

CYSE 650 – Project GIGL

CYSE 650-D01

07/18/2024

Pratam (G)avaravarapu · Rayan (I)ssa Harshya(G)avaravarapu · Chris (L)imson



Project Installation Tutorial

Downloading the repository

Ran the command: **git clone** <https://github.com/chrimsn/GIGL.git>.

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

```
PS C:\Users\96653\Onedrive\Desktop> git clone https://github.com/chrimsn/GIGL.git
Cloning into 'GIGL'...
remote: Enumerating objects: 137, done.
remote: Counting objects: 100% (137/137), done.
remote: Compressing objects: 100% (110/110), done.
remote: Total 137 (delta 77), reused 63 (delta 26), pack-reused 0
Receiving objects: 100% (137/137), 8.57 MiB | 17.95 MiB/s, done.
Resolving deltas: 100% (77/77), done.
PS C:\Users\96653\Onedrive\Desktop>
```

Enabled required decency

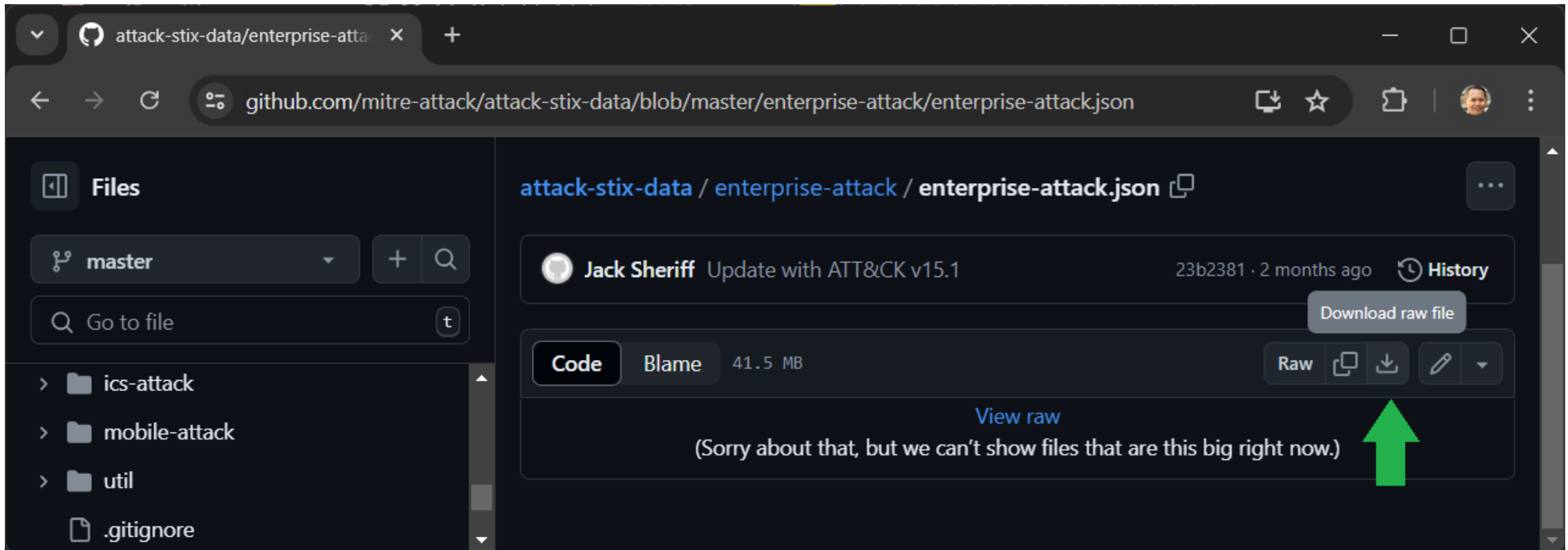
Installed the *pgmpy* library using the command '*pip install pgmpy*'.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\96653\Onedrive\Desktop> cd GIGL
PS C:\Users\96653\Onedrive\Desktop\GIGL> pip install pgmpy
WARNING: Ignoring invalid distribution -andas (c:\users\96653\appdata\local\programs\python\python310\lib\site-packages)
Collecting pgmpy
  Using cached pgmpy-0.1.25-py3-none-any.whl.metadata (6.4 kB)
Requirement already satisfied: networkx in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from pgmpy) (3.3)
Requirement already satisfied: numpy in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from pgmpy) (2.0.0)
Requirement already satisfied: scipy in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from pgmpy) (1.14.0)
Requirement already satisfied: scikit-learn in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from pgmpy) (1.5.0)
Requirement already satisfied: pandas in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from pgmpy) (2.2.2)
Requirement already satisfied: pyparsing in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from pgmpy) (3.1.2)
Requirement already satisfied: torch in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from pgmpy) (2.3.1)
Requirement already satisfied: statsmodels in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from pgmpy) (0.14.2)
Requirement already satisfied: tqdm in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from pgmpy) (4.66.4)
Requirement already satisfied: joblib in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from pgmpy) (1.4.2)
Requirement already satisfied: opt-einsum in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from pgmpy) (3.3.0)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from pandas->pgmpy) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from pandas->pgmpy) (2022.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from pandas->pgmpy) (2024.1)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from scikit-learn->pgmpy) (3.5.0)
Requirement already satisfied: patsy>=0.5.6 in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from statsmodels->pgmpy) (0.5.6)
Requirement already satisfied: packaging>=21.3 in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from statsmodels->pgmpy) (24.1)
Requirement already satisfied: filelock in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from torch->pgmpy) (3.15.4)
Requirement already satisfied: typing-extensions>=4.8.0 in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from torch->pgmpy) (4.12.2)
Requirement already satisfied: sympy in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from torch->pgmpy) (1.12.1)
Requirement already satisfied: Jinja2 in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from torch->pgmpy) (3.1.4)
Requirement already satisfied: fsspec in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from torch->pgmpy) (2024.6.1)
Requirement already satisfied: mkl<=2021.4.0,>=2021.1.1 in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from torch->pgmpy) (2021.4.0)
Requirement already satisfied: colorama in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from tqdm->pgmpy) (0.4.6)
Requirement already satisfied: intel-openmp==2021.* in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from mkl<=2021.4.0,>=2021.1.1->torch->pgmpy) (2021.4.0)
Requirement already satisfied: tbb==2021.* in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from mkl<=2021.4.0,>=2021.1.1->torch->pgmpy) (2021.13.0)
Requirement already satisfied: six in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from patsy>=0.5.6->statsmodels->pgmpy) (1.16.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from Jinja2->torch->pgmpy) (2.1.5)
Requirement already satisfied: mpmath<1.4.0,>=1.1.0 in c:\users\96653\appdata\local\programs\python\python310\lib\site-packages (from sympy->torch->pgmpy) (1.3.0)
Using cached pgmpy-0.1.25-py3-none-any.whl (2.0 MB)
WARNING: Ignoring invalid distribution -andas (c:\users\96653\appdata\local\programs\python\python310\lib\site-packages)
Installing collected packages: pgmpy
Successfully installed pgmpy-0.1.25
PS C:\Users\96653\Onedrive\Desktop\GIGL>
```

Configure (optional)

Downloaded the latest STIX data (versions 15.1) that describes MITRE ATT&CK Techniques



Execution

To run the code and create the Bayesian network, we executed the command
'python gigl.py mitre.json bayes.net'

```
PS C:\Users\96653\Onedrive\Desktop\GIGL> python gigl.py apt__ipt+_fdb.json bayes.net
Mapping techniques from STIX
Bayesian Network model is valid
Bayesian Network exported
PS C:\Users\96653\Onedrive\Desktop\GIGL>
```



Software Architecture

Import necessary libraries and define global variable

```
import json
import numpy as np
import sys
from pgmpy.models import BayesianNetwork
from pgmpy.factors.discrete import TabularCPD
from pgmpy.global_vars import logger

risk = "Risk"
```


Define a function to assign probabilities based on technique scores

```
def assign_probability(score):  
    if score == 1:  
        return 0.5 # 50% chance – used in one of two attacks  
    elif score == 2:  
        return 0.5 # 50% chance – used in one of two attacks  
    elif score == 3:  
        return 1.0 # 100% chance – used in both attacks  
    else:  
        return 0.0 # 0% chance – not used in either attack
```

Create the BN model with nodes

```
def create_bayesian_network(techniques):
    and_tactics = ['Credential_Access', 'Persistence', 'Lateral_Movement']
    model = BayesianNetwork()

    # Group techniques by tactic
    tactics = {}
    for technique in techniques:
        tech_id = technique['name']
        model.add_node(tech_id)
        tactic = technique['tactic'].replace('-', '_').replace('/', '_').title()
        if tactic not in tactics:
            tactics[tactic] = []
        tactics[tactic].append(tech_id)

    # Add techniques to each tactic
    for tactic in tactics:
        model.add_node(tactic)
        for technique in tactics[tactic]:
            model.add_edge(technique, tactic)

    # Add final result node
    model.add_node(risk)
    for tactic in tactics:
        model.add_edge(tactic, risk)
```

Create the BN model with CPDs

```
# Add CPDs for techniques
for technique in techniques:
    prob = assign_probability(technique['score'])
    cpd = TabularCPD(technique['name'], 2, [[1-prob], [prob]])
    model.add_cpds(cpd)

# Add CPDs for tactics with AND/OR logic
for tactic in tactics:
    parents = model.get_parents(tactic)
    num_parents = len(parents)
    cpd_table = np.zeros((2, 2**num_parents))

    if tactic in and_tactics:
        # AND logic for specific tactics
        for i in range(2**num_parents):
            bin_rep = format(i, '0' + str(num_parents) + 'b')
            if all(int(bit) for bit in bin_rep):
                cpd_table[1][i] = 1 # True only when all parents are True
                cpd_table[0][i] = 0
            else:
                cpd_table[1][i] = 0
                cpd_table[0][i] = 1 # False in all other cases
    else:
        # OR logic for other tactics
        for i in range(2**num_parents):
            bin_rep = format(i, '0' + str(num_parents) + 'b')
            if any(int(bit) for bit in bin_rep):
                cpd_table[1][i] = 1 # True if any parent is True
                cpd_table[0][i] = 0
            else:
                cpd_table[1][i] = 0
                cpd_table[0][i] = 1 # False if all parents are False

    cpd = TabularCPD(tactic, 2, cpd_table, evidence=parents, evidence_card=[2]*len(parents))
    model.add_cpds(cpd)
```

```
# Add OR logic CPD for the risk node
parents = model.get_parents(risk)
num_parents = len(parents)
cpd_table = np.zeros((2, 2**num_parents))

for i in range(2**num_parents):
    bin_rep = format(i, '0' + str(num_parents) + 'b')
    if any(int(bit) for bit in bin_rep):
        cpd_table[1][i] = 1.0
    else:
        cpd_table[1][i] = 0.0

cpd_table[0] = 1 - cpd_table[1]
cpd = TabularCPD(risk, 2, cpd_table, evidence=parents, evidence_card=[2]*len(parents))
model.add_cpds(cpd)

return model
```

Export the BN model to .net

```
def export_to_net(model, filename):
    with open(filename, 'w') as f:
        # Write header
        f.write("net\n{\n}\n")
        x = 0
        yb = 50
        offset = False
        has_parents = False
        new_level = False

        # Determine node placement
        for node in model.nodes():
            if not has_parents:
                parents = model.get_parents(node)
                if parents:
                    has_parents = True
                    new_level = True
            if x > 1250 or new_level:
                x = np.random.randint(0, 200)
                yb += 150
                new_level = False
            x += 200
            if offset:
                y = yb + 50
                offset = False
            else:
                y = yb
                offset = True
            if node == risk:
                x = 800
                y = yb + 250
```

```
        # Write node definitions
        f.write(f"node {node}\n")
        f.write("{\n")
        f.write("    states = (\"False\" \"True\");\n")
        f.write(f"    label = \"{node}\";\n")
        f.write(f"    position = ({x} {y});\n")
        f.write("}\n")

        # Write probability definitions
        for cpd in model.get_cpds():
            node = cpd.variable
            parents = model.get_parents(node)
            f.write(f"potential ({node}")
            if parents:
                f.write(f" | {' '.join(parents)}")
            f.write(")\n{\n")
            f.write("    data = ")

            # Flattening the CPT values for .net format
            values = cpd.values
            if values.ndim == 1: # No parents
                probs = values.flatten()
            else:
                probs = np.transpose(values, tuple(range(values.ndim - 1, -1, -1))).flatten()

            f.write("(" + " ".join(f"{p:.6f}" for p in probs) + ")")
            f.write(";\n}\n")
```

Main Execution

```
def main():
    logger.disabled = True
    json_file = sys.argv[1]
    with open(json_file, 'r') as f:
        data = json.load(f)

    # Make lookup table for technique names of IDs
    print("Mapping techniques from STIX")
    tech_map = {}
    stix_file = 'enterprise-attack.json'
    with open(stix_file, 'r') as f:
        stix = json.load(f)
    for object in stix['objects']:
        if object['type'] == 'attack-pattern':
            for xref in object['external_references']:
                if xref['source_name'] == 'mitre-attack':
                    tech_map[xref['external_id']] = object['name'].replace(' ', '_').replace('-', '_').replace('/', '_')

    techniques = data['techniques']
    for technique in techniques:
        technique['name'] = tech_map[technique['techniqueID']]
    model = create_bayesian_network(techniques)

    # Check if the model is valid
    if model.check_model():
        print("Bayesian Network model is valid")
    else:
        print("Bayesian Network model is not valid")

    # Export to .net file
    export_to_net(model, sys.argv[2])
    print("Bayesian Network exported")

main()
```



Scenarios 1

Ransomware Attack

Overview

- **Ransomware attack** targets financial services company
- Spear phishing email containing malicious excel file that executes a powershell script when opened
- Powershell script downloads additional malware
 - Scheduled task
- Zero-day vulnerability exploited - gain higher privileges
- Stole credentials and remote services used
 - Lateral Movement and Exfiltration of customer data
- After two weeks, the ransomware encrypted critical files and left a ransom that demanded a cryptocurrency payment



Tactics & Techniques

- Initial Access: Spear Phishing Attachment (T1566.001)
- Execution: PowerShell (T1059.001)
- Persistence: Scheduled Task/Job (T1053.005)
- Privilege Escalation: Exploitation for Privilege Escalation (T1068)
- Defense Evasion: Obfuscated Files or Information (T1027)
- Credential Access: Credential Dumping (T1003)
- Discovery: System Network Configuration Discovery (T1016)
- Lateral Movement: Remote Services (T1021)
- Collection: Data Staged (T1074)
- Exfiltration: Exfiltration Over Command and Control Channel (T1041)
- Impact: Data Encrypted for Impact (T1486)



Reconnaissance 10 techniques	Resource Development 8 techniques	Initial Access 10 techniques	Execution 14 techniques	Persistence 20 techniques	Privilege Escalation 14 techniques	Defense Evasion 43 techniques	Credential Access 17 techniques	Discovery 32 techniques	Lateral Movement 9 techniques	Collection 17 techniques	Command and Control 18 techniques	Exfiltration 9 techniques	Impact 14 techniques
Active Scanning (0/3)	Acquire Access	Content Injection		Account Manipulation (0/6)	Abuse Elevation Control Mechanism (0/6)	Abuse Elevation Control Mechanism (0/6)	Adversary-in-the-Middle (0/3)	Account Discovery (0/4)	Exploitation of Remote Services	Adversary-in-the-Middle (0/3)	Application Layer Protocol (0/4)	Automated Exfiltration (0/1)	Account Access Removal
Gather Victim Host Information (0/4)		Botnet		BITS Jobs	Access Token Manipulation (0/5)	Access Token Manipulation (0/5)	Brute Force (0/4)	Application Window Discovery	Internal Spearphishing	Archive Collected Data (0/3)	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction
Gather Victim Identity Information (0/3)		DNS Server	AppleScript	Boot or Logon Autostart Execution (0/14)	Account Manipulation (0/6)	Build Image on Host	Credentials from Password Stores (0/6)	Browser Information Discovery	Lateral Tool Transfer	Audio Capture	Content Injection	Exfiltration Over Alternative Protocol (0/3)	Data Encrypted for Impact
Gather Victim Network Information (0/5)	Acquire Infrastructure (0/8)	Domains	Exploit Public-Facing Application	Cloud API	Boot or Logon Initialization Scripts (0/5)	Debugger Evasion	Exploitation for Credential Access	Cloud Infrastructure Discovery	Remote Service Session Hijacking (0/2)	Automated Collection	Data Encoding (0/2)		Data Manipulation (0/3)
Gather Victim Org Information (0/4)		Malvertising	External Remote Services	JavaScript	Browser Extensions	Deobfuscate/Decode Files or Information	Forced Authentication	Cloud Service Dashboard	Remote Services (0/8)	Browser Session Hijacking (0/3)	Data Obfuscation (0/2)	Exfiltration Over C2 Channel (0/3)	Defacement (0/2)
Phishing for Information (0/4)		Serverless	Hardware Additions	Network Device CLI	Compromise Host Software Binary	Deploy Container	Forge Web Credentials (0/2)	Cloud Service Discovery	Replication Through Removable Media	Clipboard Data	Dynamic Resolution (0/3)		Disk Wipe (0/2)
Search Closed Sources (0/2)	Compromise Accounts (0/3)			PowerShell	Create Account (0/2)	Direct Volume Access	Input Capture (0/4)	Cloud Storage Object Discovery	Software Deployment Tools	Data from Cloud Storage (0/2)	Encrypted Channel (0/2)	Exfiltration Over Other Network Medium (0/1)	Endpoint Denial of Service (0/4)
Search Open Technical Databases (0/5)	Compromise Infrastructure (0/6)	Phishing (1/4)	Spearphishing Link	Visual Basic	Create or Modify System Process (0/5)	Domain or Tenant Policy Modification (0/2)	Modify Authentication Process (0/9)	Container and Resource Discovery	Taint Shared Content	Data from Configuration Repository (0/2)	Fallback Channels	Exfiltration Over Physical Medium (0/1)	Financial Theft
Search Open Websites/Domains (0/3)	Develop Capabilities (0/4)		Spearphishing Voice	Windows Command Shell	Domain or Tenant Policy Modification (0/2)	Execution Guardrails (0/1)	Multi-Factor Authentication Interception	Debugger Evasion	Use Alternate Authentication Material (0/4)	Data from Information Repositories (0/3)	Hide Infrastructure	Exfiltration Over Web Service (0/4)	Firmware Corruption
Search Victim-Owned Websites	Establish Accounts (0/3)	Replication Through Removable Media		Container Administration Command	Event Triggered Execution (0/16)	Exploitation for Defense Evasion	Multi-Factor Authentication Interception	Device Driver Discovery		Data from Local System	Ingress Tool Transfer		Inhibit System Recovery
	Obtain Capabilities (0/7)	Supply Chain Compromise (0/3)		Deploy Container	External Remote Services	File and Directory Permissions Modification (0/2)	Multi-Factor Authentication Request Generation	Domain Trust Discovery		Data from Network Shared Drive	Multi-Stage Channels	Scheduled Transfer	Network Denial of Service (0/2)
	Stage Capabilities (0/6)	Trusted Relationship		Exploitation for Client Execution	Hijack Execution Flow (0/13)	Hide Artifacts (0/12)	Network Sniffing	File and Directory Discovery		Data from Removable Media	Non-Application Layer Protocol	Transfer Data to Cloud Account	Resource Hijacking
		Valid Accounts (0/4)		Inter-Process Communication (0/3)	Implant Internal Image	Impair Defenses (0/11)	OS Credential Dumping (0/6)	Group Policy Discovery		Data Staged (0/2)	Non-Standard Port		Service Stop
				Native API	Modify Authentication Process (0/9)	Indicator Removal (0/9)	Steal Application Access Token	Log Enumeration		Email Collection (0/3)	Protocol Tunneling		System Shutdown/Reboot
				Scheduled Task/Job (0/5)	Office Application Startup (0/6)	Scheduled Task/Job (0/10)	Steal or Forge Authentication Certificates	Network Service Discovery		Input Capture (0/4)	Proxy (0/4)		
				Serverless Execution	Power Settings	Valid Accounts	Masquerading (0/9)	Network Share Discovery		Screen Capture	Remote Access Software		
				Shared Modules	Pre-OS Boot (0/5)	Scheduled Task/Job (0/5)	Steal or Forge Kerberos Tickets (0/4)	Password Policy Discovery			Traffic Signaling		
				Software Deployment Tools	Server Software Component (0/5)	Traffic Signaling (0/2)	Modify Authentication Process (0/9)	Peripheral Device Discovery			Web Service (0/3)		
				System Services (0/2)	Malicious File	Valid Accounts (0/4)	Modify Cloud Compute Infrastructure (0/5)	Permission Groups Discovery (0/3)					
					Malicious Image		Modify Registry	Process Discovery					
					Malicious Link		Modify System Image (0/2)	Query Registry					
							Network Boundary Bridging (0/1)	Remote System Discovery					
							Obfuscated Files or Information (0/13)	System Information Discovery					
							Plist File Modification	System Location Discovery (0/1)					
							Pre-OS Boot (0/5)	System Network Configuration Discovery (0/2)					



Scenarios 2 APT Attack Combination (IPT and FBD)

APT 1: APT Attack Intellectual Property Theft (IPT)

- APT group: state-sponsored
 - Target: Leading semiconductor design company
 - Goal: Steal their chip designs
- Phishing campaign launched
 - Deceptive emails to key researchers and mid-level engineers
- Stolen credentials were leveraged to use valid accounts for network access
 - Valid accounts were continually exploited to ensure long-term access
- Attackers relied on OS credential dumping to gather more credentials
 - Facilitated lateral movement using remote services
- Over a long period of time, attackers collected sensitive data from information repositories that included important chip design documents and source code.





APT 2: APT Attack Financial Data Breach (FBD)

Overview of Financial Data Breach:

- APT group: state sponsored
 - Target: Multinational financial institution that handles about trillions in transactions daily
- Attackers exploited a vulnerability in the company's external VPN service
- They used previously stolen credentials as valid accounts to gain access to the network
- The execution of malicious scripts allowed them to attain a detailed layout of the company's digital architecture
- Attackers used different tools to harvest sensitive credentials
 - enabled them to move laterally across the network remotely
- Stayed hidden for months and exfiltrated large amounts of sensitive financial data using an encrypted command and control channel

Tactics & Techniques

APT 1: IPT

Initial Access: Phishing (T1566), Valid Accounts (T1078)

Execution: User Execution (T1204)

Persistence: Valid Accounts (T1078)

Credential Access: OS Credential Dumping (T1003)

Lateral Movement: Remote Services (T1021)

Collection: Data from Information Repositories (T1213)

APT 2: FBD

Initial Access: External Remote Services (T1133), Valid Accounts (T1078)

Execution: Scripting (T1059)

Persistence: Valid Accounts (T1078)

Credential Access: OS Credential Dumping (T1003)

Lateral Movement: Remote Services (T1021)

Exfiltration: Exfiltration Over C2 Channel (T1041)

Discovery: System Network Configuration Discovery (T1016)

Reconnaissance 10 techniques	Resource Development 8 techniques	Initial Access 10 techniques	Execution 14 techniques	Persistence 20 techniques	Privilege Escalation 14 techniques	Defense Evasion 43 techniques	Credential Access 17 techniques	Discovery 32 techniques	Lateral Movement 9 techniques	Collection 17 techniques	Command and Control 18 techniques	Exfiltration 9 techniques	Impact 14 techniques
Active Scanning (0/3)	Acquire Access	Content Injection	Cloud Administration Command	Account Manipulation (0/6)	Abuse Elevation Control Mechanism (0/6)	Abuse Elevation Control Mechanism (0/6)	Adversary-in-the-Middle (0/3)	Account Discovery (0/4)	Exploitation of Remote Services	Adversary-in-the-Middle (0/3)	Application Layer Protocol (0/4)	Automated Exfiltration (0/1)	Account Access Removal
Gather Victim Host Information (0/4)	Acquire Infrastructure (0/8)	Drive-by Compromise	Command and Scripting Interpreter (0/10)	BITS Jobs	Access Token Manipulation (0/5)	Access Token Manipulation (0/5)	Brute Force (0/4)	Application Window Discovery	Internal Spearphishing	Archive Collected Data (0/3)	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction
Gather Victim Identity Information (0/3)	Compromise Accounts (0/3)	Exploit Public-Facing Application	Container Administration Command	Boot or Logon Autostart Execution (0/14)	Account Manipulation (0/6)	BITS Jobs	Credentials from Password Stores (0/6)	Browser Information Discovery	Lateral Tool Transfer	Audio Capture	Content Injection	Exfiltration Over Alternative Protocol (0/3)	Data Encrypted Impact
Gather Victim Network Information (0/6)	Compromise Infrastructure (0/8)	External Remote Services	Deploy Container	Boot or Logon Initialization Scripts (0/5)	Boot or Logon Autostart Execution (0/14)	Debugger Evasion	Exploitation for Credential Access	Cloud Infrastructure Discovery	Remote Service Session Hijacking (0/2)	Automated Collection	Data Encoding (0/2)	Exfiltration Over C2 Channel	Data Manipulation
Gather Victim Org Information (0/4)	Develop Capabilities (0/4)	Hardware Additions	Exploitation for Client Execution	Browser Extensions	Boot or Logon Initialization Scripts (0/5)	Deobfuscate/Decode Files or Information	Forced Authentication	Cloud Service Dashboard	Remote Services (0/5)	Browser Session Hijacking	Data Obfuscation (0/3)	Exfiltration Over Other Network Medium (0/1)	Defacement (0/2)
Phishing for Information (0/4)	Establish Accounts (0/3)	Phishing (0/4)	Inter-Process Communication (0/3)	Compromise Host Software Binary	Create or Modify System Process (0/5)	Deploy Container	Forge Web Credentials (0/2)	Cloud Service Discovery	Replication Through Removable Media	Clipboard Data	Dynamic Resolution (0/3)	Exfiltration Over Physical Medium (0/1)	Disk Wipe (0/2)
Search Closed Sources (0/2)	Obtain Capabilities (0/7)	Replication Through Removable Media	Native API	Create Account (0/3)	Create or Modify System Process (0/5)	Direct Volume Access	Input Capture (0/4)	Cloud Storage Object Discovery	Software Deployment Tools	Data from Cloud Storage	Encrypted Channel (0/2)	Exfiltration Over Physical Medium (0/1)	Endpoint Denial Service (0/4)
Search Open Technical Databases (0/5)	Stage Capabilities (0/6)	Supply Chain Compromise (0/3)	Scheduled Task/Job (0/5)	Event Triggered Execution (0/16)	Domain or Tenant Policy Modification (0/2)	Execution Guardrails (0/1)	Modify Authentication Process (0/9)	Container and Resource Discovery	Taint Shared Content	Data from Configuration Repository (0/2)	Fallback Channels	Exfiltration Over Web Service (0/4)	Financial Theft
Search Open Websites/Domains (0/3)		Trusted Relationship	Serverless Execution	External Remote Services	Domain or Tenant Policy Modification (0/2)	Exploitation for Defense Evasion	Multi-Factor Authentication Interception	Debugger Evasion	Use Alternate Authentication Material (0/4)	Data from Information Repositories (0/3)	Hide Infrastructure	Scheduled Transfer	Firmware Corruption
Search Victim-Owned Websites		Valid Accounts (0/4)	Shared Modules	Event Triggered Execution (0/16)	Escape to Host	File and Directory Permissions Modification (0/2)	Multi-Factor Authentication Request Generation	Device Driver Discovery		Data from Local System	Ingress Tool Transfer	Transfer Data to Cloud Account	Network Denial Service (0/2)
			Software Deployment Tools	Hijack Execution Flow (0/13)	Event Triggered Execution (0/16)	Hijack Execution Flow (0/13)	OS Credential Dumping (0/6)	Domain Trust Discovery		Data from Network Shared Drive	Multi-Stage Channels		Resource Hijacking
			System Services (0/2)	Implant Internal Image	Exploitation for Privilege Escalation	Hijack Execution Flow (0/13)	OS Credential Dumping (0/6)	File and Directory Discovery		Data from Removable Media	Non-Application Layer Protocol		Service Stop
			User Execution (0/3)	Modify Authentication Process (0/9)	Hijack Execution Flow (0/13)	Impair Defenses (0/11)	Steal Application Access Token	Group Policy Discovery		Data Staged (0/2)	Non-Standard Port		System Shutdown/Reboot
			Windows Management Instrumentation	Office Application Startup (0/6)	Process Injection (0/12)	Indicator Removal (0/9)	Steal or Forge Authentication Certificates	Log Enumeration		Email Collection (0/3)	Protocol Tunneling		
				Power Settings	Scheduled Task/Job (0/5)	Indirect Command Execution	Steal or Forge Kerberos Tickets (0/4)	Network Service Discovery		Input Capture (0/4)	Proxy (0/4)		
				Pre-OS Boot (0/5)	Valid Accounts (0/4)	Masquerading (0/9)	Steal Web Session Cookie	Network Share Discovery		Screen Capture	Remote Access Software		
				Scheduled Task/Job (0/5)		Modify Authentication Process (0/9)	Unsecured Credentials (0/8)	Network Sniffing		Video Capture	Traffic Signaling (0/2)		
				Server Software Component (0/5)		Modify Cloud Compute Infrastructure (0/5)		Password Policy Discovery			Web Service (0/3)		
				Traffic Signaling (0/2)		Modify Registry		Peripheral Device Discovery					
				Valid Accounts (0/4)		Modify System Image (0/2)		Permission Groups Discovery (0/3)					
						Network Boundary Bridging (0/1)		Process Discovery					
						Obfuscated Files or Information (0/13)		Query Registry					
						Plist File Modification		Remote System Discovery					
						Pre-OS Boot (0/5)		Software Discovery (0/1)					
								System Information Discovery					
								System Location Discovery (0/1)					
								System Network Configuration Discovery (0/2)					
								System Network Connections Discovery					

Importance of Our Tool: GIGL

- GIGL converts MITRE ATT&CK Navigator layers into Bayesian networks, providing scenario-specific insights
- Enables efficient analysis - mapping
- Facilitates strategic planning through quantification
- Supports adaptive defense mechanisms - updates/refinements
- Enhances defensive coverage assessment
- Enables real-life implementation by integrating with existing security operations





Limitations & Future works

Limitations

- Simple probability assignments
- The Bayesian Network model is static
- The current approach relies on a predefined mapping of technique IDs to technique names using Stix.
- The model does not incorporate real-world attack data or incident reports.
- Simplified AND-OR gateway relationships

Future work

- Incorporating real-world data
- Advanced probability assignment
- Dynamic and adaptive models