2020
한국정보보호학회 하계학술대회

CISC-S'20

Conference on Information Security and Cryptography - Summer 2020

07.15 WED
온라인 컨퍼런스
http://www.cisc.or.kr/

주최 주관 한국정보보호학회 Korea Institute of Information Security & Cryptology

후원 국가정보원 National Intelligence Service Korea · 과학기술정보통신부 · 행정안전부 · KISA 한국인터넷진흥원 KOREA INTERNET & SECURITY AGENCY · ETRI 한국전자통신연구원 Electronics and Telecommunications Research Institute · NSR 국가보안기술연구소 National Security Research Institute · cen 아이티센 ITcen

시스템 모니터링 기법을 통한 UAV Stealthy-Monitoring 플랫폼 디자인

최기철*, 박기웅†

*세종대학교 시스템보안연구실, † 세종대학교 정보보호학과

Design of UAV Stealthy-Monitoring Platform over the System Monitoring Techniques

Ki-Choel Choi*, Ki-Woong Park†

*SysCore Lab., Sejong University.
† Department of Computer and Information Security, Sejong University

요 약

최근 드론이 신사업으로 각광받으며 각국 정부 및 기업들의 관심이 높아지고 있다. 이에 대한 결과로 관련 보안 취약점 및 사고사례도 증가하고 있다. 따라서 관련 보안 시스템의 연구가 진행되어야 하며 드론의 특성상 시시각각 환경이 변화하므로 실시간에 가까운 시스템 모니터링이 가능해야한다. 본 논문에서는 이를 해결할 수 있는 방법에 대해 알아보았으며 실시간에 가까운 이상 행위 탐지 특성을 만족하기 위해 부하를 최소한으로 줄일 수 있는 시스템 모니터링 기법에 대해 조사해 보았다.

I. 서론

최근 4차 산업혁명과 맞물려 전 세계적으로 드론에 관해 관심이 높아지고 있다. 이에 관한 결과로 드론 사용이 증가하고 있으며 [1] 관련 보안 취약점 연구 및 해킹 사례도 증가하고 있다. 따라서 이를 해결하기 위해 IoT에서 적용된 보안 기술을 드론에 적용해보려고 한다.

하지만 드론의 경우 전원을 상시 공급받을 수 있는 IoT와는 다르게 배터리를 사용하여 동작하기 때문에 사용 가능한 하드웨어의 성능은 IoT와 같거나 떨어지는 수준이다. 따라서 IoT에 적용 가능한 보안 기술임에도 드론에 적용하지 하는 경우가 생길 수 있고

드론 시스템의 특성상 이동 및 임무 수행 동작에 영향을 미치는 보안 기술을 사용할 수 없다.

따라서 실시간으로 제어가 이루어져야 하는 드론의 특성상 이상 행위 모니터링 시 발생할 수 있는 부하를 최소화해 적용 가능한 기법들을 조사하는 것을 목적으로 한다.

앞서 제시된 문제점을 해결하기 위해 본 논문에서는 모니터링 기법을 3가지로 분류하여 조사해 보았다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 조사한 이상 행위 모니터링 기법을 소개한다. 3장에서는 드론 시스템 이상 행위 모니터링 플랫폼 디자인 설계에 관해 설명한다. 이어서 4장에서는 결론과 향후 연구에 관해 기술한다.

II. 시스템 모니터링 기법 조사

IoT에서 사용 가능한 시스템 모니터링 기법들을 조사하고 3가지 기준에 의해 분류해보았다. 분류 결과를 토대로 드론에 적용 가능 여부를

† 교신저자: 박기웅 (세종대학교 정보보호학과 교수)

본 연구는 ETRI부설연구소 위탁과제 (2020-114)의 지원, 과학기술정보통신부의 재원으로 정보통신기획평가원(IITP)의 지원(No.2018-0-00420, No.2019-0-00273) 및 한국연구재단 연구과제(NRF-2020R1A2C4002737)의 지원을 받아 수행된 연구임.

분류	기법	강점	약점
소프트웨어 기반 모니터링	에이전트	광범위한 정보 수집 가능	많은 자원 소모
	가상화 샌드박스	실행된 악성 행위가 시스템에 영향을 주지 않음	많은 자원 소모
하드웨어 지원을 통한 모니터링	하드웨어 성능 카운터	소프트웨어 기반 모니터링보다 시스템 동작에 영향을 주지 않음	프로세서에서 지원하지 않는 하드웨어 이벤트는 사용할 수 없으며 레지스터 수에 따라 동시 모니터링 가능 여부가 결정
하드웨어 기반 모니터링	JTAG 디버깅	시스템 동작에 영향을 주지 않고 보드 모니터링 가능	상태 값 모니터링 시 일시적으로 정지되는 문제
	전력 소모 모니터링	호환성 여부에 상관없이 이상행위 모니터링 가능	전력 소모량이 예외적인 이유로 변동되면 예측이 불가능

표 1. 3가지로 분류한 시스템 모니터링 기법

알아보는 것이 목적이며 요약 결과는 표 1과 같다.

2.1 소프트웨어 기반 모니터링

Agent 방식의 경우 광범위한 정보 수집이 가능하다는 것이 특징이다. 수집된 정보는 Cloud 서버로 전송되어 분석되며 핵심적인 기능들은 서버에서 동작하여 복잡한 분석이 가능하다. 하지만 OS가 올라가지 않은 시스템에서는 사용이 불가능하다는 단점이 있으며 많은 하드웨어 자원을 소모하여 시스템 동작에 영향을 미친다[2].

또한, Sandbox 중 VM 기반의 Sandbox는 [3] 실행되는 악성 행위가 시스템에 영향을 주지 않도록 하는 것이 특징이다. 하지만 이를 구현하기 위해서는 Agent 방식과 마찬가지로 많은 하드웨어 자원이 소모되어 임베디드 시스템에서 사용되기 어렵다.

2.2 하드웨어 지원을 통한 모니터링

HPC(Hardware Performance Counter)는 [4] 하드웨어 관련 활동의 수를 저장하기 위한 특수한 목적의 레지스터로 PMU(Performance Monitor Unit)로 불리기도 한다. 각 프로세서에서 사용 가능한 HPC의 수와 이벤트 종류는 서로 달라 측정 가능한 수에 제한이 있으며 소프트웨어 기반의 프로파일러와 비교하였을 때 부하가 적고 일반적으로 소스 코드를 수정할 필요가 없다는 장점이 있다. 여러 CPU 제조회사들이 지원하지만, 그중에서도 임베디드 장비에서 자주 쓰이는 ARM 계열 CPU에서 이를 지원하는 목록 중 일부는 표 2와 같다.

Processor	사용 가능한 HW Counter 수
ARM Cortex-A5	2
ARM Cortex-R4	3
ARM Cortex-R5	3
ARM Cortex-A7	4
ARM Cortex-A17	4
ARM Cortex-A9	6
ARM Cortex-A15 MP Core	6
ARM Cortex-M55	8

표 2. HPC를 사용 가능한 ARM 프로세서 (일부)

하지만 HPC의 가장 큰 단점은 프로세서마다 지원하는 HPC 수와 Hardware Event 목록이 다르다는 것이다. 따라서 Software 방식의 모니터링 기법보다는 성능상의 Overhead가 작지만, 모니터링 범위가 불규칙하다는 단점이 존재한다. 또한, 표 2에서 확인할 수 있듯 드론과 같은 임베디드 시스템에서 주로 사용되는 Cortex-M계열의 MCU는 현재 기준으로 M55밖에 없다. M55를 제외한 Cortex-M

계열의 MCU는 대신 DWT(Data Watch-Point and Trace)라는 레지스터가 존재하여 비슷하게나마 사용할 수 있으며 이를 활용하여 PMC와 비슷한 결과를 낼 수 있는지에 대해서는 추가적인 조사가 필요하다[5][6][7].

2.3 하드웨어 기반 모니터링

JTAG(Joint Test Action Group) 또는 Boundary Scan이라고도 불리는 그림 1의 기술은 IEEE 1149.1에 정해진 표준으로 직렬통신 방식을 활용하여 CPU의 내부로 데이터를 전송하거나 Pin의 출력 또는 상태를 읽어와 보드를 원하는 대로 제어할 수 있다. 또한, 위의 특성을 활용하여 플래시 메모리에 부트로더와 같은 작은 프로그램을 저장하는 용도로 사용하기도 한다[8]. 이처럼 호스트에서 JTAG을 사용하여 대상 보드의 상태를 얻어오거나 제어할 수 있도록 관련 API를 제공하는 장치를 JTAG 디버거 장치라고 한다. 또한 JTAG 인터페이스를 활용하여

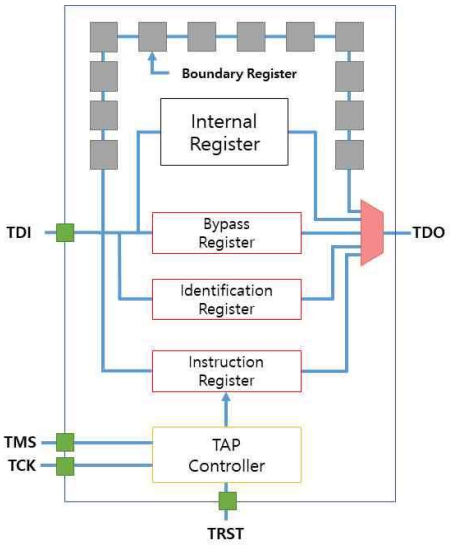

그림 2. JTAG 장치 구조

대상 장치에 대해 프로그램 저장 및 디버깅을 지원하는 오픈소스 디버거 프로그램을 OpenOCD라고 한다[9]. 이러한 디버깅 도구 및 API들을 활용하여 보드에서는 성능적인 문제로 분석할 수 없었지만, PC에서 메모리 분석과 같은 추가적인 분석이 가능하다[10][11][12]. 하지만 JTAG의 경우 상태 값을 읽어오려면 CPU를 일시적으로 정지시켜야 한다. 일시적으로 정지하여도 큰 영향을 미치지 않는 IoT 장비와는 다르게 실시간으로 모터 제어가 이루어져야 하는 드론의 특성상 일시적으로 정지하는 시간이 추락으로 이어질 수 있다. 따라서 이를 해결하는 방법에 대해 추가 조사가 필요하다.

추가로 전력 소모량 분석을 활용한 모니터링 기법이다. IoT처럼 같은 동작을 반복하는 임베디드 시스템에서 소모되는 전력은 일정하고 만약 악성코드 등으로 인해 추가적인 동작 과정이 실행되는 경우 전력 소모량이 변화한다는 가설에 따라 나오게 된 연구 이론이다. 초당 전력 사용량 값을 시계열 기반으로 추출하여 지도학습 기반의 분류 모델을 사용하여 이상 행위 여부를 탐지한다[13][14].

III. 드론 시스템 모니터링 플랫폼 디자인

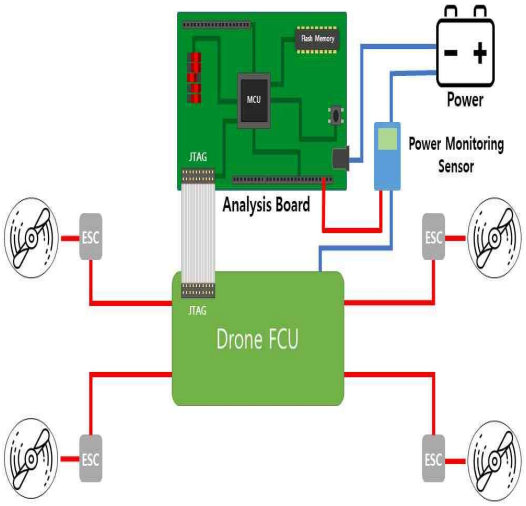

그림 3. 드론 시스템 모니터링 플랫폼

앞서 시스템 모니터링 기법 조사를 통해 드론에 적용 가능한 기법들에 대해 추려보았다. 제안하는 드론 시스템 모니터링 플랫폼 시스템 디자인은 그림 2와 같으며 드론 FCU에 접근하여 분석하기 위한 별도의 보드이다. 추가적인 실험 및 연구를 통해 이상 행위 탐지를 위해 사용될 수 있는 기법을 적용하는 것을 목표로 한다.

IV. 결론 및 추후연구

앞서 조사를 통해 소개한 기법들과 드론 시스템 모니터링 플랫폼 디자인에서 제시한 부분들은 이미 나와 있는 기술 중에 드론에 적용할 수 있다고 판단되어 분류 및 적용한 기술들이다. 하지만 드론에 최적화된 시스템 모니터링 기법 개발을 위해서는 별도의 실험 및 연구를 통해 어떤 부분이 실제 적용이 가능한지와 이에 따른 문제점은 무엇인지 알아보아야 한다.

[참고문헌]

[1] “Goldmansachs Technology Driving Innovation - drone,” [Online]: https://www.goldmansachs.com/insights/technology-driving-innovation/drones/

[2] Ma Xinglu, Qu Yingjie, Research on Embedded Agent System Architecture, September, 2008

[3] Kai-Chi Chang, Raylin Tso, Min-Chun Tsai, IoT Sandbox - To Analysis IoT malware Zollard, March, 2017

[4] “Hardware performance counters,” [Online]: http://en.wikipedia.org/wiki/Hardware performance counter.

[5] “ARM Infocenter,” [Online]: http://infocenter.arm.com/

[6] Hojoon Lee, Hyungon Moon, Daehee Jang, Kihwan Kim, Jihoon Lee, Yunheung Paek, Brent ByungHoon Kang, KI-Mon: A Hardware-assisted Event-triggered Monitoring Platform for Mutable Kernel Object, August, 2013

[7] Xueyang Wang, Charalambos Konstantinou, Michail Maniatakos, Ramesh Karri, ConFirm: Detecting firmware modifications in embedded systems using Hardware Performance Counters, November, 2015

[8] “JTAG,” [Online]: https://ko.wikipedia.org/wiki/JTAG

[9] “OpenOCD,” [Online]: http://openocd.org/about/

[10] Gildo Torres, Zhiliu Yang, Zander Blasingame, James Bruska, Chen Liu, Detecting Non-Control-Flow Hijacking Attacks Using Contextual Execution Information, June, 2019

[11] Kim Hyung Chan, Park Il Hwan, An Implementation of JTAG API to Perform Dynamic Program Analysis for Embedded Systems, February, 2014

[12] Charalambos Konstantinou, Eduardo Chielle, Michail Maniatakos, PHYLAX: Snapshot-based Profiling of Real-Time Embedded Devices via JTAG Interface, March, 2018

[13] Robert A. Bridges, Jarilyn M. Hernandez Jimenez, Jeffrey Nichols, Katerina Goseva-Popstojanova, Stacy Prowell, Towards Malware Detection via CPU Power Consumption: Data Collection Design and Analytics, August, 2018

[14] Shane S. Clark, Benjamin Ransford, Amir Rahmati, Shane Guineau, Jacob Sorber, Kevin Fu, Wenyuan Xu, WattsUpDoc: Power Side Channels to Nonintrusively Discover Untargeted Malware on Embedded Medical Devices, August, 2013