Code: **COS80013** Unit Title: **Internet Security**

Assignment number and title: **Assignment_2_Part_C_Literature Review**

Research Review of Emerging Technologies to defend against Ransomware based Cyber Attacks

Due date: **02/06/2025**

Lab/tutor group: **Friday – 6:30 pm – 8:30 pm** Tutor: **Yasas Akurudda Liyanage Don**

Lecturer: **Rory Coulter , Yasa Akurudda Liyanage Don**

Full name: **Arun Ragavendhar Arunachalam Palaniyappan** Id no: **104837257**

I declare that this assignment is my individual work. I have not worked collaboratively nor have I copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part been written for me by another person.

Signature: Arun Ragavendhar

Marker's comments:

Total Mark:

Contents

Student ID: 104837257

Student Name: Arun Ragavendhar Arunachalam Palaniyappan

Total word Count (excluding references, tables, table of contents): 2166

1. Introduction

The STARFLEET cyber forensic investigation revealed a full-chain ransomware attack carried out by **APT38 (Lazarus Group)**, a North Korean state-backed threat actor known for targeting enterprise systems to cause financial damage. The attacker followed a multi-stage plan, starting with phishing and ending in system-wide file encryption.

This review explores how emerging cybersecurity defence trends can help detect, block, or respond to such attacks. It draws on current research into machine learning (ML), honeypots, multi-factor authentication (MFA), and real-time log monitoring using SIEM (Security Information and Event Management) systems. These defences are analysed based on how they address real-world attacker behaviours, also known as MITRE TTPs (Tactics, Techniques, and Procedures), that were observed in the STARFLEET incident. The main goal is to evaluate whether these tools can actually be useful in detecting and mitigating threats like APT38 and their TTPs at each stage of an attack.

2. Identified TTPs and Current Trends

2.1 Threat actor and TTPs Observed in the STARFLEET Attack

The STARFLEET breach began with a fake job offer sent via email to a staff member. This phishing email successfully tricked the employee into entering his login credentials, giving the attacker their first entry point. From there, the attacker launched a carefully staged intrusion that moved across systems and bypassed detection.

They cracked the Admin password on the Domain Controller using brute-force method, then laterally moved from one machine to another inside the network. They disabled Windows Defender using PowerShell, ran memory-only scripts, and used batch tasks triggered by timeout.exe to keep malware running in the background. Logs were deleted, security tools were turned off, and key internal files like starfleet_secrets.txt were stolen. Finally, ransomware was deployed to lock up user data.

Each step matches known attacker behaviours from the MITRE ATT&CK framework. These TTPs are used by many real-world groups like APT38, and STARFLEET incident is a clear example.

Tactic	Technique ID & Name	Description
Initial Access	T1566 – Phishing	Fake job offer email used to steal Chris Pike's credentials
Initial Access	T1133 – External Remote Services	RDP connection from Tor IP to RM.20
Credential Access	T1110 – Brute Force	Admin account cracked using a weak password (1q2w3e4r5t6y)

Execution	T1059.001 – PowerShell	PowerShell used to disable Defender and run malware scripts
Persistence	T1053.005 – Scheduled Task or Job	Hidden batch file ran repeatedly using timeout.exe
Defense Evasion	T1562.001 – Disable or Modify Tools	Defender's real-time protection turned off to bypass antivirus
Defense Evasion	T1070 – Indicator Removal	Logs deleted and event logging service shut down
Lateral Movement	T1021.001 – Remote Services: RDP	Movement from RM.20 → Domain Controller → Chris → MrSuru
Privilege Escalation	T1078 – Valid Accounts	Stolen Admin and fake "Klingon" account used to gain higher access
Impact	T1486 – Data Encrypted for Impact	agent.exe ransomware encrypted files on Chris's device
Exfiltration	T1041 – Exfiltration Over C2	Sensitive file (starfleet_secrets.txt) downloaded using SMB
Persistence / Evasion	T1012 – Query Registry	PowerShell policy changes found in memory to allow unrestricted script running

Table 1: MITRE ATT&CK Tactics and Techniques identified in the STARFLEET incident

The next section looks at the latest research into emerging cyber defensive trends.

2.2 Detecting Ransomware Using Machine Learning

Recent literature shows ransomware can be detected using file-based or behaviour-based machine learning models. **Arabo et al. (2023)** trained XGBoost and decision tree models using opcode patterns from PE files, achieving 98.3% accuracy on known ransomware. These models detect malware before execution, but only if the full file is scanned, making them ineffective for fileless or in-memory attacks like those in the STARFLEET incident.

Kumar and Singh (2022) moved detection closer to real-time activity. Their models used Windows Event IDs 4688 (process creation) and 4657 (registry edits), which matched what happened when PowerShell was used to disable Defender. They achieved 97% detection using Random Forest and Logistic Regression.

Patel et al. (2023) tested various ML algorithms on dynamic datasets. Random Forest and deep neural networks outperformed traditional models at spotting early-stage infections. But these models struggle if training data is limited or unbalanced, meaning regular tuning and full log coverage are needed to detect stealthy TTPs.

2.3 Honeypots to Trigger Early Warnings

Honeypots are decoy systems or files made to look like real targets. **Zhuravchak et al. (2021)** used symbolic links as fake files. When ransomware accessed them, the system triggered alerts and isolated the host. This stopped malware before real files were hit, blocking impact (T1486) and script execution (T1059).

Voerman et al. (2020) introduced adaptive honeypots that changed bait based on attacker behaviour. These detected file browsing, login attempts, or odd timing patterns. Dionaea, another honeypot, mimicked SMB services, the same used in STARFLEET to steal secrets.txt. These tools help detect lateral movement (T1021) and data theft (T1041) early.

2.4 Multi-Factor Authentication (MFA)

MFA could have blocked the attacker's initial login using stolen credentials (T1078**). AlSaleem and Alshoshan (2021)** introduced a graphical MFA tied to device IDs, which helped stop phishing and keylogger attacks.

Gadducci et al. (2021) reviewed ten MFA types and found hardware tokens and biometrics the safest, while SMS and email-based MFA were easy to bypass. However, MFA only protects the login phase. It cannot prevent actions after access, like PowerShell use (T1059) or privilege escalation (T1078). It must be paired with behaviour monitoring to detect post-login activity.

2.5 SIEM and Real-Time Log Monitoring

SIEM systems help link suspicious activities. **Zhang et al. (2024)** used Windows audit logs to build process graphs and trained a Graph Neural Network that flagged PowerShell abuse, registry edits, and odd process chains, clear signs in STARFLEET.

Singh et al. (2022) combined SIEM with IDS alerts and used Random Forest to detect login failures, off-hours activity, and script use, matching the brute-force attack and evasive steps on MrSuru's system.

Ali et al. (2024) applied SIEM to web server logs, finding URL entropy, time variance, and user-agent anomalies to spot bot traffic and scripted logins. These methods directly address T1110, T1070, and T1133. **Rahim et al. (2024)** showed SIEM works even in niche setups like car networks. With proper logs, SIEM can detect threats in any environment, including STARFLEET.

2.6 Challenges in Applying Emerging Defence Trends

Across all research papers, one challenge is clear: real-time monitoring of system behaviour detects threats much faster than scanning files alone, yet most defences are only effective when combined. Tools that merge honeypots, SIEM, and machine learning give the best results. Zhuravchak's honeypots worked well with SIEM, isolating infected machines the moment a trap triggered. ML models tracking PowerShell use, task scheduling, or registry edits caught activity similar to the STARFLEET attack. Since the attacker used stolen credentials, remote logins, memory-only scripts, and data theft, the response must also be multi-layered. The literature shows only a combined setup using MFA, SIEM, honeypots, and ML can stop all parts of a complex attack.

3. Use of Features in Threat Detection and Response

3.1 What Are Detection Features?

In cybersecurity, features are bits of information that help systems decide if something is normal or dangerous. They can come from a file's content, system logs, unusual network traffic, or odd user behaviour. Good features make it easier for tools, especially machine learning models, to detect attacks early.

But not all features are easy to use. Some are noisy (causing false alarms), some are hard to collect in real time, and others work only in certain setups. This section reviews feature types in recent research and how well they perform against threat actors and TTPs like those in STARFLEET case.

3.2 File-Based Features

File-based features come from the code or structure of malware files. **Arabo et al. (2023)** extracted opcode patterns from executables to detect ransomware using XGBoost and decision trees. These reached 98.3% accuracy but require the full file to be scanned before execution. This fails in cases like STARFLEET, where malware (e.g., RunMe.ps1) ran directly from memory, skipping files. Entropy, a measure of randomness, is another feature. Malware often uses encryption or packing, raising entropy. **Kumar & Singh (2022)** used entropy thresholds for detection. But high entropy also occurs in ZIP files or installers, so it can cause false positives if not handled properly.

3.3 System Behaviour and Log Features

Some of the most powerful features come from what the system does during an attack. Event ID 4688 logs process starts, this was triggered in STARFLEET when PowerShell disabled Defender. Event ID 4657 logs registry edits, such as changes to script execution policies. Event ID 7045 logs new services, which can show malware installing itself for persistence. **Patel et al. (2023)** used these features in ML models and caught ransomware as it unfolded.

Zhang et al. (2024) went further by creating process graphs from logs. These graphs tracked how actions linked together, for example, PowerShell generating a script that alters Defender settings, then launching a hidden task.

Their Graph Neural Network learned these chains and flagged them accurately. This kind of real-time behaviour tracking is crucial for detecting fileless threats, which don't leave disk traces but do appear in logs, just like the AAAAAAAAAAAAAAAA.bat seen on MrSuru's system in the STARFLEET case.

3.4 Network Traffic Features

Network features track how devices communicate and behave over time. Frequent small packets may signal data exfiltration, while repeated login attempts suggest brute-force attacks. Unusual ports or protocols often point to lateral movement, especially when internal services

are accessed in odd patterns. **Sethia & Jeyasekar (2019)** used Dionaea to simulate SMB services and capture logs showing IPs, file names, and hashes, key indicators when malware spreads through shared drives, like how *starfleet_secrets.txt* was stolen via SMB and a C2 server (T1041).

Ali et al. (2024) analysed web logs using an Isolation Forest model with features like URL length, user-agent string, and login timing. These helped catch automated brute-force attacks (T1110), directly linked to how the Admin password was cracked in the domain controller.

3.5 User Behaviour and Login Patterns

When attackers steal credentials, their behaviour often differs from the real user. Logging in during off-hours, using new devices or unfamiliar IPs, or quickly navigating and running commands can all be signs of compromise. These patterns help flag suspicious logins even when correct credentials are used. **AlSaleem & Alshoshan (2021)** developed a graphical MFA system that checked image click order and device fingerprint, flagging attempts that didn't match the user's usual pattern. **Gadducci et al. (2021)** collected data on OTP reuse, backup methods, and device switching to assess MFA security. These types of behavioural features are useful in spotting phishing and stolen-password attacks (T1566, T1078).

3.6 Honeypot Interaction Features

Honeypots generate specific features because attackers interact with them differently from normal users. They may access decoy files, run commands on fake services, or connect to unusual SMB shares, actions that clearly stand out.

Zhuravchak et al. (2021) logged process ID, timestamp, and the exact file accessed when symbolic link honeypots were triggered, offering early ransomware warnings. **Voerman et al. (2020)** added honeypot logs into SIEM, creating new ML training data. The system learned attacker traffic patterns and improved over time. These features offer strong indicators—but only if the honeypot is realistic enough to attract attackers.

3.7 Challenges in Using Features

While powerful, detection features come with limitations. Noisy features like entropy levels or registry edits can trigger false positives, especially in legitimate software updates. Most system logs are benign, so rare attack patterns often go unnoticed unless models are finely tuned. Real-time collection, particularly for memory-resident scripts, needs specialised tools like Volatility or Sysmon that aren't always deployed. Tracking user behaviour also raises ethical, privacy, and compliance concerns. Finally, features are system-specific, a PowerShell log is only useful on Windows, and honeypots only help if attackers interact with them. That's why researchers favour combining diverse sources—logs, traffic, user actions, and files—using SIEM or ML classifiers for better accuracy.

4. Discussion

4.1 How the Reviewed trends Help Detect and Stop STARFLEET-Like Threats and TTPs

This table reflects a detailed synthesis of research findings, mapping each defence technique to STARFLEET's TTPs, highlighting their strengths and weaknesses.

Stage	Tactic & Technique	What the Attacker Did	How It Can Be Detected or Blocked	Research Backing	Strengths	Weaknesses
1. Entry Point	**T1566 – Phishing**	Sent a fake job offer email with a malicious attachment that stole login credentials	Use MFA with behavioural checks (e.g., device, time); flag unusual sender domains and link reputation; monitor click behaviour	**AlSaleem & Alshoshan (2021):** Image-based MFA blocked 95% of phishing attempts **Gadducci et al. (2021):** Hardware tokens resisted spoofing **Ali et al. (2024):** Logged timing/user-agent changes	Works early in the chain; stops unauthorised logins before they happen	Requires user participation; phishing may still succeed if training or alerts are weak
2. Cracking Passwords	**T1110 – Brute Force**	Repeatedly guessed the Admin password ("1q2w3e4r5t6y") until successful login	SIEM alerts on 5+ failed logins in short time; block IP after pattern; flag off-hours activity	**Singh et al. (2022):** Detected login abuse with 94% accuracy **Ali et al. (2024):** Found brute-force via login timing anomalies	Early-stage detection via login patterns; automated rules can stop repeated attempts	Must tune thresholds; low-and-slow brute-force attacks may go undetected
3. Using Stolen Logins	**T1078 – Valid Accounts**	Used Chris Pike's credentials to log into his workstation and move laterally	Detect login from new devices/IPs, odd hours, or rapid command use; alert on fingerprint mismatch	**Gadducci et al. (2021):** Analysed gaps in SMS/email MFA **AlSaleem & Alshoshan (2021):** Flagged login fingerprint mismatches	Strong signals when attackers log in from new environments; stops misuse early	Cannot block access outright—only flags anomalies after login is successful
4. Remote Access	**T1021.001 – RDP**	Used RDP to access internal machines from outside, starting from RM.20	Restrict RDP to known locations/IPs; alert on RDP from Tor nodes; place honeypots to catch unauthorised access	**Sethia & Jeyasekar (2019):** Dionaea honeypots captured SMB/RDP sessions **Singh et al. (2022):** Detected RDP abuse via SIEM chain analysis	Useful for detecting external remote access; works well with network-based logging	False positives possible if RDP is used for legit remote access; honeypots may be bypassed
5. Running Commands	**T1059.001 – PowerShell**	Used PowerShell to disable Defender and execute scripts to prep ransomware drop	Monitor process creation (ID 4688), registry edits (ID 4657); flag script-block policy changes	**Kumar & Singh (2022):** Detected PowerShell misuse with 97% accuracy **Zhang et al. (2024):** GNN flagged Defender-related process chains	Highly effective for tracking in-memory script activity; maps to real system behaviour	Only works if PowerShell logging and event tracking are fully enabled

6. Staying Hidden	**T1053.005 – Scheduled Task**	Ran batch script repeatedly using timeout.exe to keep malware active in memory	Spot abnormal repetition in task creation; use symbolic honeypots to flag hidden tasks early	**Zhuravchak et al. (2021):** Symbolic file honeypots triggered system lockdown **Zhang et al. (2024):** GNN detected unusual task patterns	Early detection using memory-resident triggers; AI-based correlation spots hidden chains	Needs strong baseline of normal task behaviour; honeypots may be ignored if too obvious
7. Disabling Security	**T1562.001 – Modify Tools**	Used PowerShell to disable Defender and shut off Windows logging	Monitor Defender settings and policy changes; track real-time logs from PowerShell execution	**Kumar & Singh (2022):** Logged real-time Defender disabling **Zhang et al. (2024):** Process graphs showed PowerShell tampering	Real-time detection of script-based defence evasion via logs	Attackers may delay or chain commands to avoid immediate detection
8. Hiding Tracks	**T1070 – Indicator Removal**	Deleted event logs and disabled the Windows Event Logging service	Alert on missing or cleared logs using SIEM; detect gaps in event timelines	**Singh et al. (2022):** Detected event log removal and sequence gaps **Ali et al. (2024):** ML detected timeline breaks	Effective against tampering if logs are monitored externally (e.g., central SIEM)	Only works if logs are centralised; can't recover already deleted data
9. Spreading Internally	**T1021.001 – RDP (Lateral)**	Used RDP from DC to move between internal machines like Chris's and MrSuru's device	Monitor credential reuse across systems; place honeypots that simulate critical machines to flag unauthorised movement	**Sethia & Jeyasekar (2019):** Simulated SMB honeypots recorded lateral actions **Voerman et al. (2020):** Adaptive honeypots logged RDP paths	Helps track internal spread of attacker via behavioural cues	Honeypots must be placed on likely paths; attackers may ignore if not convincing
10. Encrypting Files	**T1486 – Data Encrypted for Impact**	Ran agent.exe ransomware to encrypt all user files	Alert when entropy of files spikes; lock down host if honeypot files are touched	**Arabo et al. (2023):** Detected ransomware with 98.3% accuracy **Zhuravchak et al. (2021):** Fake file triggers enabled instant response	Very effective for flagging encryption attempts quickly	Ineffective for fileless encryption; needs quick response to avoid full encryption
11. Stealing Data	**T1041 – Exfiltration Over C2**	Downloaded starfleet_secrets.txt via SMB to an external IP	Monitor shared folders for large or unusual file transfers; log SMB access times	**Sethia & Jeyasekar (2019):** Honeypots logged IPs and file activity **Voerman et al. (2020):** Triggered alerts from fake file access	Identifies attackers during or just before data exfiltration	SMB logs must be complete; stealthy attackers may use encrypted tunnels to hide data flow
12. Stealth / Persistence	**T1012 – Query Registry**	Changed PowerShell settings to allow unrestricted scripts	Monitor registry for repeated edits to script policies; alert if PowerShell logging is reduced	**Zhang et al. (2024):** GNN flagged registry-based persistence **Kumar & Singh (2022):** Logged script policy changes	Powerful for detecting stealth and persistence changes pre-encryption	Windows-only method; registry edits may be missed if logging is not enabled

Table 2: Emerging defence trends that can detect or block the Identified TTPs in the STARFLEET incident

4.2 Actual Usability of the Reviewed Defences in Real Environments

In attacks similar to the STARFLEET incident, the usefulness of these defenses depends on three main factors. First is log availability. Many tools rely on Windows event logs like 4688, 4657, and 7045. While these are standard in most systems, they must be properly enabled and centralised something STARFLEET failed to do. Second is compatibility. Entropy checks work for file-based ransomware but miss fileless scripts like RunMe.ps1. Hybrid tools using logs, file traits, and behaviour (like Zhang et al.'s graph models) are more reliable for stealthy threats. Third is practicality. Tools like MFA and basic honeypots are easy to roll out. But adaptive honeypots and graph-based models need skilled teams and tuning, which smaller networks may lack. A STARFLEET like system, would benefit more from pre-set SIEM rules or managed cloud-based tools.

In All, the reviewed solutions are usable, but only when log collection is active, models stay updated, and tools match the organisation's size and skills. **No single tool can stop every attack step, but a layered mix of logging, detection, and behaviour analysis is highly useful.** For example, Honeypots, ML models, and login behaviour features can work together to flag ransomware execution (T1486), catch PowerShell evasion (T1059.001) and block credential misuse (T1078) respectively.

4.3 Conclusion and Lessons from STARFLEET incident

STARFLEET's breach wasn't unstoppable, it happened because core defences were missing. MFA could have blocked stolen logins. Monitoring could have caught off-hour logins and script misuse. Honeypots could have flagged malware, and SIEM could have connected key warning signs. Even having one of these defences could have raised an alert. A honeypot would have caught the ransomware. Behaviour models could have detected script and registry misuse. SIEM would have showed the link between Defender shutdown and ransomware launch.

The main lesson is clear: attacks like this succeed in steps. Defence must also be layered. Each tool—MFA, honeypots, SIEM, ML—targets a specific part of the attack. But only together can they stop the full chain. STARFLEET's failure shows that logs alone aren't enough, they must be used actively to detect and respond before damage is done.

5. References

AI/ML for Ransomware Detection

1. Arabo, A., Abdulghani, M. I., & Nordin, R. (2023). A digital DNA sequencing engine for ransomware detection using machine learning. *NDSS Symposium*. https://www.ndss-symposium.org/ndss-paper/a-digital-dna-sequencing-engine-for-ransomware-detection-using-machine-learning/
2. Kumar, R., & Singh, R. (2022). AI-based ransomware detection. *Computers & Security*, *115*, 102605. https://doi.org/10.1016/j.cose.2022.102605

3. Patel, P., Shah, K., & Desai, M. (2023). Exploring ransomware detection using ML and AI. *IEEE Transactions on Information Forensics and Security*, *18*(6), 893–906. https://doi.org/10.1109/TIFS.2023.3245641

Honeypots

4. Zhuravchak, K., Kryvinska, N., & Strauss, C. (2021). Ransomware prevention system design based on file symbolic linking honeypots. *Computers & Security*, *104*, 102226. https://doi.org/10.1016/j.cose.2021.102226

5. Voerman, E., Bou-Harb, E., & Hassan, M. (2020). The role of honeypots in modern cybersecurity strategies. *Springer Lecture Notes in Computer Science*, *12123*, 214–229. https://doi.org/10.1007/978-3-030-50729-5_13

6. Sethia, M., & Jeyasekar, R. (2019). Malware capturing and analysis using Dionaea honeypot. *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, 917–921. https://doi.org/10.1109/ICOEI.2019.8862615

Multi-Factor Authentication (MFA)

7. AlSaleem, K., & Alshoshan, A. (2021). Multi-factor authentication to systems login. *International Journal of Computer Applications*, *182*(36), 25–30. https://doi.org/10.5120/ijca2021921234

8. Gadducci, F., Loddo, G., & Marras, M. (2021). An extensive formal analysis of multi-factor authentication protocols. *ACM Transactions on Privacy and Security (TOPS)*, *24*(3), 1–38. https://doi.org/10.1145/3439724

Log Monitoring and SIEM

9. Zhang, J., Wang, H., & Liu, Q. (2024). A novel malware detection method based on audit logs and graph neural network. *Computers & Security*, *125*, 102972. https://doi.org/10.1016/j.cose.2023.102972

10. Singh, A., Malhotra, A., & Bhargava, R. (2022). Integrated security information and event management (SIEM) with intrusion detection system (IDS) for live analysis based on machine learning. *Computers & Security*, *112*, 102525. https://doi.org/10.1016/j.cose.2021.102525

11. Rahim, A., Mazlan, M., & Ariffin, S. (2024). Developing SIEM and log management for automotive network in a simulated environment. *IEEE Transactions on Vehicular Technology*, *73*(1), 812–821. https://doi.org/10.1109/TVT.2024.3123645

12. Ali, A., Ahmed, R., & Shah, M. (2024). Threat classification model for security information event management focusing on model efficiency. *Computers & Security*, *126*, 102998. https://doi.org/10.1016/j.cose.2024.102998