KARAMBIT.AI

Software Behavior Assurance for Safety and Cybersecurity

DISA TEM 09-16-2025

Andrew Hendela
andrew@karambit.ai

BLUF:

- Attackers are increasingly compromising software supply chains, including DoD
- This hurts Warfighter readiness and makes it risky to onboard new and innovative software
- Karambit.AI makes the software supply chain safe by verifying software behaviors for safety, cybersecurity, and effectiveness, without source code or execution. This increases safety, decreases time to field software, and eliminates remediation costs by stopping problems before they impact production and mission success
- Currently Awardable in the Platform One Marketplace
  - https://acqbot.niprgpt.mil/marketplaces/36da839d-0b76-47af-bffa-ad5c70ec3cc0/d9e573f8-6797-4a61-a587-076a25f5bb1d
- TRL 9: Microsoft hidden behavior detection validated in production, scanning over *4 billion* files per month; TRL 7: Linux behavior analysis of docker images, drone autopilot and medical device firmware safety and efficacy subsystem analysis
- Past Performance: JFAC OTA (finished) for AF Platform One docker container behavior scanning; ARPA-H contract for cybersecurity and safety of medical device firmware

Attackers abusing trusted software to cause billions of dollars in damage

SolarWinds is 'largest' cyberattack ever, Microsoft president says

The hack sent malware to about 18,000 public and private organizations.

3CX Software Supply Chain Compromise Initiated by a Prior Software Supply Chain Compromise; Suspected North Korean Actor Responsible

China compromised F-35 subcontractor and forced expensive software system rewrite, academic tells MPs

CSIS policy wonk describes supply cha

SUPPLY CHAIN SECURITY

XZ Utils Backdoor Attack Brings Another Similar Incident to Light

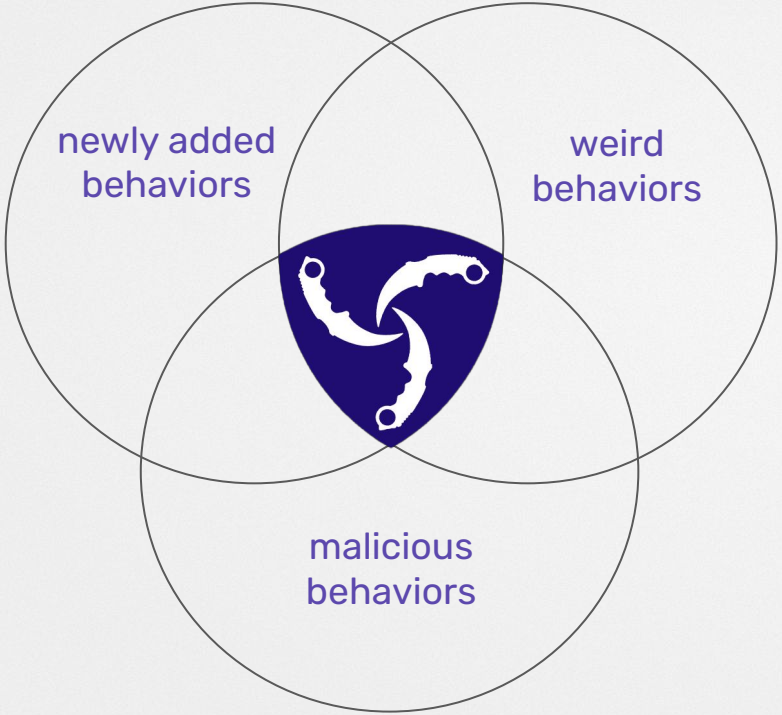
The discovery of the XZ Utils backdoor reminds an F-Droid developer of a similar incident that occurred a few years ago.

If I use this software, will it cause problems?

Karambit.AI makes the software supply chain safe by validating software behaviors and confirming developer intent

Solution

Automated Reverse Engineering to provide a Software Bill of Behaviors



We discover malicious or safety-critical behaviors hidden in your software **before deployment,** keeping customers from being compromised

“Karambit’s use of a content's historical functionality and changes, set itself apart from traditional static analysis.”

Principal Product Manager
Mike Bush

Detecting XZ Attack

Bill of Behaviors

DATA-MANIPULATION

• encode data using Base64	0.033%
• encode data using XOR	0.151%
• encrypt data using RC4 PRGA	0.056%
• hash data using murmur3	0.029%

HOST-INTERACTION

• create thread	0.159%
• get system information on Linux	0.017%
• read file on Linux	0.26%

Benign XZ 5.4.5

Bill of Behaviors

DATA-MANIPULATION

• encode data using XOR	0.151%
• encrypt data using RC4 PRGA	0.056%
• hash data using murmur3	0.029%

OTHER

• allocates memory dynamically	0.0%
• dynamic function resolution	0.0%
• employs evasion techniques	0.0%
• enumerates/manipulates processes	0.0%
• function pointer/table modification	0.0%
• may perform code injection	0.0%
• modification of control flow	0.0%
• obfuscation of code logic	0.0%
• performs complex memory operations	0.0%
• reflectively loads modules	0.0%
• resolves functions at runtime	0.0%
• uses indirect function calls	0.0%

Malicious XZ 5.6.0

KARAMBIT.AI

View XZ attack results Online: https://karambit.ai/app/diff/diff-liblzma-5.4.5-5.6.0

Industry Today

Software Bill of Materials (SBOMs)

Shows list ingredients of software, nothing else

Vulnerability Detection

Does not detect malicious behaviors

Manual Reverse Engineering

Expensive, and labor-intensive

Impact

Eliminate Remediation Costs

Find hidden malicious and safety critical behaviors before deployment

Less time cost for remediation

Speed

Quickly identify known good versions

Deep insight into unique binaries

Software Zero Trust

Verify trusted software

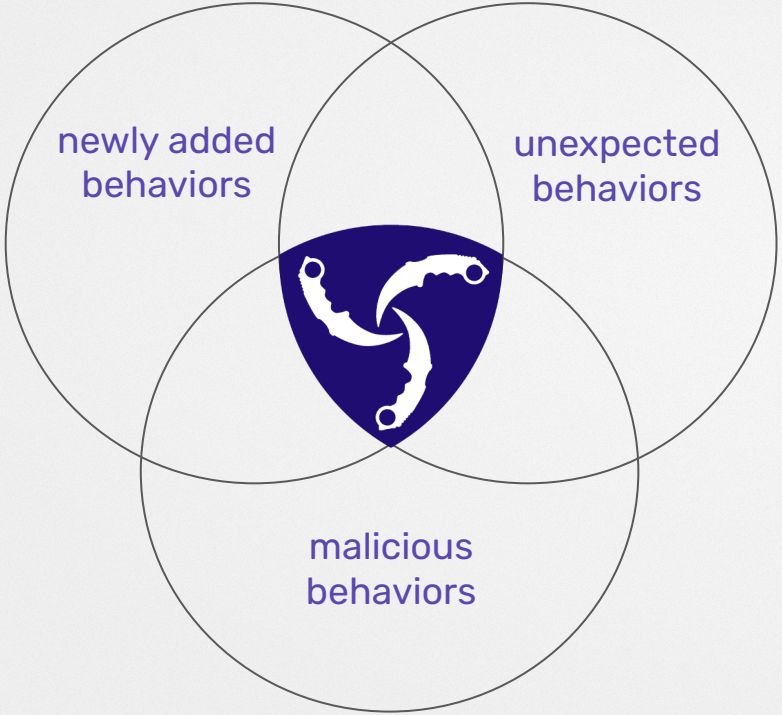
Intimately understand what 3rd party software is capable of

KARAMBIT.AI

Use Case:
Scalable cyber behavior analysis, 1st and 3rd party software

Work with Microsoft TSS Team

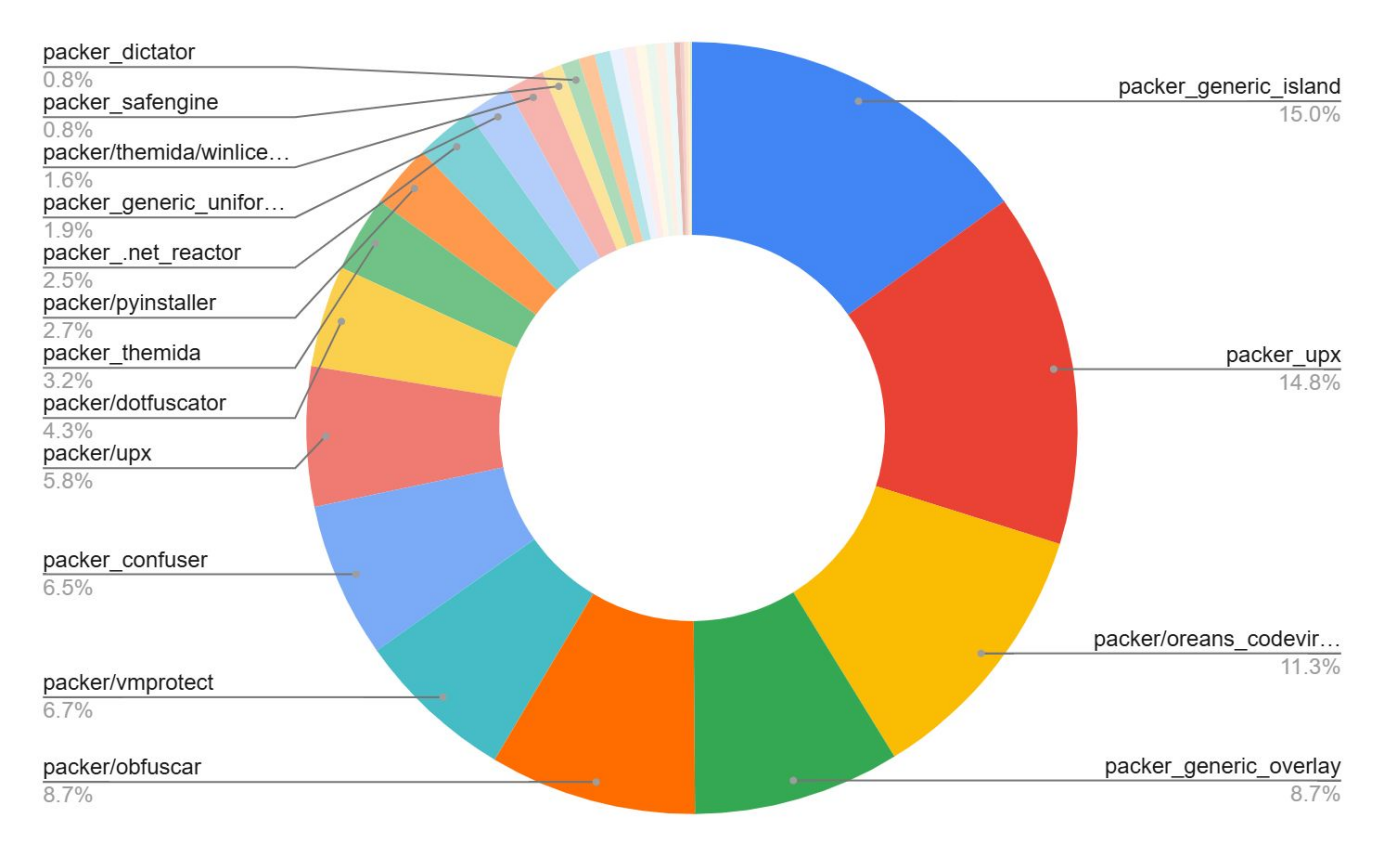
Find hidden malicious behaviors at scale



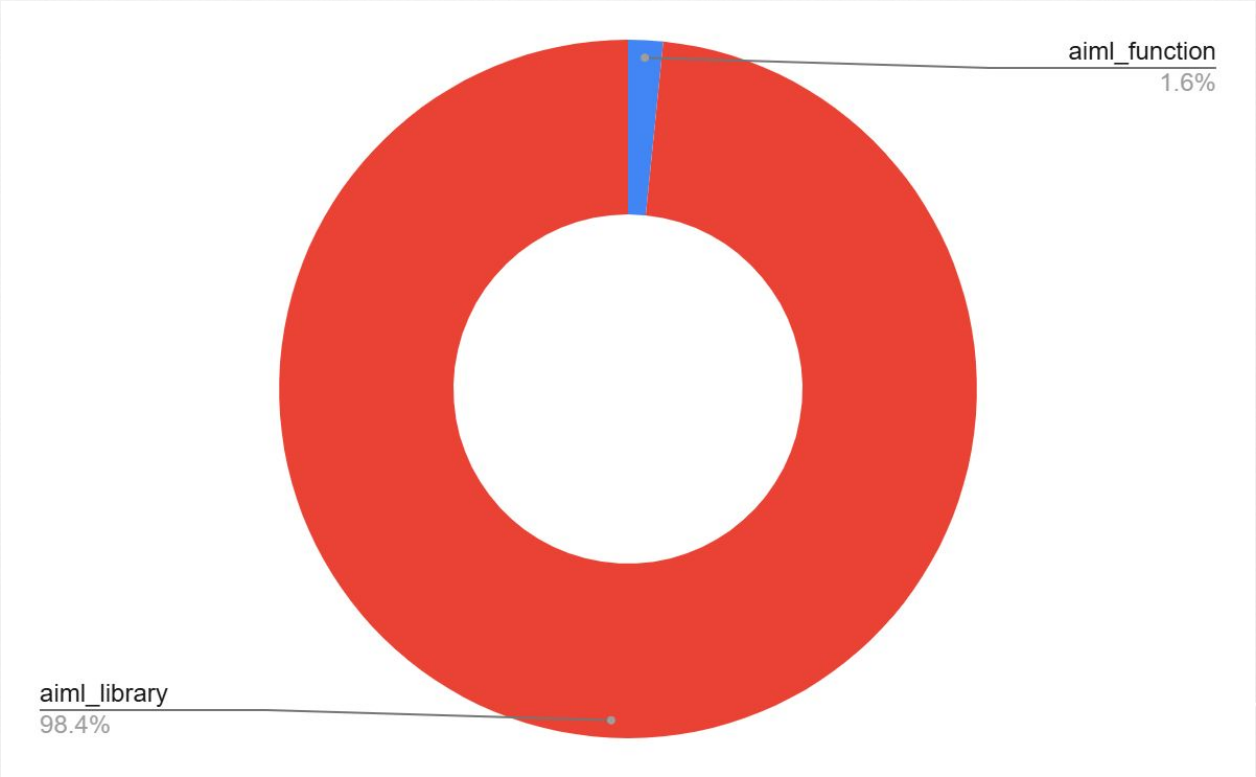
We scan **4B files/month** to discover **hidden malicious behaviors** in software **before deployment,** keeping customers from being compromised by enforcing disallowed behavior policies

KARAMBIT.AI

Packer and Obfuscation Detection



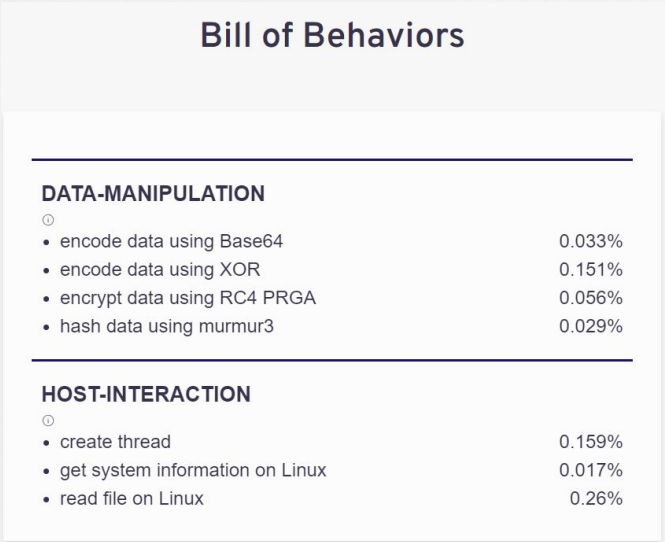
AI/ML Detection



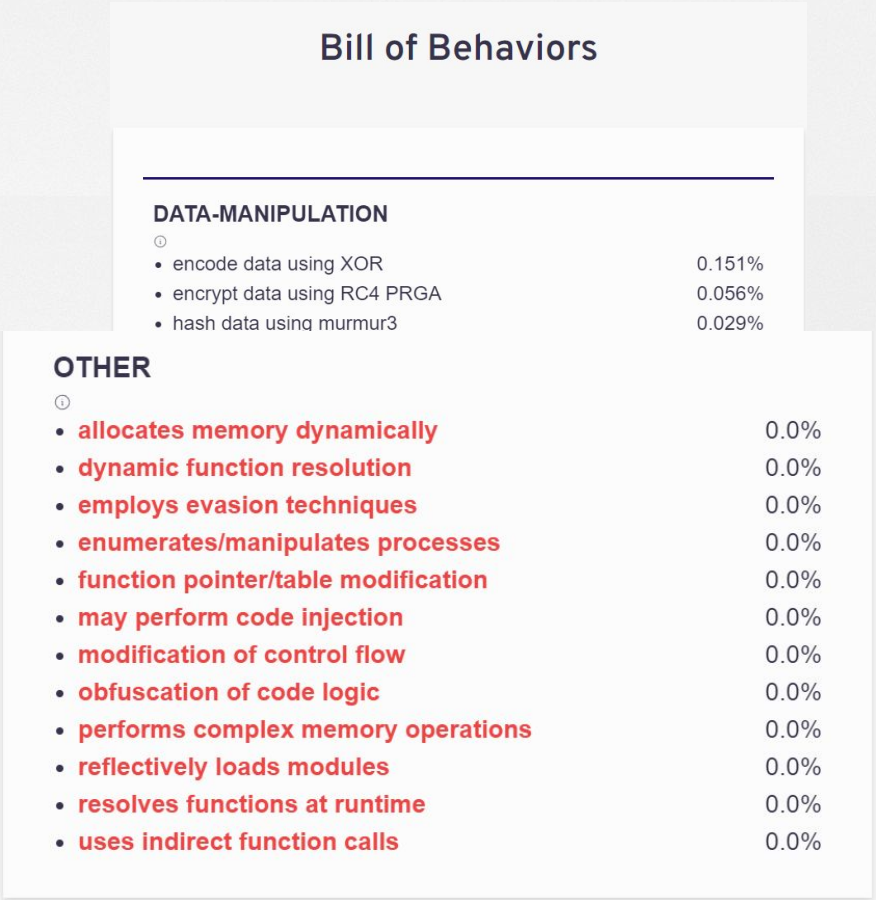
KARAMBIT.AI

Malicious Open Source: XZ

Karambit.AI automatically detects the malicious behaviors added by adversaries.



Benign XZ 5.4.5



Malicious XZ 5.6.0

KARAMBIT.AI

Some libraries have risky behaviors

ANTI-ANALYSIS
- check for software breakpoints

C2
- **download and write a file**
- **execute shell command received from socket on Linux**
- **receive and write data from server to client**
- write and execute a file

COMMUNICATION
- **act as TCP client**
- **connect TCP socket**
- **create TCP socket**
- **create UDP socket**
- receive data
- receive data on socket
- send data on socket

DATA-MANIPULATION
- authenticate HMAC
- encode data using XOR
- encrypt data using RC4 KSA
- encrypt data using RC4 PRGA
- **hash data using CRC32b**
- hash data using fnv
- hash data using murmur3

HOST-INTERACTION
- create thread
- delete file
- execute command
- get local IPv4 addresses
- **get memory information**
- get system information on Linux
- **lock file**
- move file
- read file on Linux
- resolve DNS
- write file on Linux

268CC0

download and write a file
execute command
execute shell command received from socket on Linux
get memory information
read file on Linux
receive and write data from server to client
receive data
receive data on socket
write and execute a file
write file on Linux

‘galera-4,’ which is described as a “Replication framework for transactional applications”

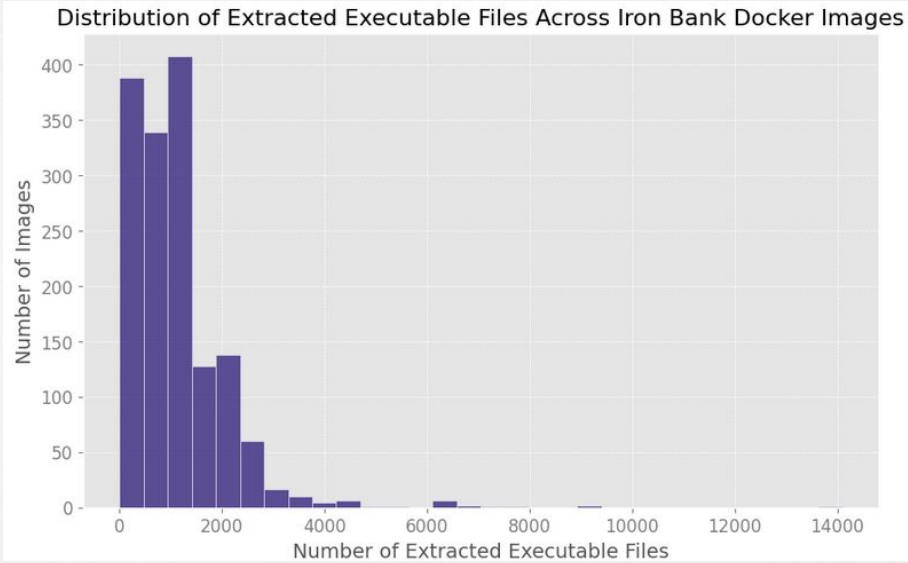
KARAMBIT.AI

Use Case: Docker Container Behavior Analysis

Karambit.AI Processed all AF Iron Bank Docker Containers

- Discovered a series of anti-analysis, C2, and other behaviors
  - Context is key: ssh has C2 behaviors

Total Images Processed with 'latest' tag, March 26, 2024	**1,512**
Total Extracted ELF Files	**1,697,337**
Total Unique ELF Files	**174,219**
Percentage of Total ELF Files that are Distinct	**10.26%**
Total Images with Unique ELF Files	**1,108**



Can detect same component versions, different behaviors

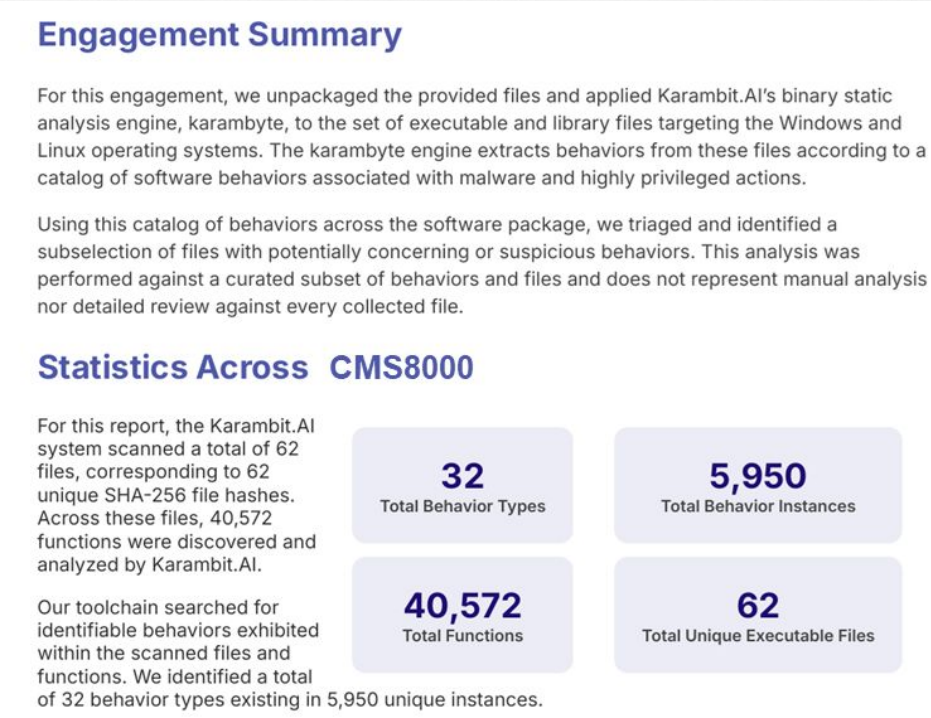
- Detect if one file with the same version has different behaviors
  - Malicious behaviors or benign different compiler options
- Provides the ability to have a “gold” version of a library or component and validate that behaviors match

File with versions with multiple hashes	1,543
Comparisons with no change	6757
Minor suspected compiler-based changes	752
Multiple behavior changes	1593
Max behavior changes	11

Use Case:
Software & Product Assurance

Product *Behavior* Assurance

- Automatically analyze products to be deployed in high assurance environments
  - DoD, Civilian Gov, Medical Devices, Aerospace, etc.
- Provide behavior validation and details to Executive, Product Security and development teams
  - Verify behaviors that are expected
  - Fix behaviors that are unintended
- Human and machine readable output
  - Long form PDF
  - JSON for integration
  - API available



Engagement Summary

For this engagement, we unpackaged the provided files and applied Karambit.AI's binary static analysis engine, karambyte, to the set of executable and library files targeting the Windows and Linux operating systems. The karambyte engine extracts behaviors from these files according to a catalog of software behaviors associated with malware and highly privileged actions.

Using this catalog of behaviors across the software package, we triaged and identified a subselection of files with potentially concerning or suspicious behaviors. This analysis was performed against a curated subset of behaviors and files and does not represent manual analysis nor detailed review against every collected file.

Statistics Across CMS8000

For this report, the Karambit.AI system scanned a total of 62 files, corresponding to 62 unique SHA-256 file hashes. Across these files, 40,572 functions were discovered and analyzed by Karambit.AI.

Our toolchain searched for identifiable behaviors exhibited within the scanned files and functions. We identified a total of 32 behavior types existing in 5,950 unique instances.

32 Total Behavior Types	**5,950** Total Behavior Instances
40,572 Total Functions	**62** Total Unique Executable Files

KARAMBIT.AI

Automatically detecting malicious backdoor in software

Bill of Behaviors

ANTI-ANALYSIS
- check for software breakpoints

C2
- write and execute a file

COMMUNICATION
- create TCP socket
- create UDP socket
- receive data
- receive data on socket
- send data on socket

Summary of Function Operation

The function `FUN_001a6244` is responsible for managing a series of operations, likely involved in connecting to a network, mounting a filesystem, and updating some local files and directories. Here's a step-by-step breakdown of its operations:

1. **Initialization:** The function initializes some variables and checks if `param_3` is zero, indicating it should proceed with the operations.
2. **Network Setup:**
   - It configures network interfaces using the `ifconfig eth0 up` command.
   - It attempts to mount a Network File System (NFS) from `202.114.4.119:/pm` to `/mnt`.
3. **File and Directory Operations:**
   - If the mount is successful, it checks for the existence of `/mnt/monitor`.
   - Copies files from the mounted directory to `/opt/bin` and perform a series of file checks and operations, such as copying configuration files, managing directories related to `Wireless` and ppp, and handling a specific file `key.wav`.

Automatically detect and summarize unapproved behaviors in a patient monitor

Informal Safety Severity Score: High

Score: High

The function executes shell commands that modify system files and directories and initiates network operations which could enable unauthorized access and control. The invocation of commands like ifconfig, mount, and cp suggests potential exploitation paths that are often leveraged for malicious backdoor installation or unauthorized changes. These operations lack visible checks and balance mechanisms to authenticate or validate user permissions, raising concerns about possible misuse or deliberate malicious intent.

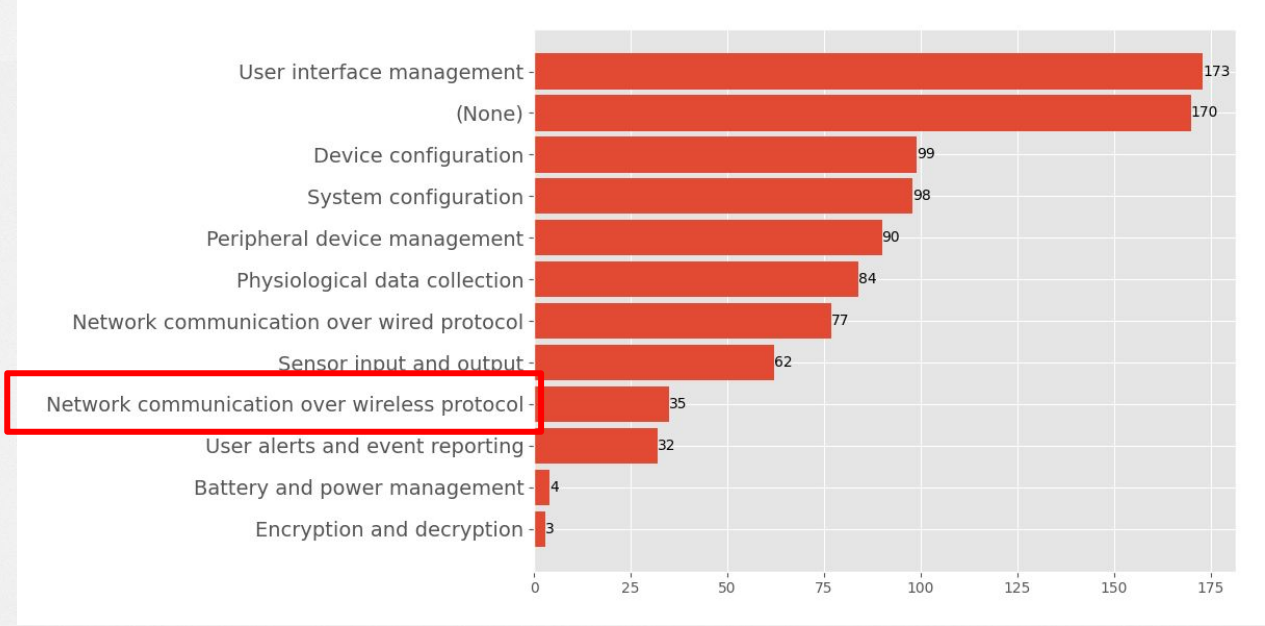
KARAMBIT.AI

Can automatically detect and summarize in human readable language the unapproved behaviors

https://www.fda.gov/medical-devices/safety-communications/cybersecurity-vulnerabilities-certain-patient-monitors-contec-and-epsimed-fda-safety-communication

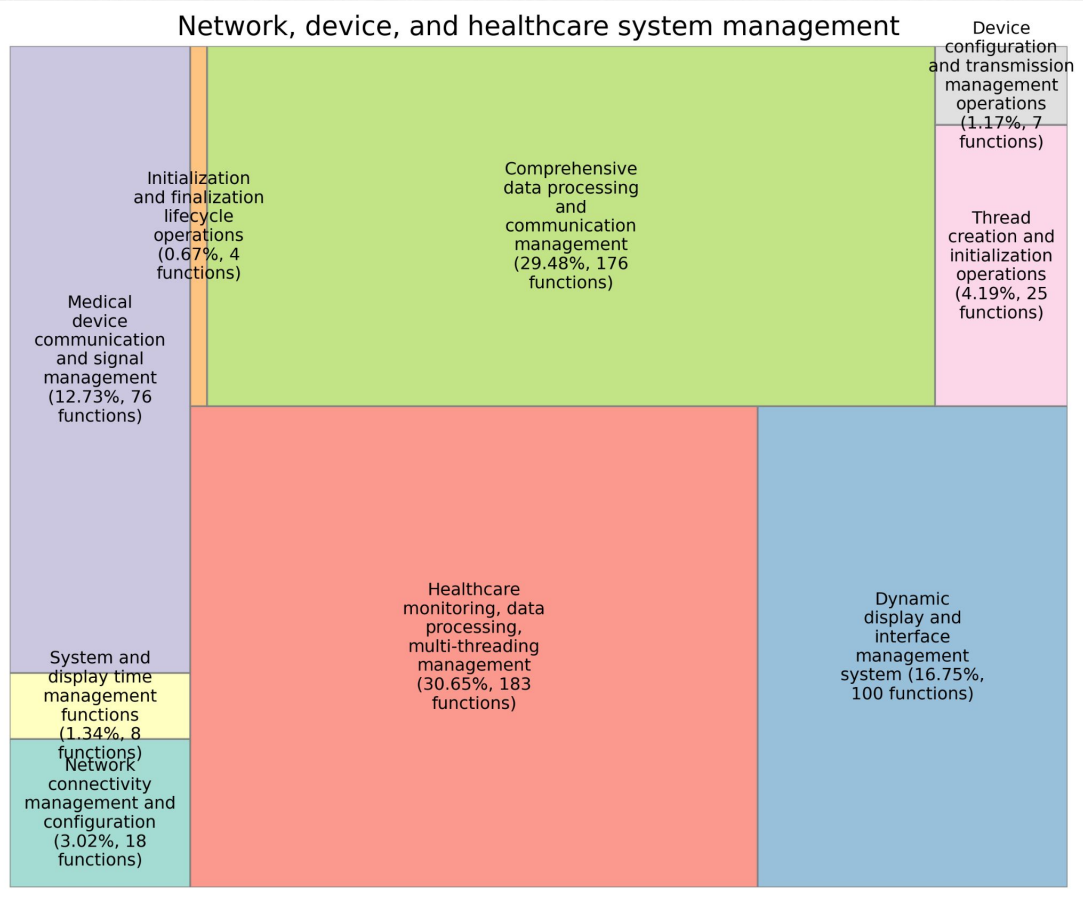
Use Case: Automated Software Behavior Understanding and Description

Categorizing software by function and behavior



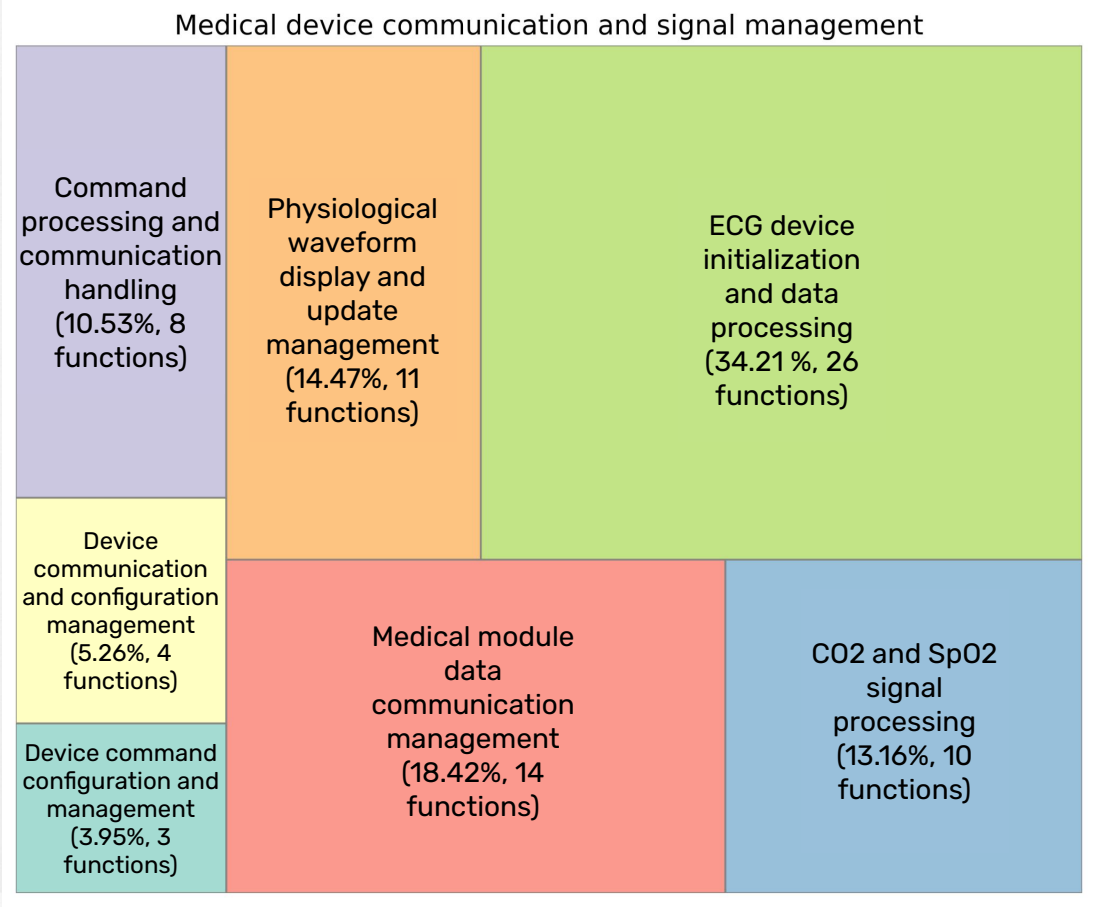
CMS8000 FDA 510(k) did not approved wireless monitoring

Software Functionality Characterization



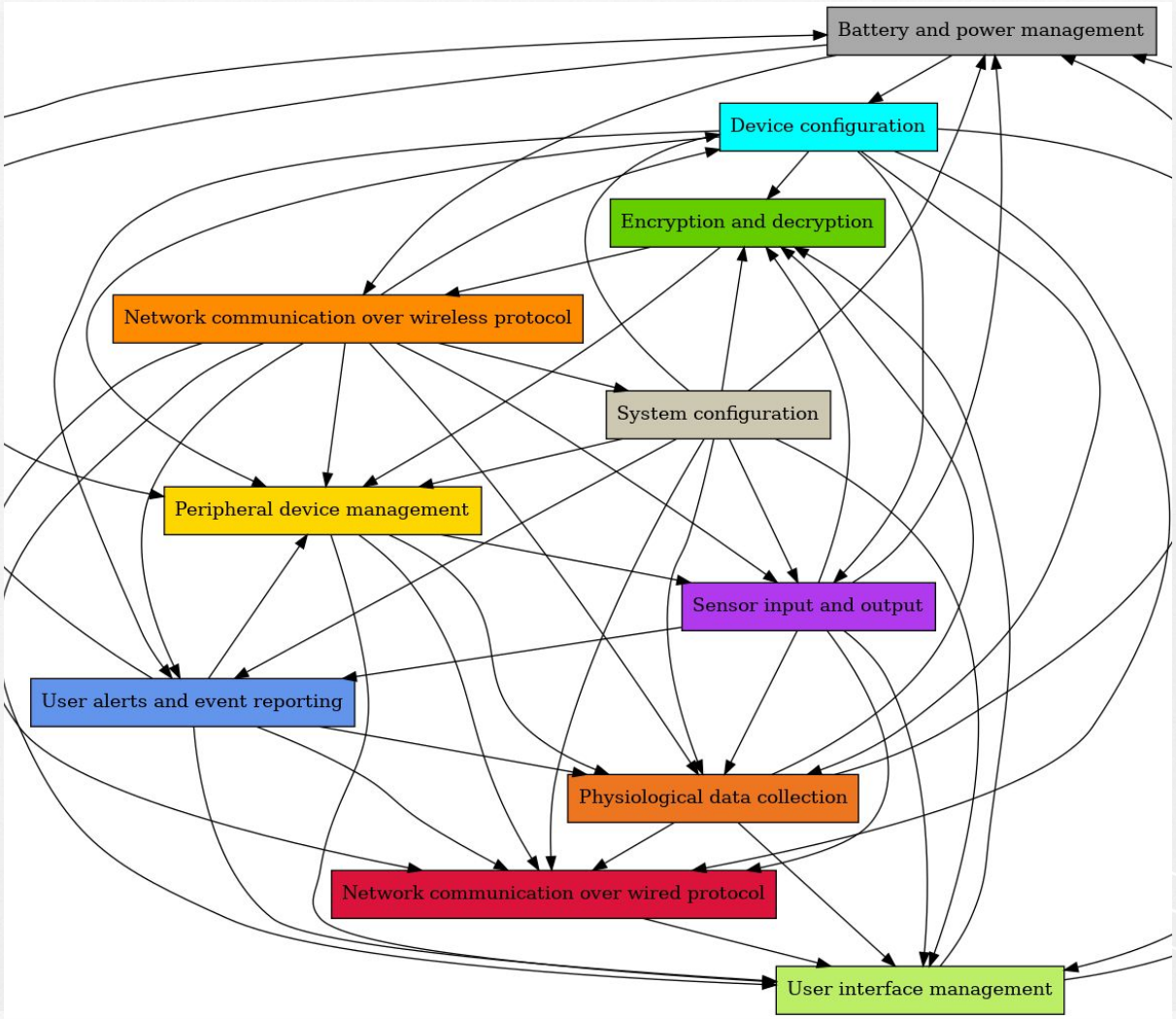
KARAMBIT.AI

Medical Functionality Characterization



KARAMBIT.AI

Architecture

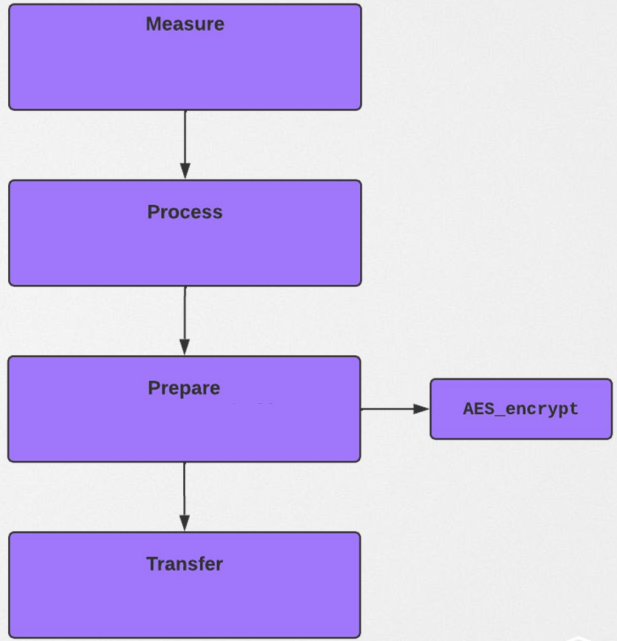


Automated Aggregated Analysis of Essential Functionality

- Natural language descriptions directly from software
- Enables streamlined processes and decision support

The application's data flow begins with the establishment and management of network connections that facilitate communication between healthcare devices and external systems. The core of the application focuses on the processing and management of medical device data. Overall, the application orchestrates a complex flow of data from capture to processing, culminating in display and communication, ensuring the accurate and efficient management of patient data and system resources in a healthcare environment.

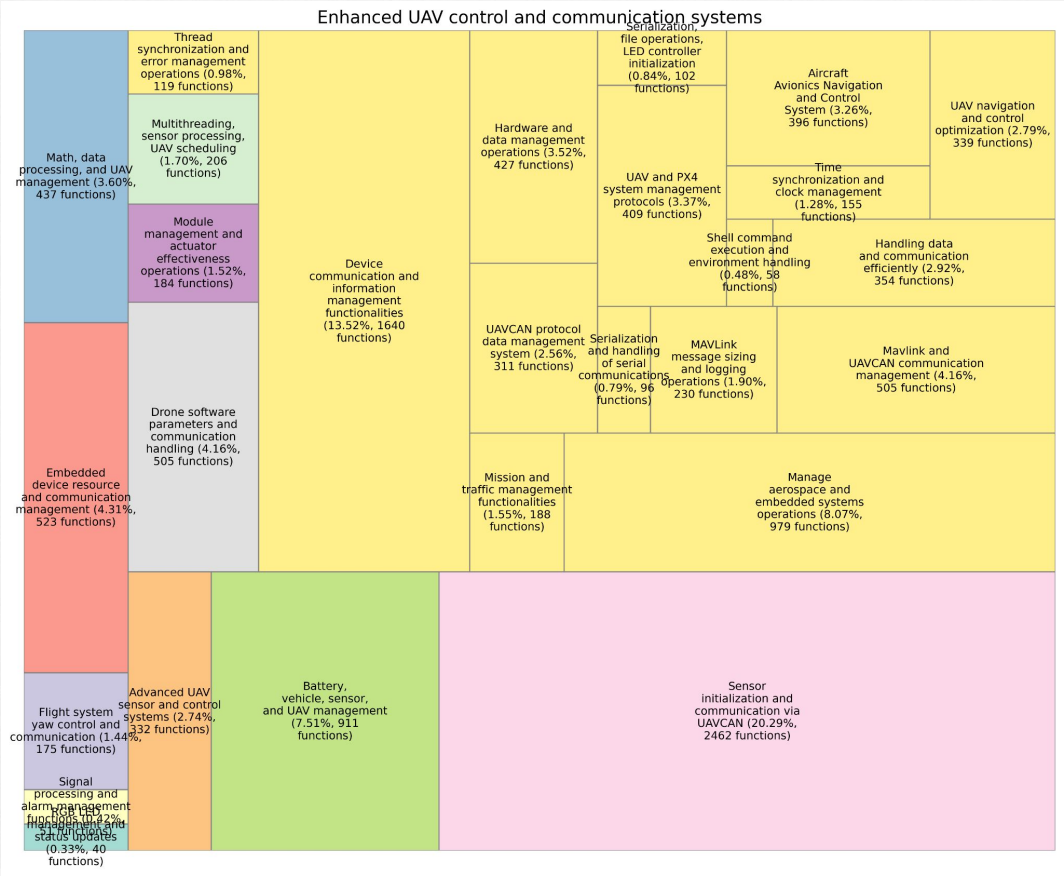
Automated natural language description of data gathering and network communication



Summary graph of software functionality

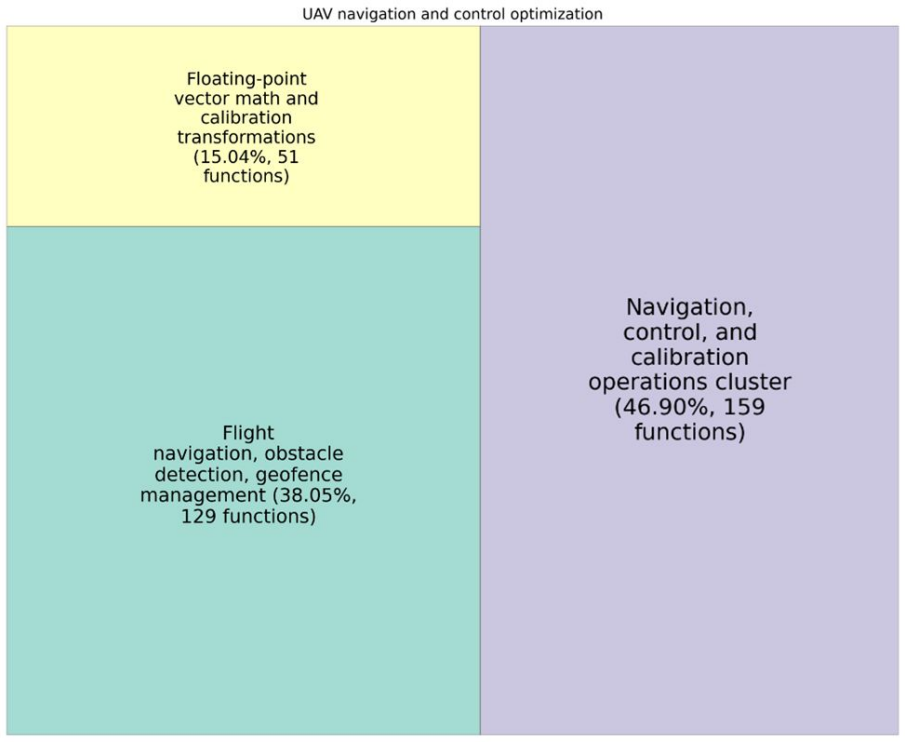
Analysis of Drone Software: Automated Software Behavior Understanding and Description

UAV Control Software Functionality Characterization

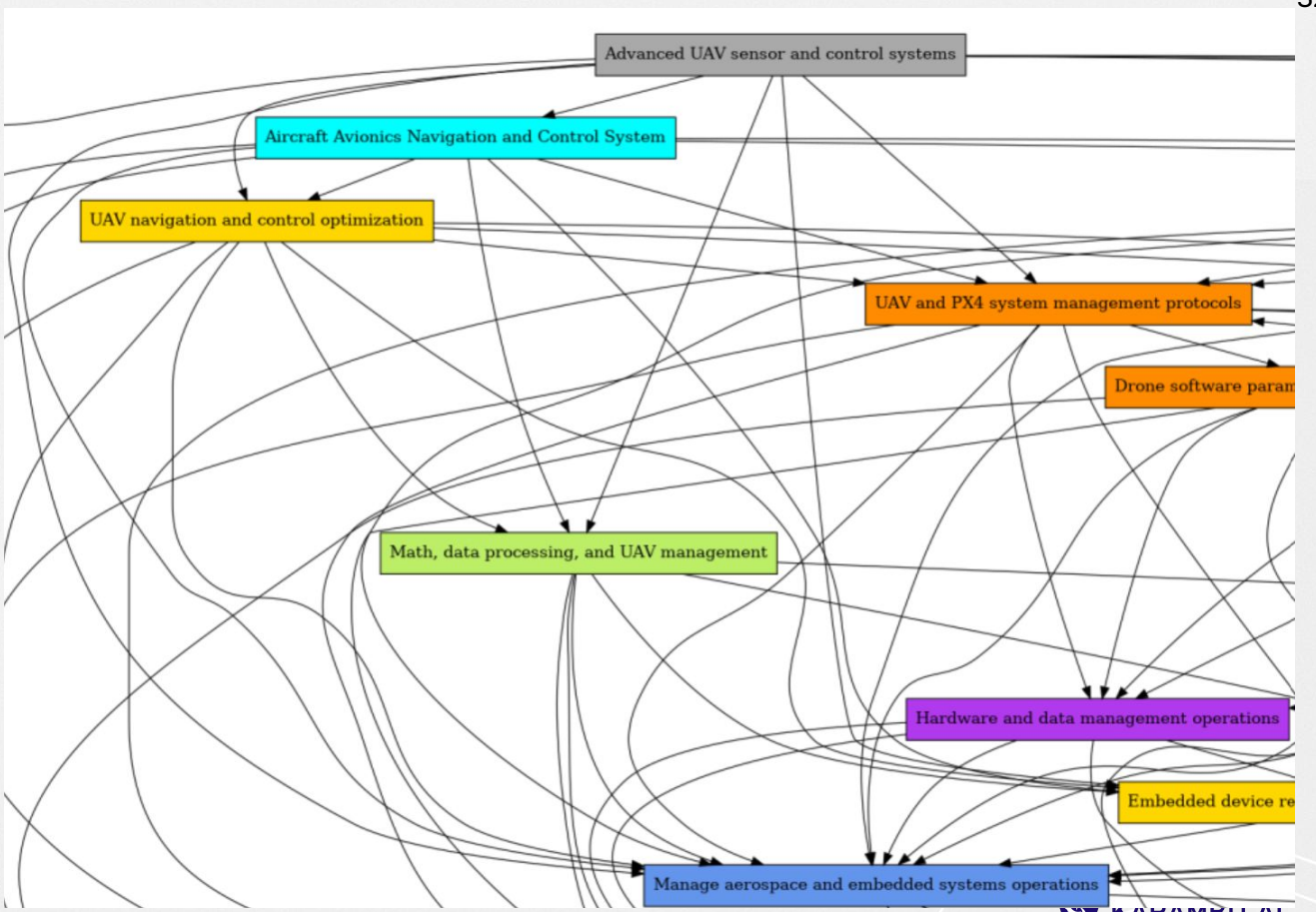


KARAMBIT.AI

UAV Navigation and Control Functionality Characterization



Architecture



Use Case
Risk analysis of Behavior and Function Changes In Software Updates

Enables Assessment of Software Update Risk

- Show behavior changes
  - Safety/Essential behaviors
  - Risky or malware-like behaviors
- Tracks what pieces of the software have changes
  - Uses information on essential functionality

Statistic	Value
Percentage of updated functions	10.5%
Percentage with major changes	13.8%
Percentage with minor changes	86.2%

Statistics for an exemplar software update

Detects the software locations where the small percentage of changes occur while characterizing and explaining the changes

Both `Old_Function` and `New_Function` check the connection status of the BLE driver using the same logic and functionality. **Since the functional logic and the way the status is determined remain unchanged, there is no impact on the operation, safety, or efficacy of the code**.

Example characterization of a BLE function update

The Karambit.AI Founders

Andrew Hendela
Co-founder

- Serial Entrepreneur
- Over a decade of cybersecurity leadership and business development

We have been working together for a decade automating hard cybersecurity problems

cyber attribution,
exploit development,
malware analysis,
vulnerability research

Eric Lee
Co-founder

- Scalable program analysis R&D
- Offensive cybersecurity
- Developed autonomous bug-hunting system for the world's first all-machine cyber hacking tournament

Thank You
Signup for a free account:
https://app.karambit.ai/signup

Andrew Hendela

andrew@karambit.ai

https://karambit.ai

www.linkedin.com/in/andrew-hendela