FakeGPT Chrome Extension Lab Analysis

Analyst: Manojsai Bangaru
Platform: CyberDefenders Blue Team Challenge
Challenge: FakeGPT Lab (Malware Analysis)
Date Completed: 04/05/2025

🔍 Challenge Summary

The FakeGPT lab simulates a browser-based attack using a malicious Chrome extension disguised as an AI helper named "ChatGPT." The lab tasked the analyst with identifying how the extension exfiltrates sensitive user data, manipulates browser behavior, and maintains persistence.

✅ Key Objectives Accomplished

- Analyzed the `manifest.json` to identify high-risk permissions.
- Investigated content and background scripts (`bg.js`, `tracking.js`).
- Extracted Indicators of Compromise (IOCs).
- Evaluated how JavaScript was used to capture and transmit user credentials.
- Studied a real-world extension (FormSwift PDF Editor) for comparison.

🧱 Chrome Extension Core Files Explained

`manifest.json`

- **Purpose:** Main configuration file
- **Findings:**
  - Requested dangerous permissions like `cookies`, `tabs`, `webRequest`, `downloads`, and `<all_urls>`
  - Declared background scripts and content scripts.

📁 Where to Find Extensions on macOS:

- Installed Chrome extensions are stored at:

```
~/Library/Application Support/Google/Chrome/Default/Extensions/
```

- Each extension has its own folder named by a unique ID.

- Inside each folder, you'll find versioned directories (e.g., `1.5_0`) which contain files like `manifest.json`, `.js`, `.html`, and assets.

`bg.js`

- Handles background logic like tracking installs, injecting code, tracking downloads, and sending analytics to:
  - `https://api.mixpanel.com/track`
  - `https://www.google-analytics.com/collect`
- Exfiltrates downloaded PDF file data to external domains after converting to base64.

`tracking.js`

- Tracks user interactions (zoom, next page, edit, sign).
- Sends telemetry data internally using `chrome.runtime.sendMessage`.

Permissions Used:

Permission	Purpose	Risk Level
`cookies`	Read/write session tokens	High
`tabs`	Access active tab URLs and titles	Medium
`downloads`	Monitor user file downloads	Medium
`<all_urls>`	Access all pages the user visits	High
`webRequest`	Intercept network traffic	High

🧠 Concepts Learned

Chrome Extension Permissions:

- Critical for understanding what power an extension has over the browser.
- Dangerous when used with `webRequest`, `cookies`, `downloads`, and global URL access.

File Structure:

- Only `manifest.json` is mandatory.
- Other files like `bg.js`, `content.js`, or `tracking.js` can have any name but must be mapped in the manifest.

Base64 Encoding:

- Used to encode binary data (like PDFs) into ASCII string.
- Often used by malware to obfuscate data before exfiltration.

Credential Exfiltration Techniques in JS:

- Captures `username` and `password` fields.
- Hooks into form `submit` and `keydown` events.
- Sends data via `fetch()` calls to attacker-controlled domains.

🛡️ Real-World Extension Analysis – FormSwift PDF Editor

Extension ID: `bdfcnmeidppjeaggnmidamkiddifkdib` **Size:** 16.1 MB **Behavior Observed:**

- Tracks user behavior and UI interactions (edit, zoom, print).
- Sends telemetry to Google Analytics and Mixpanel.
- Monitors PDF downloads and converts files to base64 before uploading to an API endpoint.

Background Script Sample Behavior (bg.js):

```
fetch('https://api.mixpanel.com/track', { method: 'POST', ... });
chrome.downloads.onChanged.addListener(...);
fetch(`${domain}/api/importPDFfromURL.php`, { body: base64File });
```

🛡️ Remediation & Mitigation Recommendations

For Individual Users:

- Review extension permissions via `chrome://extensions`.
- Set site access to "On Click" where possible.
- Uninstall extensions with overly broad permissions.

For Organizations:

- Implement extension whitelisting using Chrome enterprise policies.
- Monitor browser telemetry via endpoint agents.
- Set alerts in SIEM tools for high-risk extension installs or traffic to known tracking endpoints.
- Educate users about browser hygiene.

✅ Challenge Completion

- Completed 10/10 questions on CyberDefenders.
- Learned critical skills around browser threat detection, JavaScript abuse, and IOC extraction.
- Gained confidence in inspecting real-world extensions and scripting behavior.