National College of Ireland

Configuration Manual

MSc Research Project
MSc in Cybersecurity

Devyesh Thomas
Student ID: 23285419

School of Computing
National College of Ireland

Supervisor: Michael Pantridge

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Devyesh Thomas
Student ID:	23285419
Programme:	MSc in Cybersecurity **Year:** 2024-2025
Module:	Practicum
Lecturer:	Michael Pantridge
Submission Due Date:	11/08/2025
Project Title:	Automating Adaptive Deception in Endpoint Detection and Response Systems and Optimizing Decoy Placement
Word Count:	1538 **Page Count:** 9

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: Devyesh Thomas

Date: 11/08/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

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

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

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Devyesh Thomas
Student ID: 23285419

1 System Requirements and Dependencies

This section shows the minimum technical setup required to reproduce an adaptive cyber deception framework that's developed in this project.

1.1 Host Machine Requirements

The project can be performed directly on a standard workstation or laptop that runs macOS, Windows or Linux. A python virtual environment (e.g. *venv* or *conda*) to manage dependencies and ensure reproducibility is recommended.

Component	Minimum Specification	Recommended Specification
CPU	Dual-Core 2.5 GHz	Quad-Core 3.0 GHz+
RAM	4 GB	8 GB+
Storage	2 GB free space	5 GB free space
Network	Basic internet connection (only required for optional Canarytoken API calls)	Stable broadband connection

1.2 Development Environment

The project only needs a local development setup with Python installed.
The scripts may be run and modified using any modern code editor or integrated development environment (IDE), e.g. Visual Studio Code, PyCharm, or Sublime Text.
The scripts may also be executed directly via the system terminal.

1.3 Required Software

The software components below must be installed,

Tool / Library	Version	Purpose
Python	3.11.x	Primary scripting language
pandas	Latest	Data processing
numpy	Latest	Numerical operations
scipy	Latest	Statistical analysis (optional)
matplotlib	Latest	Data visualisation
requests	Latest	Canarytoken API calls (optional)
Canarytokens (optional)	Latest	Deploy file/URL decoys for live testing

1.4 Pre-Installation Checklist

Before execution, the following steps have to be completed:

- Install Python 3.11 and required libraries using

pip install pandas numpy scipy matplotlib requests

- (Optional) Configure Canarytokens with an API endpoint and webhook URL if testing live decoy triggers
- Download or clone the project repository containing:
  simulate_logs.py
  risk_score_calc.py
  decoy_mapper.py
- Verify that the working directory has write permissions for CSV output files

2 Configuration Parameters & Mapping Tables

The static configuration files and mapping of parameters are provided in this section, upon which the simulation, risk scoring and decoy assignment process are based. These parameters must be present and properly setup prior to running the scripts in Section 3.

2.1 Organizational Segment Weight Table

Org Segment	Weight
Board / Directors	5
Domain Admin	4.5
Finance Dept	4
IT Admin	3.5
General Employee	3
Intern / Guest	2

2.2 Attack Severity Weight Table

Attack Type Keyword / MITRE Technique	Weight
Credential Access (T1552, etc.)	5
Execution (T1059, T1204, etc.)	3
Discovery (T1083, T1057, etc.)	2
Lateral Movement (T1021)	4.5
Command and Control (T1071, T1105)	4
Persistence (T1543, T1053, etc.)	3
Impact (T1486)	5
Initial Access (T1566)	4.5

2.3 Risk Score Formula

The Risk Score is computed as:

Risk Score = (Org Segment Weight + Attack Severity Weight) × 10

Risk Level thresholds:
Critical: ≥ 91
High: 71–90
Medium: 51–70
Low: 31–50
Informational: ≤ 30

2.4 Risk Level to Decoy Mapping

Each and every decoy type is designed to attract and log attacker interactions relevant to the assessed threat severity.

Risk Level	Triggered Decoy(s)	Use Case / Purpose

Critical	creds.txt, keylist.docx, smbshare.bak	Fake credentials, documents, and shared folders
High	admin_config.xml, fake_powershell.ps1	Fake admin configs and PowerShell bait scripts
Medium	ftp_access.log, user_notes.txt, Fake scripts (startup.sh, update.bat)	Logs and low-impact files (insider bait). Engaging attackers
Low	browser_history.db, session_data.json, Fake URLs or webhooks	Generic files for casual attacker interaction
Informational	(None)	No decoy—log only

2.5 Canarytokens Configuration (optional)

For live deployments, Canarytokens CLI must be configured by:

- API endpoint from Thinkst Canary console.
- Webhook URL for alert callbacks.
- Decoy token type (file, URL, document).

Example CLI generation command:

python canarytokens.py --type doc-msword --memo "All Passwords Decoy" --webhook https://<webhook-url>

3 Installation & Execution

This section describes the setup and execution of the three core automation scripts created as part of this project. Each script ingests the previous output, forming a full complete pipeline from simulated alert generation to scoring the risk and assignment of context-aware decoys.

3.1 Log Generation Script

Generates an artificial dataset of enterprise threat alerts aligned with MITRE ATT&CK techniques, including tactic, technique ID, organisational segment, and asset criticality score.

Execution Steps:

1. Go to repository folder in command line.
2. Run the script:
   python simulate_logs.py
3. The script will output *simulated_threat_logs.csv* in the same directory.

Example Output:
(first 5 rows)

Alert ID	Attack Type	MITRE ID	Tactic	Org Segment	Asset Criticality
2025-Q1-001	Exfiltration Over Alternative Protocol	T1048	Exfiltration	Board	5
2025-Q1-002	OS Credential Dumping: SAM	T1003.002	Credential Access	Finance Dept	4
2025-Q1-003	Abuse Elevation Control Mechanism: Sudo	T1548.003	Privilege Escalation	General Employee	3
2025-Q1-004	System Information Discovery	T1082	Discovery	Intern	2
2025-Q1-005	Exploitation for Client Execution	T1203	Execution	Intern	2

simulated_threat_logs.csv

Alert ID	Attack Type	MITRE ID	Tactic	Org Segment	Asset Criticality
2025-Q1-001	Exfiltration Over Alternative Protocol	T1048	Exfiltration	Board	5
2025-Q1-002	OS Credential Dumping: SAM	T1003.002	Credential Access	Finance Dept	4
2025-Q1-003	Abuse Elevation Control Mechanism: Sudo	T1548.003	Privilege Escalation	General Employee	3
2025-Q1-004	System Information Discovery	T1082	Discovery	Intern	2
2025-Q1-005	Exploitation for Client Execution	T1203	Execution	Intern	2
2025-Q1-006	Credential Access via Brute Force	T1110	Credential Access	Board	5
2025-Q1-007	Unsecured Credentials: Bash History	T1552.003	Credential Access	Board	5
2025-Q1-008	Remote System Discovery	T1018	Discovery	Finance Dept	4
2025-Q1-009	Application Layer Protocol: HTTPS	T1071.001	Command and Control	Finance Dept	4
2025-Q1-010	Valid Accounts: Local Account	T1078.003	Persistence	IT Admin	3.5
2025-Q1-011	Application Layer Protocol: File Transfer Protocols	T1071.002	Command and Control	General Employee	3
2025-Q1-012	Create or Modify System Process: Windows Service	T1543.003	Persistence	Board	5
2025-Q1-013	Create or Modify System Process: Windows Service	T1543.003	Persistence	Finance Dept	4
2025-Q1-014	Obfuscated Files or Information	T1027	Defense Evasion	General Employee	3
2025-Q1-015	System Information Discovery	T1082	Discovery	General Employee	3
2025-Q1-016	Valid Accounts: Domain Accounts	T1078.002	Defense Evasion	Finance Dept	4
2025-Q1-017	Data Encrypted for Impact	T1486	Impact	Intern	2
2025-Q1-018	Security Software Discovery	T1518.001	Discovery	IT Admin	3.5
2025-Q1-019	Exfiltration Over C2 Channel	T1041	Exfiltration	Intern	2
2025-Q1-020	Valid Accounts: Local Account	T1078.003	Persistence	IT Admin	3.5

3.2 Risk Scoring Script

Processes the simulated_threat_logs.csv to score a numerical risk score for each alert based on:
- Asset criticality (organisational segment weight)
- Attack severity weight (MITRE tactic/technique)

Execution Steps:

4. Go to repository folder in command line.
5. Run the script:
   python risk_score_calc.py
6. The script will generate *risk_scores_summary.csv*.

Example Output:
(first 5 rows)

Alert ID	Attack Type: MITRE ID	Risk Score	Risk Level
2025-Q1-001	Exfiltration Over Alternative Protocol: T1048	60	Medium
2025-Q1-002	OS Credential Dumping: SAM: T1003.002	90	High
2025-Q1-003	Abuse Elevation Control Mechanism: Sudo: T1548.003	75	High
2025-Q1-004	System Information Discovery: T1082	40	Low
2025-Q1-005	Exploitation for Client Execution: T1203	50	Low

risk_scores_summary.csv

Alert ID	Attack Type: MITRE ID	Risk Score	Risk Level
2025-Q1-001	Exfiltration Over Alternative Protocol: T1048	60	Medium
2025-Q1-002	OS Credential Dumping: SAM: T1003.002	90	High
2025-Q1-003	Abuse Elevation Control Mechanism: Sudo: T1548.003	75	High
2025-Q1-004	System Information Discovery: T1082	40	Low
2025-Q1-005	Exploitation for Client Execution: T1203	50	Low
2025-Q1-006	Credential Access via Brute Force: T1110	100	Critical
2025-Q1-007	Unsecured Credentials: Bash History: T1552.003	100	Critical
2025-Q1-008	Remote System Discovery: T1018	60	Medium
2025-Q1-009	Application Layer Protocol: HTTPS: T1071.001	80	High
2025-Q1-010	Valid Accounts: Local Account: T1078.003	65	Medium
2025-Q1-011	Application Layer Protocol: File Transfer Protocols: T1071.002	70	Medium
2025-Q1-012	Create or Modify System Process: Windows Service: T1543.003	80	High
2025-Q1-013	Create or Modify System Process: Windows Service: T1543.003	70	Medium
2025-Q1-014	Obfuscated Files or Information: T1027	60	Medium
2025-Q1-015	System Information Discovery: T1082	50	Low
2025-Q1-016	Valid Accounts: Domain Accounts: T1078.002	70	Medium
2025-Q1-017	Data Encrypted for Impact: T1486	70	Medium
2025-Q1-018	Security Software Discovery: T1518.001	55	Medium
2025-Q1-019	Exfiltration Over C2 Channel: T1041	30	Informational
2025-Q1-020	Valid Accounts: Local Account: T1078.003	65	Medium

3.3 Decoy Deployment

Uses risk levels from risk_scores_summary.csv to assign suitable decoys. Applying predefined mapping logic (e.g., High = admin_config.xml, Medium = startup.sh, Low = generic URLs).

Execution Steps:

7. Go to repository folder in command line.
8. Run the script:
   python decoy_mapper.py
9. The script will generate *decoy_assignment_table.csv*.

Example Output:

(first 5 rows)

Alert ID	Risk Score	Risk Level	Triggered Decoy(s)	Decoy Use Case
2025-Q1-001	60	Medium	startup.sh, update.bat, ftp_access.log, user_notes.txt	Logs and low-impact files like fake startup or updater scripts
2025-Q1-002	90	High	admin_config.xml, fake_powershell.ps1	Fake system admin configs and PowerShell bait scripts
2025-Q1-003	75	High	admin_config.xml, fake_powershell.ps1	Fake system admin configs and PowerShell bait scripts
2025-Q1-004	40	Low	https://internal-monitoring.fake/login, browser_history.db, session_data.json	Generic files and fake URLs for casual attacker interaction
2025-Q1-005	50	Low	https://internal-monitoring.fake/login, browser_history.db, session_data.json	Generic files and fake URLs for casual attacker interaction

decoy_assignment_table.csv

Alert ID	Risk Score	Risk Level	Triggered Decoy(s)	Decoy Use Case	Tool
2025-Q1-001	60	Medium	startup.sh, update.bat, ftp_access.log, user_notes.txt	Logs and low-impact files like fake startup or updater scripts	Canarytokens
2025-Q1-002	90	High	admin_config.xml, fake_powershell.ps1	Fake system admin configs and PowerShell bait scripts	Canarytokens
2025-Q1-003	75	High	admin_config.xml, fake_powershell.ps1	Fake system admin configs and PowerShell bait scripts	Canarytokens
2025-Q1-004	40	Low	https://internal-monitoring.fake/login, browser_history.db, session_data.json	Generic files and fake URLs for casual attacker interaction	Canarytokens
2025-Q1-005	50	Low	https://internal-monitoring.fake/login, browser_history.db, session_data.json	Generic files and fake URLs for casual attacker interaction	Canarytokens
2025-Q1-006	100	Critical	creds.txt, keylist.docx, smbshare.bak	Fake credentials, documents, and shared folders	Canarytokens
2025-Q1-007	100	Critical	creds.txt, keylist.docx, smbshare.bak	Fake credentials, documents, and shared folders	Canarytokens
2025-Q1-008	60	Medium	startup.sh, update.bat, ftp_access.log, user_notes.txt	Logs and low-impact files like fake startup or updater scripts	Canarytokens
2025-Q1-009	80	High	admin_config.xml, fake_powershell.ps1	Fake system admin configs and PowerShell bait scripts	Canarytokens
2025-Q1-010	65	Medium	startup.sh, update.bat, ftp_access.log, user_notes.txt	Logs and low-impact files like fake startup or updater scripts	Canarytokens
2025-Q1-011	70	Medium	startup.sh, update.bat, ftp_access.log, user_notes.txt	Logs and low-impact files like fake startup or updater scripts	Canarytokens
2025-Q1-012	80	High	admin_config.xml, fake_powershell.ps1	Fake system admin configs and PowerShell bait scripts	Canarytokens
2025-Q1-013	70	Medium	startup.sh, update.bat, ftp_access.log, user_notes.txt	Logs and low-impact files like fake startup or updater scripts	Canarytokens
2025-Q1-014	60	Medium	startup.sh, update.bat, ftp_access.log, user_notes.txt	Logs and low-impact files like fake startup or updater scripts	Canarytokens
2025-Q1-015	50	Low	https://internal-monitoring.fake/login, browser_history.db, session_data.json	Generic files and fake URLs for casual attacker interaction	Canarytokens
2025-Q1-016	70	Medium	startup.sh, update.bat, ftp_access.log, user_notes.txt	Logs and low-impact files like fake startup or updater scripts	Canarytokens
2025-Q1-017	70	Medium	startup.sh, update.bat, ftp_access.log, user_notes.txt	Logs and low-impact files like fake startup or updater scripts	Canarytokens
2025-Q1-018	55	Medium	startup.sh, update.bat, ftp_access.log, user_notes.txt	Logs and low-impact files like fake startup or updater scripts	Canarytokens
2025-Q1-019	30	Informational	None	No decoy triggered - Log only	-
2025-Q1-020	65	Medium	startup.sh, update.bat, ftp_access.log, user_notes.txt	Logs and low-impact files like fake startup or updater scripts	Canarytokens

3.4 Simulated Decoy Interaction Results

After decoy assignment, interactions were simulated and mapped with visibility and detection scoring parameter from the DeTTECT framework.

Example Interaction Table: (first 5 rows)

Interaction ID	Source Alert ID	Triggered Decoy(s)	Interaction Type	Time to Detection	Detected By	Notes
INT-2025Q1-001	2025-Q1-006	creds.txt	File Opened	8 sec	Sysmon + ELK	Accessed via PowerShell (T1059)
INT-2025Q1-002	2025-Q1-003	fake_powershell.ps1	Script Executed	14 sec	Sysmon + ELK	Executed from %TEMP% with encoded base64 payload
INT-2025Q1-003	2025-Q1-004	browser_history.db	File Read	26 sec	Zeek	Accessed over SMB from remote IP
INT-2025Q1-004	2025-Q1-007	keylist.docx	File Opened	11 sec	Sysmon	Opened by suspicious Excel instance
INT-2025Q1-005	2025-Q1-009	admin_config.xml	File Access Attempt	7 sec	Canarytoken	Download triggered webhook from attacker IP

Visibility/Detection Score Mapping from Interaction Logs:

Interaction ID	Source Alert ID	Triggered Decoy(s)	Visibility	Detection
INT-2025Q1-001	2025-Q1-006	creds.txt	4	5
INT-2025Q1-002	2025-Q1-003	fake_powershell.ps1	4	4
INT-2025Q1-003	2025-Q1-004	browser_history.db	2	1
INT-2025Q1-004	2025-Q1-007	keylist.docx	4	4
INT-2025Q1-005	2025-Q1-009	admin_config.xml	3	3

References

Git Documentation (2025) *Git reference manual*. Available at: https://git-scm.com/doc (Accessed: 3 August 2025).

Matplotlib Developers (2025) *Matplotlib documentation*. Available at: https://matplotlib.org/stable/contents.html (Accessed: 5 August 2025).

MITRE ATT&CK® (2025) *ATT&CK framework*. MITRE Corporation. Available at: https://attack.mitre.org/ (Accessed: 1 August 2025).

NumPy Developers (2025) *NumPy documentation*. Available at: https://numpy.org/doc/ (Accessed: 2 August 2025).

pandas Development Team (2025) *pandas documentation*. Available at: https://pandas.pydata.org/docs/ (Accessed: 9 August 2025).

Python Requests Library (2025) *Requests documentation*. Available at: https://requests.readthedocs.io/en/latest/ (Accessed: 8 August 2025).

Python Software Foundation (2025) *Python 3.11 documentation*. Available at: https://docs.python.org/3.11/ (Accessed: 4 August 2025).

SciPy Developers (2025) *SciPy documentation*. Available at: https://docs.scipy.org/doc/scipy/ (Accessed: 7 August 2025).

Thinkst Applied Research (2025) *Canarytokens documentation*. Available at: https://docs.canarytokens.org/ (Accessed: 6 August 2025).