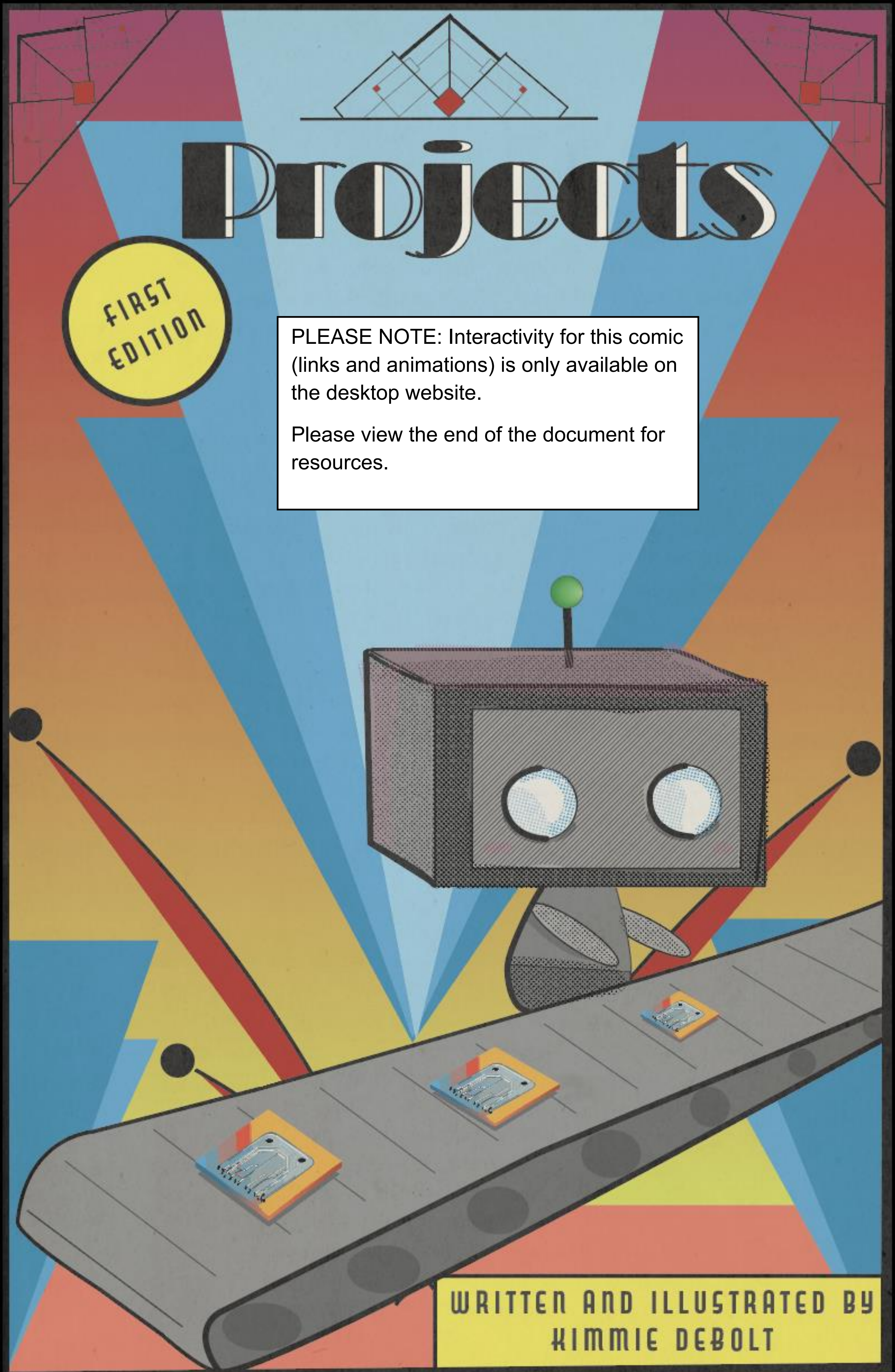
Projects



PLEASE NOTE: Interactivity for this comic (links and animations) is only available on the desktop website.

Please view the end of the document for resources.

WRITTEN AND ILLUSTRATED BY KIMMIE DEBOLT

CONTENT

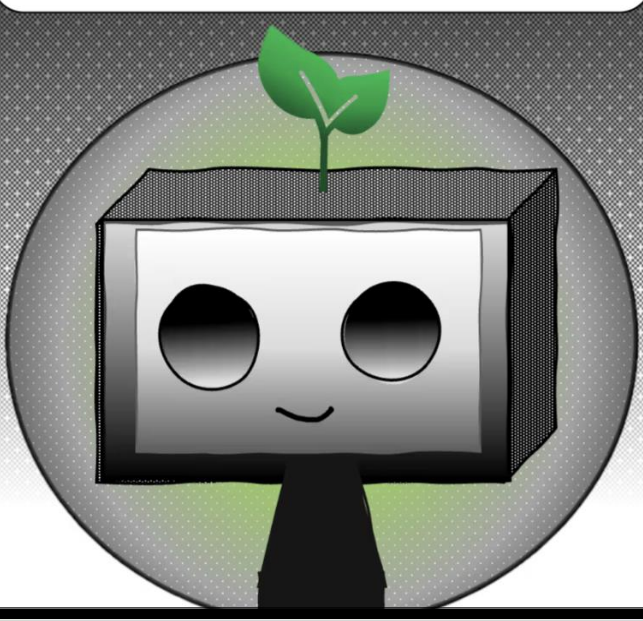
Project 1 Smart Greenhouse with GUI

Project 2 Malware Analysis and Reverse Engineering

Project 3 Technical Writing

Project 4 Penetration Testing

Project 5 Digital Forensics and Log Analysis



Project 1: Smart Greenhouse with GUI

This project was one of my personal favorites and it earned third place at the annual Henry Ford College Engineering Competition. My team and I designed and built a mini smart greenhouse model, aimed at addressing food security. It was complete with a custom GUI and developed utilizing Arduino.

The system featured many components, including: A fan which was integrated and connected to a temperature and humidity sensor. Once the internal temperature exceeded the user set threshold, the fan would activate. An irrigation system that was controlled by a soil moisture sensor was also developed. Each component could be manually operated via the IoT interface, giving users remote control and real-time feedback.

We programmed the system using Arduino-based C++ code, and integrated live notifications and responsive controls. I also gained experience with the hardware assembly and soldering, which allowed me to gain a deeper appreciation for circuit design and board layout.

Greenhouse Conclusion and Resources

This greenhouse prototype will hopefully be used to build larger scale community greenhouses around the world.

It is statistically proven that in economically struggling communities, nutritious foods are harder to obtain.

With anticipation, these smart greenhouses will be scattered across the globe, providing many communities with healthier food options and one less thing to worry about in an uncertain world.

Greenhouse Final Report

List of Materials

Project 2: Malware Analysis and Reverse Engineering

This project was instrumental in deepening my understanding of malware analysis and strengthening my ability to perform the associated tasks effectively. Unlike some of my hardware-based projects, this one focused entirely on software—a common characteristic in the field of cybersecurity.

Malware analysis requires a robust toolkit. I gained hands-on experience with a wide range of tools used to examine malicious executables and dynamic libraries. These included:

File Identification & Analysis Tools: HxD, Exeinfo PE, PEiD, CFF Explorer, pestudio, and IDA Freeware

Debuggers & Disassemblers: WinDbg, IDA Free, x64dbg

Hash Generators: HashMyFiles, HashCalc, HashTab

Threat Intelligence: VirusTotal for identifying known malware samples via hash matching

Malware Analysis & Reverse Engineering Conclusion and Resources

These tools helped me assess critical characteristics such as file type, target architecture, operating system, and PE format (e.g., .exe, .dll). I also became proficient in identifying suspicious strings, analyzing PE headers, unpacking executables, and applying de-obfuscation techniques.

If you're interested in learning more about malware analysis or want to try a hands-on exercise, check out my Notion guide linked below. As always, be sure to work within a secure, isolated environment—such as a virtual machine or sandbox—to avoid risk to your system.

I have also linked an introduction to Malware Analysis by Hackersploit, which is another great place to start.

Malware Analysis Notion Page

HackerSploit's Introduction to Malware Analysis

Project 3: Technical Writing

These projects are heavily focused within technical writing. They helped cement my knowledge in writing informational resources, and including relevant data in a way that is not only easy to understand but productive towards helping the reader gain knowledge.

Throughout this process, I also learned and re-familiarized myself with the fundamentals of computing and CVE resources. It was insightful to get back to the basics after advancing so far in my studies.

The perspectives that I now carry would have proved very valuable to me when I first began my journey; but alas, I hope by curating these resources I can help anyone who's interested.

Technical Writing Conclusion and Resources

These projects also assisted in furthering my understanding of technical writing and it's aspect of perspective, information, and articulateness. What good is the information, if it is not clearly stated in a way that is ultimately digestible by the reader? The practical applications of technical writing are incredibly important in being successful within any career space (especially a... technical one).

Please find a couple of the technical writing resources I have created below!

A Guide for Utilizing CVEs for Threat Management

Inside the Central Processing Unit

Project 4: Penetration Testing

This project simulated a full-scale penetration test using the De-ICE virtual machine. This environment is intentionally vulnerable and designed to mimic real-world vulnerabilities. The goal was to find, exploit, and record vulnerabilities in a simulated network, as if in a professional red-team engagement.

Environment & Setup:

The De-ICE VM ran in VirtualBox, isolated in a controlled testing environment. I used Kali Linux as the attacking machine. It had standard penetration testing tools like Nmap, Nikto, Burp Suite, Hydra, and Metasploit.

Penetration Testing Conclusion and Resources

Following a structured penetration testing approach, I applied elements of the PTES (Penetration Testing Execution Standard) and OWASP Testing Guide.

Reconnaissance:

I utilized Nmap for port scanning and service enumeration
Identified open ports (HTTP, FTP, SSH, etc.) and banners for version fingerprinting
Vulnerability Identification
Ran web vulnerability scans using Nikto and manual analysis in Burp Suite.
Here I discovered misconfigurations and outdated services susceptible to known exploits.

Exploitation:

Performed brute-force attacks with Hydra against SSH and FTP login forms.
Leveraged Metasploit to exploit a vulnerable web service and gain initial access.
Escalated privileges using local enumeration scripts (linPEAS, Linux Exploit Suggester).

Post-Exploitation:

Extracted flags and sensitive configuration files
Demonstrated potential data exfiltration techniques.

Sample Penetration Test Report

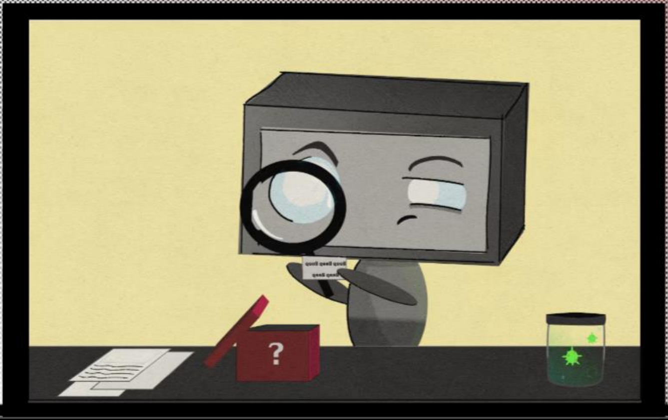
Project 5: Digital Forensics and Log Analysis

This project involved a structured forensic investigation and log analysis exercise to identify evidence of suspicious activity within a compromised system. The objective was to simulate a response scenario, using real-world tools and techniques to trace attacker behavior, extract key evidence, and document findings.

Scope & Environment

The investigation was performed in a controlled lab environment using a Windows 10 virtual machine as the target system. Logs were collected from multiple sources, including:

Windows Event Logs (Security, System, Application)
Firewall logs
Web server logs (Apache)
Network traffic captures via Wireshark



Digital Forensics and Log Analysis Conclusion

Tools Used

Autopsy and FTK Imager for disk and memory forensics
Event Viewer and Log Parser for system log inspection
Wireshark for packet analysis
Sysinternals Suite (especially PsExec and Autoruns)

This project reinforced my understanding of digital forensics fundamentals, enhanced my ability to analyze structured and unstructured logs, and demonstrated how to extract actionable intelligence from raw system data. It also emphasized the importance of maintaining forensic soundness and clear documentation throughout an investigation.

Resources:

Green House Final Report

List of Materials

Malware Analysis Notion Page

HackerSploit's Introduction to Malware Analysis

A Guide for Utilizing Common Vulnerability and Exposures for Threat Management in Cybersecurity

Inside the Central Processing Unit: An Informative Look into the Function of CPUs

Sample Penetration Test Report