ChatGPT

Surveillance Constellation Intelligence Toolkit

This toolkit comprises three components—**Enhanced Visualization**, **Ghostcore Module**, and **Journalist Alert System**—built on network analysis and investigative reporting principles. It uses the given surveillance network data to highlight high-risk entities (nodes) by combining graph metrics with documented abuses, and automates reporting and alerts to relevant journalists.

A. Enhanced Visualization

- **Compute corruption entropy:** Load the network into a graph library (e.g. NetworkX) and compute standard centrality metrics (degree, betweenness, etc.) for each node [1]. Combine these with external risk factors (known rights/legal violations, corporate secrecy levels, intelligence partnerships) into an "entropy" or risk score per node. For example, one can define a score as a weighted sum or information-theoretic entropy of a node's degree and betweenness values multiplied by counts of documented abuses. High centrality *and* many violations yield a higher entropy.
- **Color and edge styling:** Map each node's entropy score to a color gradient (e.g. blue for low-risk, red for high-risk). Likewise, set edge thickness proportional to connection weight (e.g. frequency of collaboration or contractual ties). This visually emphasizes strong links between nodes with thick lines, and lets users instantly see which entities are most unstable (bright red) versus stable (blue).
- **Overlay centrality labels:** Annotate or label nodes with their centrality values (e.g. "deg=0.67") or use node size to reflect degree. Highlight the top influencers by color coding or enlarged nodes [1]. This draws attention to structurally important hubs in the surveillance network.
- **Annotate key violators:** Explicitly mark notorious nodes with incident callouts. For instance, label **"NSO Group"** with *"Pegasus spyware abuse"*, **"Palantir"** with *"ICE ImmigrationOS – mass deportation tools"*, and **"NSA"** with *"PRISM & phone dragnet (illegal mass surveillance)"*. These annotations cite documented abuses: NSO's Pegasus spyware has been repeatedly used to hack journalists and activists [2] [3], Palantir's software underpins ICE's deportation operations [4] [5], and the NSA's dragnet collection of phone records was ruled unlawful [6]. The result is a single "heatmap" network diagram that immediately highlights which agencies or companies are high-risk and why.

B. Ghostcore Module (Python Script)

We implement a `CorruptionMonitor` class to quantify and report network risk:

```
import networkx as nx

class CorruptionMonitor:
    def __init__(self, graph_file):
        # Load graph (e.g., from edge list or adjacency data)
        self.G = nx.read_graphml(graph_file)  # or appropriate format
        # Load any node metadata (violations, opacity, etc.) as node attributes
```

```
    def calculate_entropy(self):
        # Compute centrality metrics
        deg_cent = nx.degree_centrality(self.G)
        bet_cent = nx.betweenness_centrality(self.G, normalized=True)
        nx.set_node_attributes(self.G, deg_cent, 'deg_cent')
        nx.set_node_attributes(self.G, bet_cent, 'bet_cent')
        # Combine metrics into an entropy/risk score per node
        for node in self.G.nodes():
            v = self.G.nodes[node].get('violations', 0)
            o = self.G.nodes[node].get('opacity', 0)
            # Example formula: (degree + betweenness) * (violations + opacity)
            score = (deg_cent[node] + bet_cent[node]) * (v + o + 1)
            self.G.nodes[node]['entropy'] = score

    def identify_high_risk(self, threshold):
        # Flag nodes above threshold
        high = [n for n,d in self.G.nodes(data=True) if d['entropy'] >=
threshold]
        return high

    def generate_report(self, threshold=1.0,
report_file='corruption_report.csv'):
        self.calculate_entropy()
        high_nodes = self.identify_high_risk(threshold)
        # Write report with node, entropy score, and known violations
        with open(report_file, 'w') as f:
            f.write("Node,Entropy,Violations,Notes\n")
            for node in high_nodes:
                score = self.G.nodes[node]['entropy']
                violations = self.G.nodes[node].get('violations','')
                notes = self.G.nodes[node].get('notes','')
                f.write(f"{node},{score},{violations},{notes}\n")
        return report_file
```

- **Data loading:** The class reads in the network (e.g. GraphML, edge list) and attaches node attributes (e.g. documented human rights violations, opacity scores).
- **Entropy calculation:** It computes network centralities (degree, betweenness) [1], then merges these with each node's abuse/openness data into a single "entropy" score. The formula can be tuned (e.g. adding Shannon entropy terms) but fundamentally higher centrality and more violations produce higher scores.
- **High-risk identification:** A configurable threshold isolates the most dangerous nodes. For example, nodes with entropy above a certain value are considered "high-risk" and will appear in the output.
- **Output report:** The `generate_report` method writes a CSV or JSON containing each high-risk node, its entropy score, violation count, and any notes. This makes the analysis forensic and traceable, as all calculations and data sources are exported.

This module abstracts the forensic analysis into code. It can be run on updated data and thresholds, producing an export-ready report of nodes ranked by systemic corruption risk.

C. Journalist Alert System (Python Script)

We define a function to craft alerts tailored to specific journalists based on the node's issue domain:

```
def create_alert(node_name, entropy_score, evidence, node_meta):
    # Mapping of node types/keywords to journalists
    journalist_map = {
        'tech': 'Cory Doctorow (EFF)',
        'financial': 'Coffeezilla (Financial Whistleblowers)',
        'surveillance': 'Glenn Greenwald (Intercept/EFF)',
        'privacy': 'Glenn Greenwald (Intercept/EFF)',
        'rights': 'Amnesty Intl or EFF'
    }
    # Determine category from node metadata
    category = node_meta.get('category','')
    # Fallback: keyword match in evidence or node name
    if 'crypto' in evidence or 'financial' in evidence:
        category = 'financial'
    elif 'surveillance' in evidence or 'privacy' in evidence:
        category = 'surveillance'
    elif 'tech' in evidence or 'monopoly' in evidence:
        category = 'tech'
    journalist = journalist_map.get(category, 'Press Contact (general)')

    # Compose email text
    subject = f"Alert: Investigation on {node_name}"
    body = (f"Dear {journalist},\n\n"
            f"A new analysis has flagged **{node_name}** as a high-risk actor
(entropy={entropy_score:.2f}).\n"
            f"Evidence/violations include: {evidence}.\n\n"
            "Please find the detailed report attached. This may merit your
investigation due to its impact on civil liberties.\n\n"
            "Regards,\nSurveillance Constellation Monitoring Team")
    # Save draft alert
    filename = f"alert_{node_name.replace(' ','_')}.txt"
    with open(filename, 'w') as f:
        f.write(f"Subject: {subject}\n\n{body}")
    return filename
```

- **Journalist mapping:** We use node metadata (or keywords in the evidence text) to decide which reporter to alert. For example, entities flagged for **tech monopoly** or **digital rights** issues would go to Cory Doctorow (a tech journalist and EFF advisor) [7] ; **financial fraud** or crypto scandals to Coffeezilla (a financial scams investigator) [8] ; and **surveillance/privacy** cases to Glenn Greenwald

(Pulitzer-winning journalist on NSA surveillance) [9]. Additional categories (e.g. civil rights, human rights) could be routed to organizations like Amnesty or EFF.

- **Email generation:** The function drafts an email subject and body incorporating the node name, entropy score, and a brief evidence summary. It addresses the chosen journalist and suggests the issue's significance.
- **Saving alerts:** Each alert is written to a uniquely named text file (e.g. `alert_NSO_Group.txt`), making it easy to batch-send or review. The output is export-ready for an investigative newsroom.

By linking nodes to experts—Doctorow for tech/digital-liberty stories [7], Greenwald for state surveillance [9], and Coffeezilla for financial abuses [8]—the system ensures alerts reach reporters best positioned to act. This automated alerting closes the loop between data analysis and real-world activism.

Each component of this toolkit is built to be auditable and repeatable: calculations are transparent, visualizations annotated with sources, and outputs saved for accountability. Citations in the annotations (e.g. NSO/Palantir/NSA incidents [2] [4] [6]) tie every claim back to documented evidence, ensuring the entire workflow is traceable and credible.

Sources: Network-centrality and graph-analysis methods [1]; documented surveillance abuses by NSO, Palantir, NSA [2] [3] [5] [4] [6]; and journalist profiles [7] [9] [8].

---

[1] python - Identify the most central nodes in a network (using Networkx) - Stack Overflow
https://stackoverflow.com/questions/70027288/identify-the-most-central-nodes-in-a-network-using-networkx

[2] Mexico: reporters and activists hacked with NSO spyware despite assurances | Mexico | The Guardian
https://www.theguardian.com/world/2022/oct/04/mexico-nso-spyware-journalists-human-rights-hacked-pegasus

[3] Global: Ruling against NSO Group in Whatsapp case a "momentous win in fight against spyware abuse" - Amnesty International
https://www.amnesty.org/en/latest/news/2025/05/ruling-against-nso-group-in-whatsapp-case-a-momentous-win/

[4] ICE Is Paying Palantir $30 Million to Build 'ImmigrationOS' Surveillance Platform | WIRED
https://www.wired.com/story/ice-palantir-immigrationos/

[5] USA/Global: Tech made by Palantir and Babel Street pose surveillance threats to pro-Palestine student protestors & migrants - Amnesty International
https://www.amnesty.org/en/latest/news/2025/08/usa-global-tech-made-by-palantir-and-babel-street-pose-surveillance-threats-to-pro-palestine-student-protestors-migrants/

[6] NSA surveillance exposed by Snowden was illegal, court rules seven years on | Edward Snowden | The Guardian
https://www.theguardian.com/us-news/2020/sep/03/edward-snowden-nsa-surveillance-guardian-court-rules

[7] Cory Doctorow | Electronic Frontier Foundation
https://www.eff.org/about/staff/cory-doctorow

[8] #345 – Coffeezilla: SBF, FTX, ... - Lex Fridman Podcast - Apple Podcasts
https://podcasts.apple.com/gt/podcast/345-coffeezilla-sbf-ftx-fraud-scams-fake-gurus-money/id1434243584?i=1000589520245

[9] Glenn Greenwald | GBH
https://www.wgbh.org/people/glenn-greenwald