CYSE 650 – Project GIGL

CYSE 650-D01

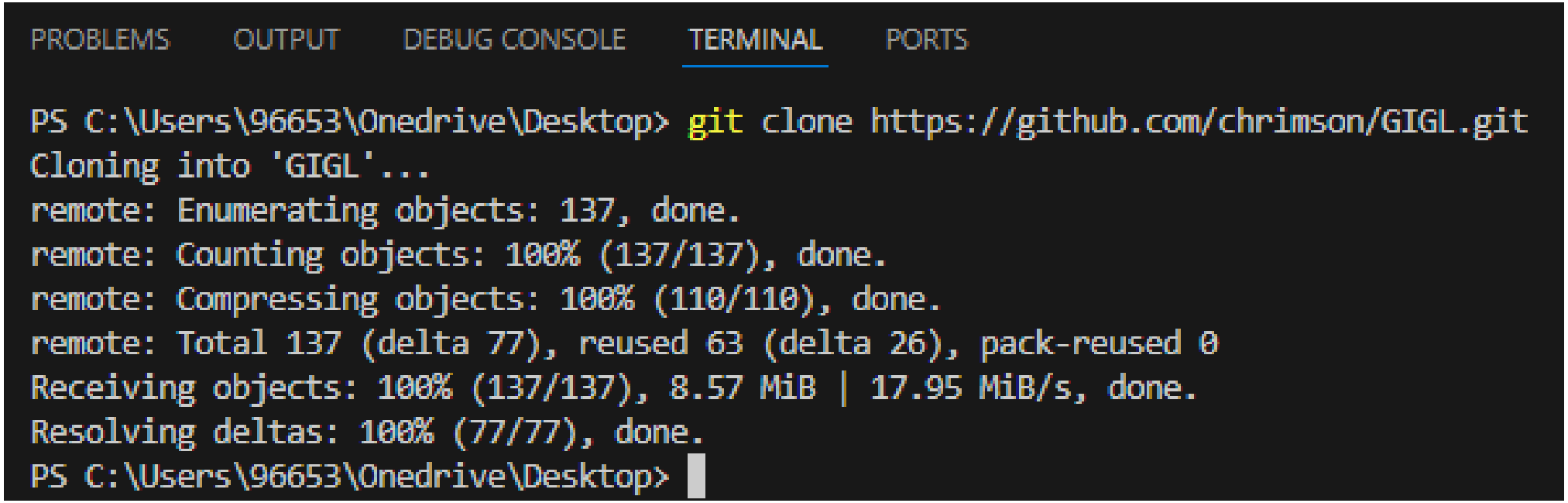
07/18/2024

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

Project Installation Tutorial

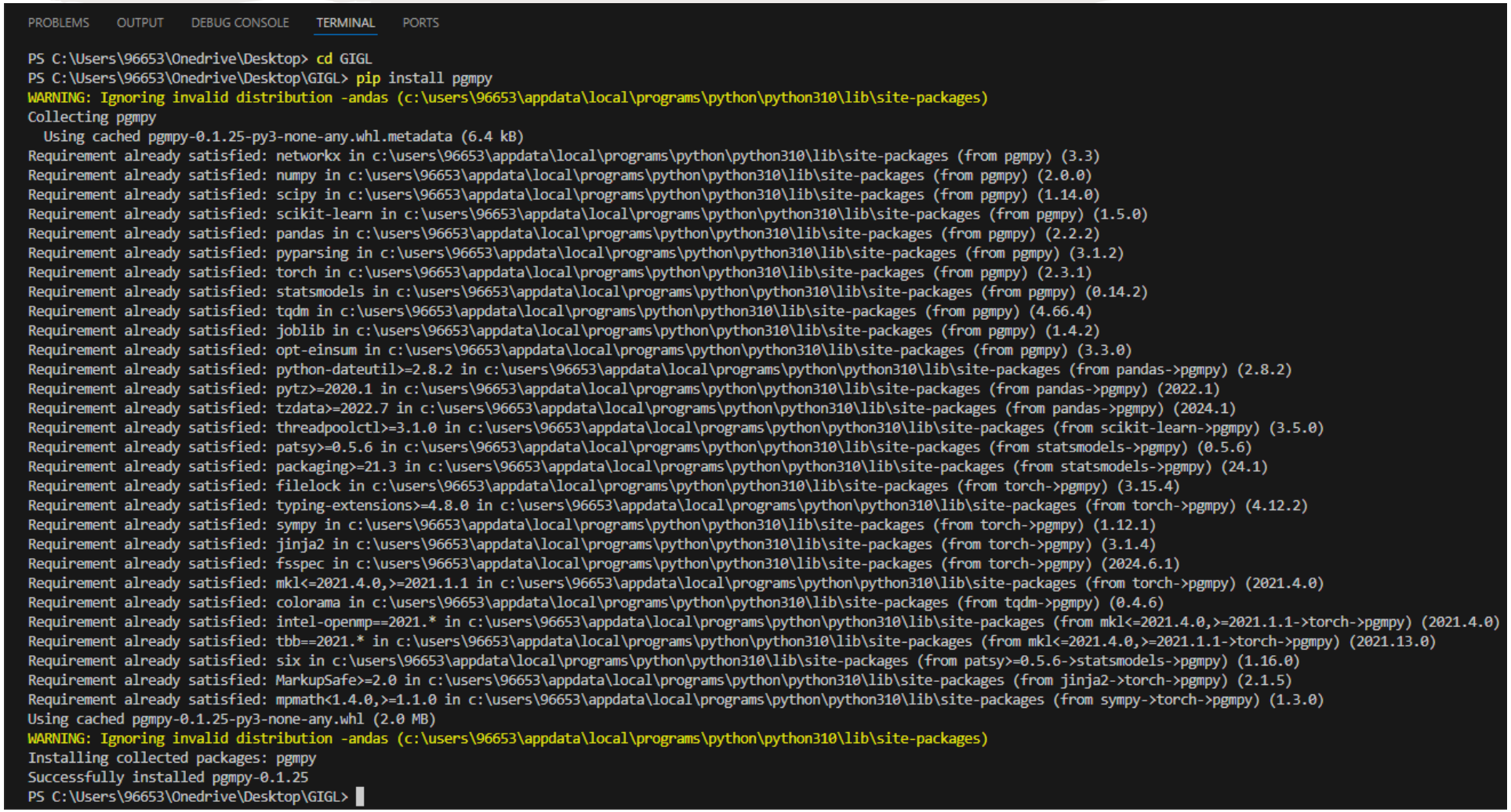
Downloading the repository

Ran the command: ***git clone*** ***https://github.com/chrimson/GIGL.git*****.**



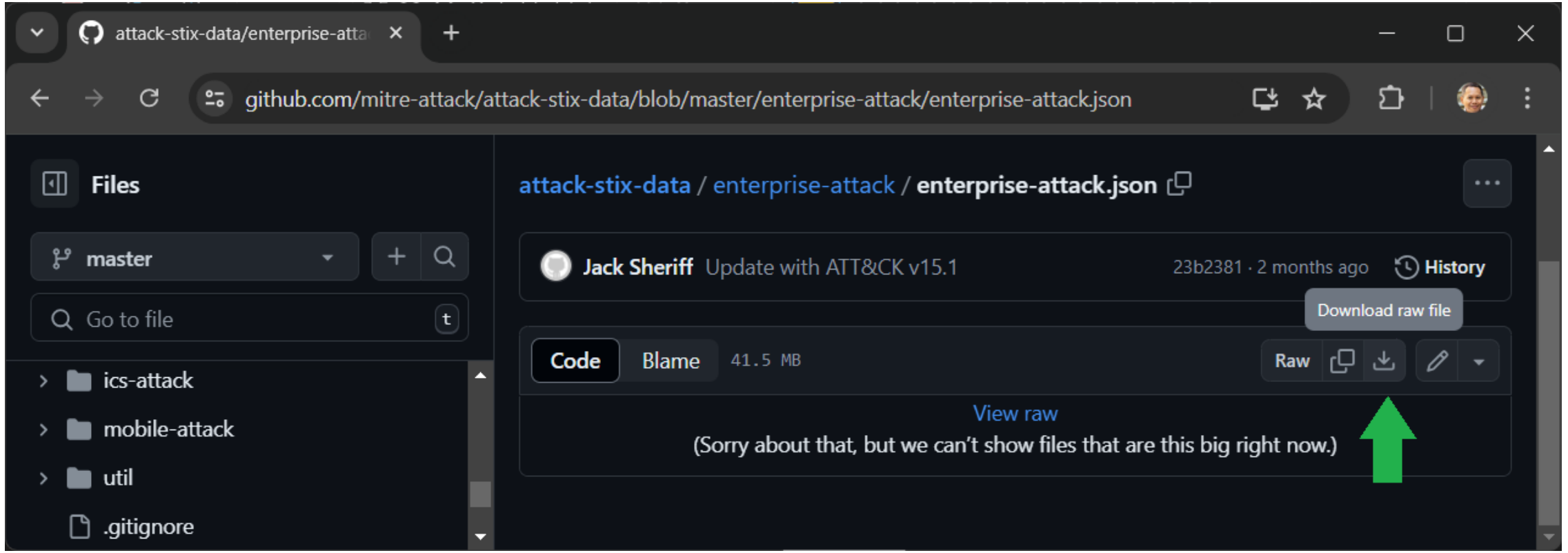
Enabled required decency

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



Configure (optional)

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



https://github.com/mitre-attack/attack-stix-data/blob/master/enterprise-attack/enterprise-attack-15.1.json

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

```python
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

```python
# 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)
```

```python
# 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

```python
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
```

```python
            # 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

```python
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

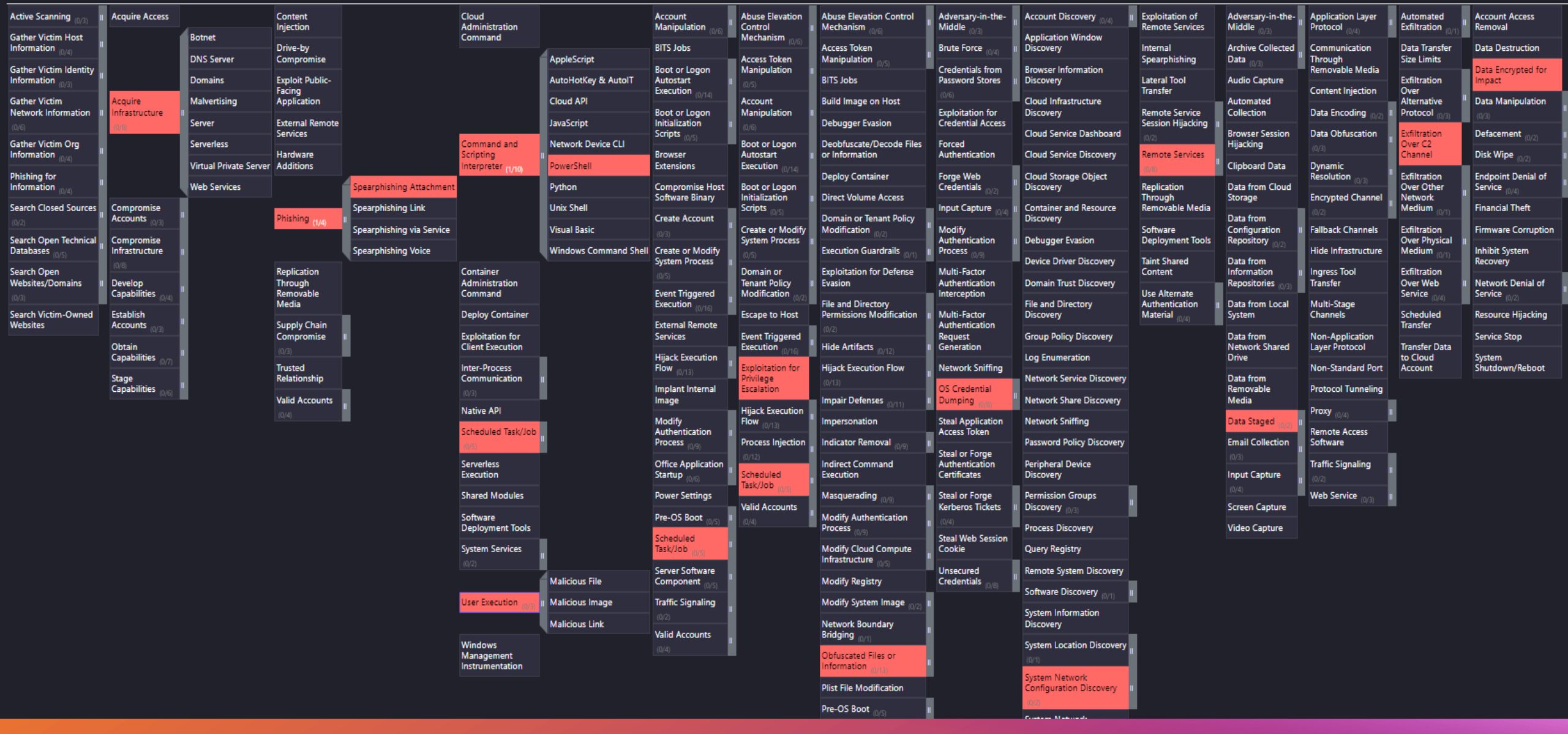
- **Ransomware attack** targets financial services company
- Spear phishing email containing malicious excel file that executes a powershell script when opened
- Powershell script downloads additional malware
  - Scheduled task
- Zero-day vulnerability exploited - gain higher privileges
- Stole credentials 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)

Finnacial Ransomware Attack

selection controls | layer controls | technique controls



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

MITRE ATT&CK™

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)

APT: IPT + FDB

selection controls | layer controls | technique controls

Reconnaissance
10 techniques

- Active Scanning (0/3)
- Gather Victim Host Information (0/4)
- Gather Victim Identity Information (0/3)
- Gather Victim Network Information (0/6)
- Gather Victim Org Information (0/4)
- Phishing for Information (0/4)
- Search Closed Sources (0/2)
- Search Open Technical Databases (0/5)
- Search Open Websites/Domains (0/3)
- Search Victim-Owned Websites

Resource Development
8 techniques

- Acquire Access
- Acquire Infrastructure (0/8)
- Compromise Accounts (0/3)
- Compromise Infrastructure (0/8)
- Develop Capabilities (0/4)
- Establish Accounts (0/3)
- Obtain Capabilities (0/7)
- Stage Capabilities (0/6)

Initial Access
10 techniques

- Content Injection
- Drive-by Compromise
- Exploit Public-Facing Application
- External Remote Services
- Hardware Additions
- Phishing (0/4)
- Replication Through Removable Media
- Supply Chain Compromise (0/3)
- Trusted Relationship
- Valid Accounts (0/4)

Execution
14 techniques

- Cloud Administration Command
- Command and Scripting Interpreter (0/10)
- Container Administration Command
- Deploy Container
- Exploitation for Client Execution
- Inter-Process Communication (0/3)
- Native API
- Scheduled Task/Job (0/5)
- Serverless Execution
- Shared Modules
- Software Deployment Tools
- System Services (0/2)
- User Execution (0/3)
- Windows Management Instrumentation

Persistence
20 techniques

- Account Manipulation (0/6)
- BITS Jobs
- Boot or Logon Autostart Execution (0/14)
- Boot or Logon Initialization Scripts (0/5)
- Browser Extensions
- Compromise Host Software Binary
- Create Account (0/3)
- Create or Modify System Process (0/5)
- Event Triggered Execution (0/16)
- External Remote Services
- Hijack Execution Flow (0/13)
- Implant Internal Image
- Modify Authentication Process (0/9)
- Office Application Startup (0/6)
- Power Settings
- Pre-OS Boot (0/5)
- Scheduled Task/Job (0/5)
- Server Software Component (0/5)
- Traffic Signaling (0/2)
- Valid Accounts (0/4)

Privilege Escalation
14 techniques

- Abuse Elevation Control Mechanism (0/6)
- Access Token Manipulation (0/5)
- Account Manipulation (0/6)
- Boot or Logon Autostart Execution (0/14)
- Boot or Logon Initialization Scripts (0/5)
- Create or Modify System Process (0/5)
- Domain or Tenant Policy Modification (0/2)
- Escape to Host
- Event Triggered Execution (0/16)
- Exploitation for Privilege Escalation
- Hijack Execution Flow (0/13)
- Process Injection (0/12)
- Scheduled Task/Job (0/5)
- Valid Accounts (0/4)

Defense Evasion
43 techniques

- Abuse Elevation Control Mechanism (0/6)
- Access Token Manipulation (0/5)
- BITS Jobs
- Build Image on Host
- Debugger Evasion
- Deobfuscate/Decode Files or Information
- Deploy Container
- Direct Volume Access
- Domain or Tenant Policy Modification (0/2)
- Execution Guardrails (0/1)
- Exploitation for Defense Evasion
- File and Directory Permissions Modification (0/2)
- Hide Artifacts (0/12)
- Hijack Execution Flow (0/13)
- Impair Defenses (0/11)
- Impersonation
- Indicator Removal (0/9)
- Indirect Command Execution
- Masquerading (0/9)
- Modify Authentication Process (0/9)
- Modify Cloud Compute Infrastructure (0/5)
- Modify Registry
- Modify System Image (0/2)
- Network Boundary Bridging (0/1)
- Obfuscated Files or Information (0/13)
- Plist File Modification
- Pre-OS Boot (0/5)

Credential Access
17 techniques

- Adversary-in-the-Middle (0/3)
- Brute Force (0/4)
- Credentials from Password Stores (0/6)
- Exploitation for Credential Access
- Forced Authentication
- Forge Web Credentials (0/2)
- Input Capture (0/4)
- Modify Authentication Process (0/9)
- Multi-Factor Authentication Interception
- Multi-Factor Authentication Request Generation
- Network Sniffing
- OS Credential Dumping (0/8)
- Steal Application Access Token
- Steal or Forge Authentication Certificates
- Steal or Forge Kerberos Tickets (0/4)
- Steal Web Session Cookie
- Unsecured Credentials (0/8)

Discovery
32 techniques

- Account Discovery (0/4)
- Application Window Discovery
- Browser Information Discovery
- Cloud Infrastructure Discovery
- Cloud Service Dashboard
- Cloud Service Discovery
- Cloud Storage Object Discovery
- Container and Resource Discovery
- Debugger Evasion
- Device Driver Discovery
- Domain Trust Discovery
- File and Directory Discovery
- Group Policy Discovery
- Log Enumeration
- Network Service Discovery
- Network Share Discovery
- Network Sniffing
- Password Policy Discovery
- Peripheral Device Discovery
- Permission Groups Discovery (0/3)
- Process Discovery
- Query Registry
- Remote System Discovery
- Software Discovery (0/1)
- System Information Discovery
- System Location Discovery (0/1)
- System Network Configuration Discovery (0/2)
- System Network Connections Discovery

Lateral Movement
9 techniques

- Exploitation of Remote Services
- Internal Spearphishing
- Lateral Tool Transfer
- Remote Service Session Hijacking (0/2)
- Remote Services (0/6)
- Replication Through Removable Media
- Software Deployment Tools
- Taint Shared Content
- Use Alternate Authentication Material (0/4)

Collection
17 techniques

- Adversary-in-the-Middle (0/3)
- Archive Collected Data (0/3)
- Audio Capture
- Automated Collection
- Browser Session Hijacking
- Clipboard Data
- Data from Cloud Storage
- Data from Configuration Repository (0/2)
- Data from Information Repositories (0/3)
- Data from Local System
- Data from Network Shared Drive
- Data from Removable Media
- Data Staged (0/2)
- Email Collection (0/3)
- Input Capture (0/4)
- Screen Capture
- Video Capture

Command and Control
18 techniques

- Application Layer Protocol (0/4)
- Communication Through Removable Media
- Content Injection
- Data Encoding (0/2)
- Data Obfuscation (0/3)
- Dynamic Resolution (0/3)
- Encrypted Channel (0/2)
- Fallback Channels
- Hide Infrastructure
- Ingress Tool Transfer
- Multi-Stage Channels
- Non-Application Layer Protocol
- Non-Standard Port
- Protocol Tunneling
- Proxy (0/4)
- Remote Access Software
- Traffic Signaling (0/2)
- Web Service (0/3)

Exfiltration
9 techniques

- Automated Exfiltration (0/1)
- Data Transfer Size Limits
- Exfiltration Over Alternative Protocol (0/3)
- Exfiltration Over C2 Channel
- Exfiltration Over Other Network Medium (0/1)
- Exfiltration Over Physical Medium (0/1)
- Exfiltration Over Web Service (0/4)
- Scheduled Transfer
- Transfer Data to Cloud Account

Impact
14 techniques

- Account Access Removal
- Data Destruction
- Data Encrypted for Impact
- Data Manipulation
- Defacement (0/2)
- Disk Wipe (0/2)
- Endpoint Denial of Service (0/4)
- Financial Theft
- Firmware Corruption
- Inhibit System Recovery
- Network Denial of Service (0/2)
- Resource Hijacking
- Service Stop
- System Shutdown/Reboot

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

Figure 1 - Project Solution

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