Poster: Automated Vulnerability Assessment for Embedded Systems

Ulrich Lang, Holmes Chuang, Reza Fatahi, Will Swift, Jason Kramer
ObjectSecurity LLC
{ ulrich | holmes | reza | will | jason }@objectsecurity.com

***Abstract* - We present the results of our Navy-sponsored research effort to develop an automated vulnerability assessment tool that meets stringent requirements for already-deployed embedded systems, incl. can be portable, offline, battery-powered, and usable**

I. INTRODUCTION

Cybersecurity for IIoT and especially embedded systems has many challenges. For example:

- Outdated/vulnerable software, un-patchable
- Lack of risk assessment/management knowledge for IIOT
- Cybersecurity tools often do not directly apply (e.g., antivirus)
- Hardware physically accessible/unprotected
- Weak encryption and other self-defenses due to limited resources
- No/limited monitoring due to bandwidth/ connectivity constraints

Particular challenges with embedded devices include:

- Often already deployed
- Often cannot physically be removed
- Usually no screen/keyboard
- Logins often unknown (manufacturer/servicer only)
- Device internals often unknown, legacy systems, no documentation

Embedded systems today are assessed by: hiring vulnerability assessors / pen-testers who will try to connect to interfaces, break into the device, extract the firmware, analyze it for vulns., create a report

II. OBJECTIVES

Funded as part a Navy (ONR) SBIR Phase I/II “Red Team in a Box for Embedded and Non-IP Devices” (Navy SBIR 2018.2 - Topic N182-131 [1]), ObjectSecurity [2] was tasked to develop an approach and prototype (“RedBox”) [3] that can overcome the limitation of human red team resources for conducting vulnerability assessments on Navy systems, in particular, cyber-physical systems. In other words, how can this be done simpler/cheaper at scale?

As part of this research, we are developing an automated vulnerability assessor / pen tester:

- Portable device that non-experts can connect to IIOT devices deployed in the field
- Extracts & analyzes firmware
- Provides easy-to-understand result
- Uses AI (deep learning) to predict what works best

We previously also completed SBIR research to develop an automated red team hacker for cyber training – and an objective was to reuse some of the results from that project too.

III. MATERIALS & METHODS

We are currently ~1.5 years into a ~3.5ßyear period of performance. We have developed a working end-to-end prototype. The actual portable device automatically executes sequences of actions on devices to identify ports (console, JTAG, UART etc.), break into a command shell, extract binaries (firmware), and run vulnerability assessments on the extracted software.

IV. SYSTEM OVERVIEW

The system runs through the following main phases:

<u>1) Connect</u> to external connectors (D-Sub, USB, serial, SD card), and internal UART/JTAG (Universal Async. Receiver/Transmitter, Joint Test Action Group) on the circuit board.

Figure 1 - External connection via RJ45

<u>2) Extract:</u> automatically gain access to the system (using basic automated pen-testing), ideally via a command shell. It then automatically extracts the firmware from the device.

Figure 2 - Connection instructions on UI

<u>3) Analyze</u>: automatically analyzes the extracted firmware for known and zero-day vulnerabilities, including binary vulnerabilities assessments, decompiling or disassembling and analyzing the decompiled source. Results are aggregated, filtered, mapped to a standard, and prioritized by potential impact.

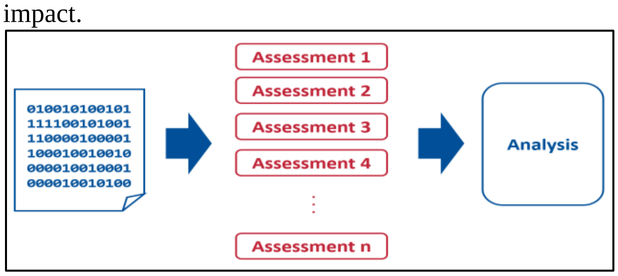


Figure 3 - Assessment pipeline

<u>4) Report:</u> simple user output on the device for non-experts (e.g. traffic light), and details are stored for further aggregation and analysis (and uploaded to a backend when RedBox has internet connection)..The left side shows a traffic-light score for non-experts, while the right side shows an ELK stack based backend with advanced visual analytics capabilities for experts:

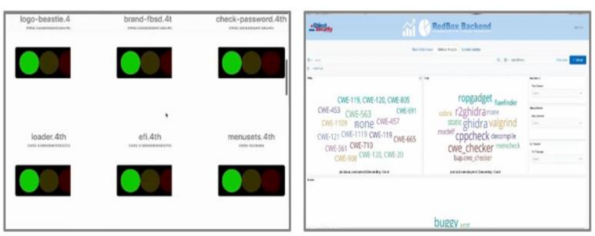


Figure 4 – Results (left: summary, right: details)

<u>5) Adapt:</u> uses artificial intelligence (AI) to learn and adapt from every device analysis. The following figure shows different thicknesses based on how reinforced each assessment sequence step and path are:

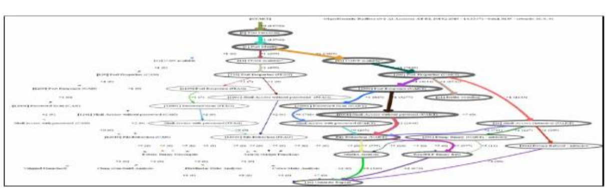


Figure 5 - Deep reinforcement learning sequencing visualization

V. RESULTS

While the project is still ongoing, there are already some preliminary results:

- First working prototype meets most of the requirements
- Requirements (portable, offline, battery-powered, usable by non-experts, for previously unknown devices) are maybe too stringent. Connect/extract requires expertise unless console port etc. externally available
- Deep reinforcement learning for sequencing not as beneficial as expected due to nature of sequencing
- Many open source and academic vulnerability assessment tools are not production-ready. Commercial tools are often very pricy (e.g. IDA Pro).
- Conventional dynamic assessment too resource-intensive for our use case. Machine learning for novel vulnerability assessments seems a good route forward.
- Market research results indicate commercial interest

VI. CONCLUSIONS

While it is immature to jump to conclusions before the project is completed, there are already some preliminary conclusions:

- Interviews with potential users indicate that there are use cases for this technology that can offer fast/cheap/automated prioritization of binaries for human testers to look at in more detail.
- Automating unwieldy vulnerability assessment tools designed for experts is at times challenging and engineering-intensive
- Identifying a code weakness does not always correlate to a vulnerability

ACKNOWLEDGEMENT

Parts of this work is sponsored by the U.S. Navy under contract N6833520C0094.

REFERENCES

[1] SBIR topic description: https://www.navysbir.com/n18_2/N182-131.htm (retrieved 2/16/2021)

[2] ObjectSecurity publications: objectsecurity.com/publist (retrieved 2/16/2021)

[3] RedBox website: objectsecurity.com/vaptbox (retrieved 2/16/2021)

Poster:
AUTOMATED VULNERABILITY ASSESSMENT FOR EMBEDDED SYSTEMS

Ulrich Lang, Holmes Chuang, Reza Fatahi, Will Swift – ObjectSecurity LLC

INTRODUCTION

Cybersecurity for IIoT and especially embedded systems has many challenges, e.g.:

- Outdated/vulnerable software, un-patchable
- Lack of risk assessment/management knowledge for IIOT
- Cybersecurity tools often do not directly apply (e.g. antivirus)
- Hardware physically accessible/unprotected
- Weak encryption and other self-defenses due to limited resources
- No/limited monitoring due to bandwidth/connectivity constraints

Particular challenges with embedded devices include:

- Often already deployed and often cannot physically be removed
- Usually no screen/keyboard
- Logins often unknown (manufacturer/servicer only)
- Device internals often unknown, legacy systems, no documentation

Embedded systems today are assessed by: hiring vulnerability assessors / pen-testers who will try to connect to interfaces, break into the device, extract the firmware, analyze it for vulns., create a report

OBJECTIVES

Funded as part a Navy (ONR) SBIR Phase I/II "Red Team in a Box for Embedded and Non-IP Devices" (Navy SBIR 2018.2 - Topic N182-131) we were tasked to develop an approach and prototype ("RedBox") that can overcome the limitation of human red team resources for conducting vulnerability assessments on Navy systems, in particular, cyber-physical systems. In other words, how can this be done simpler/cheaper at scale?

As part of this research we are developing an automated vulnerability assessor / pen tester:

- Portable device that non-experts can connect to IIOT devices deployed in the field
- Extracts & analyzes firmware
- Provides easy-to-understand result
- Uses AI (deep learning) to predict what works best

We previously also completed SBIR research to develop an automated red team hacker for cyber training – and an objective was to reuse some of the results from that project too.

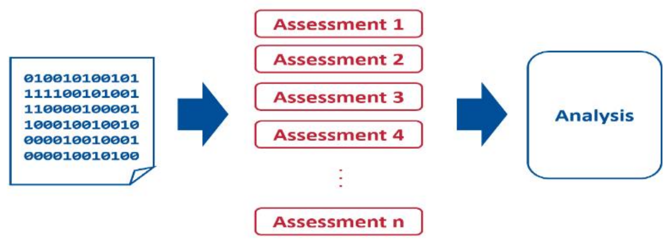
MATERIALS & METHODS

We are currently ~1.5 years into a ~3.5 year period of performance. We have developed a working end-to-end prototype. The actual portable device automatically executes sequences of actions on devices to identify ports (console, JTAG, UART etc.), break into a command shell, extract binaries (firmware), and run vulnerability assessments on the extracted software.

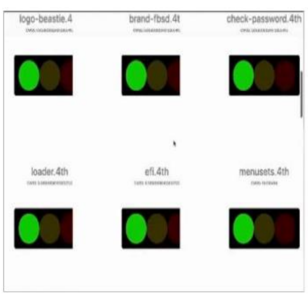
1) Connect to external connectors (D-Sub, USB, serial, SD card), and internal UART/JTAG (Universal Async. Receiver/Transmitter, Joint Test Action Group) on the circuit board

2) Extract: automatically gain access to the system (using basic automated pen-testing), ideally via a command shell. It then automatically extracts the firmware from the device

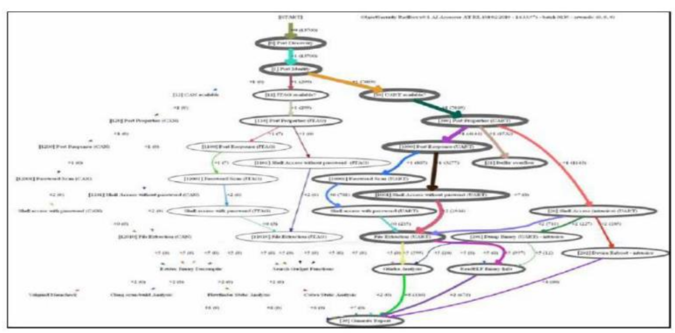
3) Analyze: automatically analyzes the extracted firmware for known and zero-day vulnerabilities, including binary vulnerabilities assessments, decompiling or disassembling and analyzing the decompiled source. Results are aggregated, filtered, mapped to a standard, and prioritized by potential impact



4) Report: simple user output on the device for non-experts (e.g. traffic light), and details are stored for further aggregation and analysis (and uploaded to a backend when RedBox has internet connection).



5) Adapt: uses artificial intelligence (AI) to learn and adapt from every device analysis



RESULTS

- First working prototype meets most of the requirements (validated portable/offline automated firmware extraction, analysis, and scoring with a range of embedded devices is viable)
- Requirements (portable, offline, battery-powered, usable by non-experts, for previously unknown devices) are maybe too stringent. Connect/extract requires expertise unless console port etc. externally available
- Deep reinforcement learning for sequencing not as beneficial as expected due to nature of sequencing
- Many open source and academic vulnerability assessment tools are not production-ready. Commercial tools are often very pricy (e.g. IDA Pro).
- Conventional dynamic assessment too resource-intensive for our use case. Machine learning for novel vulnerability assessments seems a good route forward.
- Market research results indicate commercial interest

CONCLUSIONS

- Interviews with potential users indicate that there are use cases for this technology that can offer fast/cheap/automated prioritization of binaries for human testers to look at in more detail.
- Automating unwieldy vulnerability assessment tools designed for experts is at times challenging and engineering-intensive
- Identifying a code weakness does not always correlate to a vulnerability

REFERENCES

- objectsecurity.com/vaptbox
- objectsecurity.com/publist
- https://www.navysbir.com/n18_2/N182-131.htm

CONTACT

ObjectSecurity LLC
1855 1st Ave #103, San Diego, CA 92101
info@objectsecurity.com, 650-515-3391, @objectsecurity

ACKNOWLEDGEMENTS

Parts of this work is sponsored by the U.S. Navy under contract N6833520C0094.