3D Bioprinter Firmware Attacks: Categorization, Implementation, and Impacts

Muhammad Ahsan[1], Barry Najarro-Blancas[2], Johanna Tsala Ebode[2], Nastassja Lewinski[2], and Irfan Ahmed[1]

[1]Department of Computer Science, Virginia Commonwealth University
[2]Department of Chemical and Life Science Engineering, Virginia Commonwealth University
Email: {*ahsanm5, najarroblab, tsalaebodej, nalewinski, iahmed3*}*@vcu.edu*

***Abstract*—3D bioprinting has emerged as a transformative tool in bioengineering, offering the capability to fabricate complex geometries with precision and minimal material waste. With its growing adoption in security-critical applications, securing them against adversarial threats becomes imperative. The global supply chain further exacerbates the risk by providing adversaries with practical avenues for hardware malware injection. This study introduces a structured categorization of adversarial objectives achievable through malicious firmware in 3D bioprinting technology. To ensure the persistence of malicious firmware, we propose a four-step attack vector leveraging a trojan (MalBoot) to maintain the compromised firmware on the infected device. We identified 26 potential attacks across the proposed categories and implemented and evaluated ten firmware-assisted sabotage attacks. These include intellectual property theft and targeted sabotage of the printer and the printed construct. The attacks were demonstrated on a real-world extrusion-based bioprinter, using live A549 human lung cancer cells to fabricate bioconstructs. Quality assurance parameters, such as printability and cell viability, were employed to demonstrate the immediate and long-term effects of compromise on the bioprinted construct. The attacks targeting the physical hardware components of the bioprinter were carefully evaluated under controlled conditions. The study provides an initial understanding of the adversarial effects of the potential firmware compromise, aiming to facilitate further security research in the field.**

***Index Terms*—Additive Manufacturing, Bioprinting, Cybersecurity, Supply chain**

I. Introduction

3D bioprinting as a transformative technology is increasingly used in bioengineering, enabling the creation of complex, precise structures with minimal material waste. As an advanced application of additive manufacturing, it allows for layer-by-layer deposition of biomaterials, creating complex geometries that were challenging to achieve with legacy methods like scaffold-based techniques or cell sheet engineering [1]. Bioprinters are particularly valuable in research applications, including regenerative medicine [2] and tissue engineering to transform organ transplantation and medical treatment [3], [4]. Researchers have been using 3D bioprinted models to test drugs and study terminal and infectious diseases, such as cancer [5] and SARS-CoV-2 [6], aiding in vaccine development.

The current market size of 3D bioprinting in 2024 is 4 billion dollars and is projected to grow at an annual rate of 17.2% [7]. As these devices become increasingly popular and are employed in security-critical applications, securing them against adversarial threats becomes crucial. Furthermore, with bioprinters integrated into smart hospital environments, the potential for adversarial access increases, posing significant risks to patient safety and data integrity [8].

The cross-border supply chain involved in the manufacturing, assembly, handling, and shipment of 3D bioprinting equipment has exposed these devices to potential vulnerabilities, especially from untrustworthy third-party manufacturers. Recent examples of supply chain attacks, such as the SolarWinds hack [9] and the pager attacks [10], highlight the severity of such threats. The SolarWinds hack, a large-scale software supply chain attack, involved a malicious software update that compromised US government agencies and private companies, impacting up to 18,000 clients [11]. In the context of embedded devices, the impact of malicious firmware could be even more severe, with state-backed adversaries potentially exploiting such vulnerabilities, as seen in pager attacks [10].

With bioprinting being an evolving technology, there are no standard security practices that are being followed, furthermore, being a cyber-physical system, traditional IT security measures are insufficient for detecting compromises [12], [13], [14], [15], [16], [17]. An attacker accessing the firmware code or binary could maliciously control the bioprinter to achieve their adversarial goals. This study presents a structured categorization of adversarial goals achievable through compromised firmware in 3D bioprinting. With compromised firmware, an attacker could sabotage the bioprinter or manipulate key parameters to degrade the mechanical and biochemical properties of the printed construct. Additionally, firmware compromise could facilitate the exfiltration of sensitive information, such as proprietary design data or patient-specific models [18].

The study also proposes a four-step attack vector, enabling an adversary to compromise the firmware and ensure persistence on the infected device. Leveraging vulnerabilities within the supply chain, we designed and implemented several sabotage attacks. These attacks were then tested on a real-world, extrusion-based bioprinter, using the human lung adenocarcinoma cell line to produce bioconstructs. To evaluate the impact of these attacks on the constructs, we applied multiple quality assurance metrics, including the printability ratio and cell viability, to assess both the immediate and long-term effects of malicious firmware manipulation on bioprinted outcomes. This evaluation demonstrates the potential consequences of firmware compromise in bioprinting applications, underscoring the need for specialized security measures.

The study makes the following key contributions:

- A structured framework for categorizing firmware-based attacks in 3D bioprinting, organized around adversarial goals and tailored to the unique characteristics of the bioprinting process.
- A four-step firmware attack vector designed to maintain persistent compromise on infected 3D bioprinter devices.
- Implementation and evaluation of ten firmware-based attacks, including data theft and sabotage targeting the printer and the bioprinted construct. The attacks were tested on an extrusion-based bioprinter using human cancerous lung cells.

II. Background and Related Work

A. Extrusion based Bioprinting

Extrusion-based Bioprinting (EBB), one of the most widely used bioprinting technologies, relies on mechanisms such as reciprocating screws or pneumatic pistons to drive the extrusion process. These extrusion systems provide precise control over bioink deposition, enabling the creation of complex, bioconstructs. The bioprinting process chain can generally be divided into four main stages; modeling, bioink preparation, printing, and the post-production stage, as shown in Figure 1.

In the first stage, the geometry of the print construct is obtained using tools such as CT scanning, ultrasound, MRI, or other imaging techniques. These tools capture the detailed structure of the target tissue or organ, which is then processed using computer-aided design (CAD) software to create a 3D model. The model is then processed using slicing software, which converts the 3D design into layer-by-layer instructions. Critical parameters such as dimensions, layer height, print speed, temperature settings, etc. are set and controlled at this stage. The slicing software generates a sequential set of commands, known as G-code, which is then transmitted to the bioprinter through any of the available communication channels, including the USB, ethernet connection, or SD card.

In parallel to the 3D modeling, the bioink required for the printing stage is carefully formulated. The composition is optimized for cell growth and structural integrity, ensuring a functional bioprinted construct. In the third stage, the printer firmware uses the G-code file to control actuators and motors and create a layered print construct. Once the printing is complete, the final construct undergoes post-production steps, including UV curing to achieve gelation, quality assurance checks, and incubation to support cell proliferation.

B. Quality Metrics

1) Cell Viability: Cell viability [19] refers to the percentage of live cells after the completion of the bioprinting process. Post-printing the cell viability is accessed by staining the construct with NucBlue (NB) Live Cell Stain, which marks all cells, and propidium iodide (PI), which specifically stains only dead cells. After staining, cells are photographed at multiple locations using a fluorescence microscope. The images are analyzed to count live versus dead cells at standardized thresholds and camera gain settings. For a functional bioprint, cell viability should average above 90% after cell harvest.

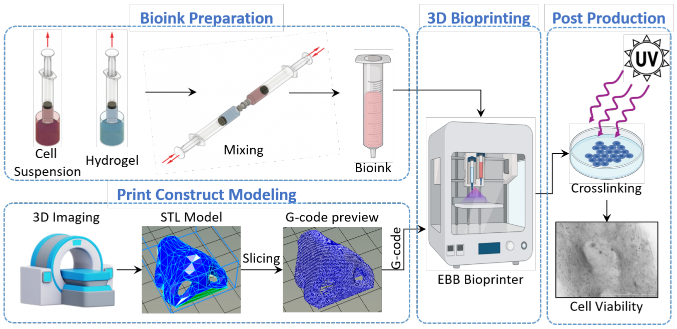


Fig. 1: Bioprinting process chain

2) Printability Ratio: The printability ratio (Pr) [19] is a dimensionless measure used to assess the bioink printability, analyzing the bioink to maintain its shape after extrusion. Mathematically, it is defined as $Pr = L^2/16A$, where 'L' is the perimeter of the cross-sectional area 'A' at the base of the pore of the printed structure. Adapted from the circularity formula, the Pr value helps evaluate the gelation and shape fidelity of printed constructs. A Pr value <1 indicates a higher degree of gelation, indicating the bioink may spread more losing shape fidelity. A Pr >1 corresponds to higher gelation, indicating a stiffer structure. A value closer to 1 indicates optimized gelation for achieving consistent structural fidelity.

C. Related Work

As a newer technology, bioprinting currently lacks research in the literature specifically addressing the adversarial effects of malicious firmware on the bioprinting process. Therefore, the related work in this study focuses first on the bioprinting literature examining how changes in printing parameters affect the bioprinted constructs. Following this, existing studies on firmware attacks related to additive manufacturing (AM) systems are discussed, drawing parallels to understand potential vulnerabilities in bioprinting.

1) Bioprinting Quality Assurance: Studies on bioprinting parameters have been conducted to understand their impact on the final construct and to enhance resolution at micron and sub-micron scales [20]. Parameters including nozzle diameter, print speed, temperature, etc. have been shown to influence printability ratio and cell viability and should therefore be optimized to improve structural and cellular integrity [21]. For example, the stand-off distance affects the filament width and can result in smearing and breaking [22].

Speit et al. [23] investigated the effects of temperature on A549 human lung cancer cells and concluded that optimal cell proliferation is achieved at normal body temperature (37°C). However, under hyperthermic conditions (42–48°C for 30–120 minutes), the cells exhibited significant genotoxic and cytotoxic effects. In addition to cell viability, temperature also affects the rheological properties of bioink [24].

UV radiation enables crosslinking and gelation; however, it can affect cell viability depending on dosage and exposure

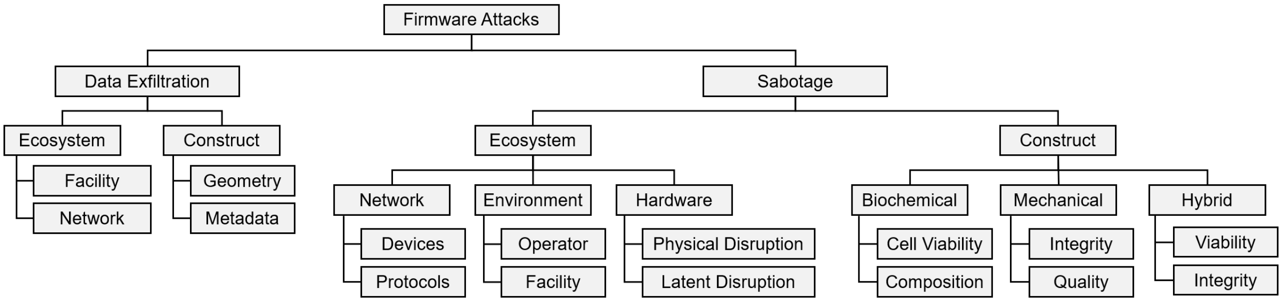


Fig. 2: Firmware attacks classification based on adversarial goals

time [25]. Shorrocks et al. [26] investigated the effects of UVA treatment on HaCaT keratinocytes derived from human skin, finding that shorter UVA doses increased cytotoxic and genotoxic effects. However, these outcomes are influenced by both the cell type and the hydrogel used, and can exhibit varying behaviors [27], [28].

Parameters such as nozzle diameter, print speed, and extrusion pressure affect print quality as studied by Webb et al. [29]. Using the Parameter Optimization Index (POI) to evaluate and optimize these parameters specifically for the hydrogel under study, they concluded that the POI is an effective criterion for optimizing them. Similarly, printability ratio and cell viability have been proposed as key criteria for evaluating print fidelity and cell health [19]. Due to the customization of bioink and the novelty of used techniques, there are currently no official standards [30], [31]. The discussed endpoints help set a measurable standard of good versus bad print and characterize parameter optimization effect on the final construct.

2) Firmware Attacks: In polymer-based AM, studies show how malicious firmware affects printer functionality and print quality (Table I). Xiao [32] demonstrated the feasibility of firmware attacks on 3D printers using an open-source firmware platform. Their work involved an exploit capable of automatically downloading the printer's firmware, modifying nozzle temperature, and re-uploading the compromised firmware.

Moore et al. [33] showed how malicious firmware could compromise the print object by adjusting the extruder feed rate. Similarly, Chhetri et al. [34] leveraged malicious firmware to manipulate kinetic printing parameters. Pearce et al. [35] introduced a Trojan, FLAW3D bootloader, for Marlin-compatible 3D printers. The trojan designed for AVR-based microcontrollers was able to reduce the strength of the printed part by up to 50%. Rais et al. [36] conducted an extensive study on firmware attacks targeting fused filament fabrication (FFF) 3D printers. They proposed an attack taxonomy and demonstrated multiple attacks on the integrity and availability of the printing process. They also showed how an adversary could steal geometry information using malicious firmware.

While these studies focus on polymer-based 3D printers, printing parameters vary significantly across applications, requiring tailored considerations to understand adversarial effects on bioprinting constructs. Therefore, this paper examines the impact of malicious firmware in the context of bioprinting.

Ref.	Material Type	Print Sabotage									Print Surv.	Printer Sabotage
		PS	Th	OG	LT	IF	FK	UV	MC	FS		
[32]	Polymer		✓									✓
[35]	Polymer						✓					
[33]	Polymer			✓			✓					
[34]	Polymer	✓		✓			✓					
[37]	Polymer	✓			✓	✓				✓		
[36]	Polymer	✓	✓	✓	✓	✓	✓			✓	✓	✓
This*	Bioink	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

* PS: Print Speed, Th: Thermal, OG: Object Geometry LT: Layer Thickness, IF: Infill, FK: Filament Kinetic, MC: Material Composition, FS: Fan Speed

TABLE I: Existing studied on 3D printer firmware attacks

III. Classification of Firmware Attacks Goals

Currently, no bioprinting-specific attack categorization exists in the literature. To address this gap and cater to the unique security needs of bioprinting, we propose a structured framework for categorizing firmware attacks organized around adversarial goals. Predominantly focusing on the impact on the bioprinting process, the attacks can be categorized into two major categories: exfiltration of sensitive data and sabotaging the printing process, as shown in Figure 2. The details of each attack category are provided in the following subsections.

A. Data Exfiltration

Attacks aimed at stealing and exfiltrating printing process information can be further divided into two distinct subcategories targeting the printed object metadata and the printing environment information.

1) Construct Information: Intellectual property (IP) data, including the design file, bioink formulation, and printing parameters such as speed, temperature, and UV curing settings, represent highly valuable information that, if disclosed, could lead to substantial financial losses for the company. In addition to stealing or leaking this sensitive data, an adversary leveraging malicious firmware can use this information to "fingerprint" the bioprinting process and gather insights into the bioprinting workflow. The information can then be exploited to design specialized attacks targeting the print geometry, parameters, and material composition to affect cell viability or degrade the structural integrity.

2) Ecosystem Information: Bioprinters can also serve as tools for gathering sensitive facility information. Through malicious firmware, an adversary could use printer hardware components, such as thermal sensors, to capture ambient temperature data or embedded cameras to visually survey the surrounding environment. This data could reveal information such as the research facility layout and operational conditions.

Moreover, malicious firmware can enable the bioprinter to scan for and collect information about other networked devices within the same facility. This reconnaissance provides crucial initial insights into the hardware specifications, network protocols, operating systems, and open ports. An adversary can subsequently use that information to initiate a targeted attack.

B. Sabotage

The sabotage attacks aim to potentially damage the integrity and/or availability of the printing process [38]. Compromised integrity means that the manipulation potentially leads to non-conformance to the user-design specifications, which get noticed in later stages, e.g., after incubation. Attack on availability on the other hand results in the denial of services to the user. An adversary using malicious firmware could perform sabotage activities targeting the printing process, including the ecosystem and the print construct [39], [40], [41].

1) Printing Ecosystem: The printing ecosystem encompasses all components involved in or interacting with the printing process, including printer hardware, the operational environment, and the supporting network infrastructure.

i) Hardware. The printer hardware includes actuators, sensors, and the electronic components enabling the printing. These components can be manipulated to achieve immediate effects on the process (Physical disruption) or degrade their performance to achieve long-term adversarial benefits (Latent disruption). For example, an adversary can damage the printer nozzle by disabling the limits in the firmware, or can shorten the life cycle of the HEPA filtration unit, resulting in degraded performance and a potentially contaminated chamber.

ii) Operational Environment. The operational environment includes both the printing facility and the human operator. Using malicious firmware, an adversary can target printing facilities by contaminating the environment, for instance, by increasing the levels of volatile organic compounds (VOCs) [36]. Additionally, they could raise the nozzle temperature to dangerous levels, potentially causing fire hazards or injury to the human operator [42].

iii) Network. A malicious printer firmware could act as a rogue network entity, launching attacks on other networked devices and communication protocols. For example, it could become part of a botnet to initiate DDoS attacks across the network or attempt to propagate malware to other devices.

2) Print Construct: An adversary could manipulate printing parameters to alter the mechanical or biochemical properties of the bioprinted construct. The footprint of such attacks could vary: it might be overt enough to cause the operator to discard the print entirely, resulting in wasted materials and operational time, or it could be subtle to pass quality assurance checks, only to cause issues after incubation or, worse, post-implantation in a patient.

i) Mechanical Properties. Attacks targeting print geometry can degrade the structural integrity or quality of the construct. For example, introducing small voids in the infill structure could weaken the object in a concealed manner, compromising its structural integrity. Similarly, increasing or decreasing the print speed could result in under- or over-extrusion of the material causing a decrease in the quality of the construct.

ii) Biochemical Properties. These attacks target bioprinted constructs to alter cellular behavior or composition. For example, an adversary could manipulate the HEPA filtration unit, potentially introducing contaminants into the construct. Alternatively, switching between different nozzles could enable changes in material type, thus altering the composition and biochemical properties of the bioprinted structure.

iii) Hybrid. In this category of attacks, parameter alterations impact both the mechanical and biochemical properties of the print construct. For example, tempering the UV curing time or changing the nozzle temperature can degrade cell viability and compromise the structural integrity of the printed construct.

IV. Threat Model

In today's globally interconnected supply chain, components and devices often cross international borders, creating opportunities for adversaries to gain access and compromise them [43], [10]. Once infiltrated, an adversary can execute malicious objectives over extended periods with minimal risk of detection. Previous research has demonstrated how adversaries can infect 3D printing devices by installing malicious firmware [35], [32]. This study builds on that premise, leveraging the malicious supply chain as the primary threat vector.

A. Assumptions

To implement and evaluate the proposed four-stage threat model, we make the following assumptions about the system:

Malicious Supply Chain (A1). A primary assumption underpinning the proposed attacks is that the adversary is embedded within the supply chain and has direct access to the hardware. This scenario might arise if the attacker is an employee at a third-party manufacturing facility or someone with access to the printer during the final production and assembly stages. In this situation, targeting a specific device intended for a particular organization is challenging, so the adversary may opt to compromise all units produced at that facility, ensuring a widespread impact. Alternatively, an adversary involved in delivery and handling could execute a more targeted attack. With access to the shipment process, such an attacker can intercept a particular device, allowing them to customize the malicious firmware for a specific organization, posing a significant risk for high-profile or sensitive installations [10].

Access to Source Code (A2). To introduce and embed persistent malicious modifications, the attacker requires access to the original firmware source code and bootloader. This need is supported by our assumption that the adversary is present during the final stages of the production process, where access to these components is feasible. With access to the source code, a skilled adversary can insert covert changes, recompile the firmware, and then burn it onto the targeted device.

Deployment capability (A3). Once the malicious package is prepared for deployment, the adversary with physical access

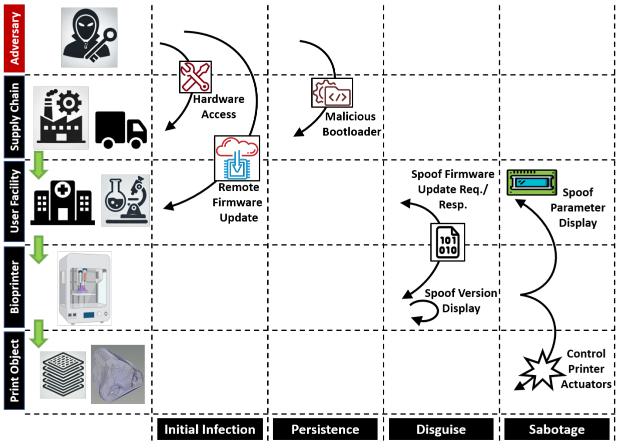


Fig. 3: Firmware persistent attack stages

to the printer hardware installs the compromised binary. We assume that the adversary possesses the capability to flash the malicious binary onto the device. This is feasible in systems where firmware integrity and authentication checks are either absent or contain exploitable vulnerabilities. In scenarios with such security mechanisms, a common approach for the adversary would involve ensuring the malicious firmware is the initial firmware installed on the device.

B. Attack Stages

With current assumptions made, we proposed a four-step firmware attack life cycle as shown in Figure 3 and is detailed in the following subsections.

1) Initial Infection: During the initial infection stage, assuming access to the firmware source, the adversary can inject malware directly into the firmware and load this compromised code onto the printer. This initial infection can be achieved at two levels in the printing process chain:

i Manufacturing Facility: At this level, the adversary, within the manufacturing facility, has access to both the physical hardware and the firmware, allowing them to install the malicious firmware onto the device before distribution.

ii User Facility: Alternatively, the adversary could attack the user's facility by delivering a malicious firmware update. This could be accomplished by exploiting a vulnerability in the update mechanism. This method allows for a more targeted attack, enabling the adversary to compromise specific devices in the field without needing direct access to the hardware.

2) Persistence: To enable a more impactful and resilient attack, we propose a trojan, termed MalBoot. MalBoot modifies the bootloader's standard functionality, particularly in handling firmware updates in the device's flash memory to retain malicious firmware within memory. This persistence mechanism enables the adversary to maintain control over the device, even when attempts are made to overwrite/update the firmware. MalBoot could be deployed through hardware access at the manufacturing facility. The implementation-specific details are provided in Section V-A.

3) Disguise: To evade detection and avoid raising user suspicion, the trojan (MalBoot) not only persists the malicious firmware but also maintains normal request/response communication with the user. MalBoot achieves this by ensuring that user commands and responses appear consistent with expected device behavior. Additionally, MalBoot captures details from any new firmware version uploaded by the user and displays this information on the user interface, giving the appearance that the device is running the latest, authentic firmware. This approach conceals the presence of the malicious firmware and helps maintain a facade of system normality.

4) Sabotage: Once installed, the malicious firmware can perform various actions to accomplish different adversarial objectives. For instance, it could alter critical printing parameters, affecting cell viability and compromising the quality of the printed construct. However, to avoid detection, the malicious firmware reports expected printing parameters to the user to maintain the appearance of normal operation.

V. Implementation Details

For implementation and evaluation, we used the Inkredible+ extrusion-based bioprinter. The printer includes a HEPA filtration unit to ensure a sterile printing environment and features dual extrusion nozzles to print with two different cell types or materials. Additionally, it supports thermal regulation, allowing precise temperature control of the bioink during the printing process. The bioprinter is controlled via Heartwear slicer software, which communicates with the printer device using a USB connection.

The printer's actuators are managed by an open-source Ultimachine RAMBo v1.3 board [44], based on the Arduino Mega 2560 controller. Furthermore, the printer operates using the open-source Marlin firmware [45]. This introduces certain weaknesses critical to the assumptions of this study:

- The use of third-party manufactured hardware (RAMBo board) aligns with **Assumption A1**, demonstrating the feasibility of supply-chain compromises.
- The reliance on open-source firmware (Marlin) supports **Assumption A2**, highlighting the potential for malicious modifications.
- With **Assumptions A1 and A2** validated, and no authentication mechanisms integrated into the RAMBo board, **Assumption A3**, regarding the adversary's ability to install malicious firmware, also holds.

The rest of this section details MalBoot's implementation and deployment for persistent firmware compromise, with attack specifics enabled by this compromise in Section VI.

A. MalBoot

MalBoot was tested on Atmel AVR boards utilizing the stk500v2 [46] protocol over serial communication. Figure 4 illustrates the setup used for installing MalBoot on the RAMBo v1.4 board. The RAMBo board exposes In-Circuit Serial Programming (ICSP) pins, which, when connected to a USBasp programmer, provide direct access to the bootloader program memory. Through the careful analysis of the firmware

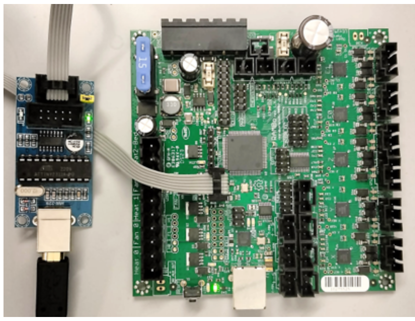


Fig. 4: Setup for testing MalBoot on RAMBo v1.4 (used in Inkredible+ 3D bioprinter).

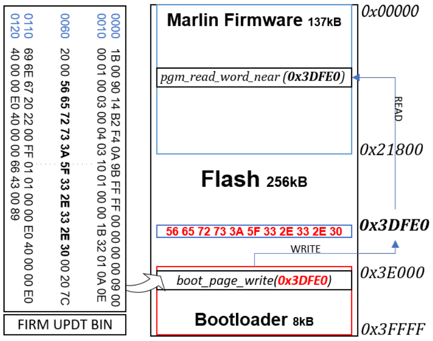

0000 1B 00 90 14 B2 F4 0A 9B FF FF 00 00 00 00 09 00
0010 00 01 00 03 00 04 03 10 01 00 00 1B 32 01 0A 0E
0060 20 00 56 65 72 73 3A 5F 33 2E 33 2E 30 00 20 7C
0110 69 6E 67 20 22 00 FF 01 01 00 00 E0 40 00 00 E0
0120 40 00 00 E0 40 00 00 66 43 00 89
FIRM UPDT BIN
Marlin Firmware 137kB
pgm_read_word_near (0x3DFE0)
Flash 256kB
56 65 72 73 3A 5F 33 2E 33 2E 30
boot_page_write(0x3DFE0)
Bootloader 8kB
READ
WRITE
0x00000
0x21800
0x3DFE0
0x3E000
0x3FFFF

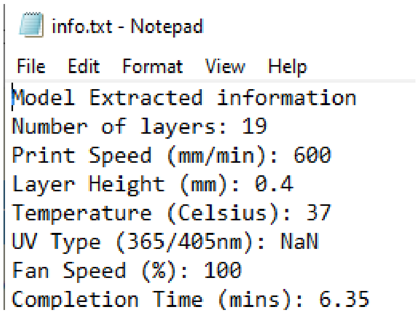
Fig. 5: MalBoot firmware version doctoring

update commands exchanged between the host (PC) and the device (RAMBo), we made the following observations.

First, under default configuration settings, the host system does not authenticate the program (firmware) being updated in the memory. Secondly, the host only validates the CRC value to validate payload integrity. Leveraging from that, the MalBoot manipulates the critical bootloader functionality to perform new firmware updates. Secondly, it maintains normal communication with the user, including interactive and networking features, to avoid raising any alerts.

The bootloader updates the new firmware binary in the flash memory in fixed-size pages (256 bytes). While in programming mode, the bootloader first erases the flash and copies the new binary page by page. MalBoot changes that functionality, keeping the malicious firmware intact and preventing the new update from being pushed to the flash memory. MalBoot, however, accurately calculates the CRC value and responds to the host as normal.

While MalBoot successfully restricts legitimate firmware updates, it enables the adversary to update the infected firmware into the flash memory using an external trigger. This trigger serves as a mechanism to allow controlled malicious firmware updates. Additionally, to facilitate the passing of version information, MalBoot utilizes a free memory region in the flash memory to store and retrieve version data. During legitimate firmware updates, MalBoot captures the version information, stores it in the free memory space, and ensures that it is displayed to the user during the execution of the main program. This process is illustrated in Figure 5.



```
Model Extracted information
Number of layers: 19
Print Speed (mm/min): 600
Layer Height (mm): 0.4
Temperature (Celsius): 37
UV Type (365/405nm): NaN
Fan Speed (%): 100
Completion Time (mins): 6.35
```

Fig. 6: Small ear model extracted metadata

VI. Firmware Attacks

This section outlines the implementation-specific details of the proposed firmware attacks on the Cellink Incredible+ 3D bioprinter. A comprehensive list of 26 potential firmware attacks is presented in Table V (Appendix C), from which ten were selected for implementation and evaluation. The bioink used in the experiments consisted of A549 human lung cancer cells embedded in a 2% gelatin-2% agarose hydrogel. Detailed preparation steps for the bioink are provided in Appendix A.

A. *Object Metadata*

Attack Motive. The primary objective of this attack is to exfiltrate critical print metadata, including parameter settings. This stolen information can be exploited to design targeted attacks [47] or to enable the production of counterfeit bioconstructs.

Attack Category. Data Exfiltration → Construct → Metadata

Implementation. In this attack, the malicious firmware intercepts and logs instructions to capture critical metadata related to various printing profiles. For example, to record the HEPA filter fan speed, the firmware scans for the 'M107' G-code command. Upon detection, the firmware extracts and stores the associated speed value in an array. To optimize memory usage, the attack utilizes Marlin firmware's built-in G-code parsing functionalities. Similarly, other captured profile data is aggregated into the array and stored in the printer's EEPROM memory. When an SD card is inserted into the printer, the firmware transfers the stored metadata onto the card. To enhance stealth, the firmware embeds an authorization token onto the SD card, allowing it to register as a malicious entity with the compromised firmware for future interactions.

Evaluation. The attack was evaluated by executing a G-code file to print a small human ear. Parameters such as the number of layers, print speed, layer height, thermal settings, UV curing, HEPA filter fan speed, and print completion time were successfully captured. The extracted metadata was then transferred to the SD card as shown in Figure 6.

B. *Wear-out Attacks*

Attack Motive. The motive of wear-out attacks is to cause incremental damage to printer hardware, leading to degraded performance over time. Such attacks are designed to exploit the normal wear-and-tear mechanisms of the printer, accelerating their impact to shorten the lifespan of critical components.

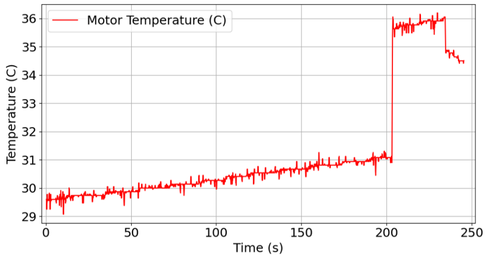


Fig. 7: Motor Temperature values during the attack

Examples include belt abrasion, reduction in HEPA filter lifespan, and decreased motor efficiency, all of which can compromise the printer's reliability and functionality.

Attack Category. Sabotage → Ecosystem → Hardware → Latent Disruption

Implementation. Several stepper motor settings can be altered to cause the motor to jam or hit its endpoints. In this particular attack instance, the 'MICROSTEP_MODES' setting was modified from the default micro-stepping (16) mode to full-step mode (1). This change results in the motor operating at higher speeds while requiring more power to complete each step. With this altered configuration, the adversary then sends an axis home command (G28). Since the firmware is designed for micro-stepping, the modified setting causes the print head to move beyond the designated printing boundaries, ultimately leading to a motor jam, skipped steps, and motor overheating.

Evaluation. To assess the attack's impact, a thermistor is attached to the stepper motor's outer metallic casing, recording temperature over time during the wear-out attack. Figure 7 presents the temperature rise, which can lead to reduced efficiency and degraded print quality.

C. Nozzle Breaking

Attack Motive. While the immediate impact of this attack may seem minor, as the nozzle is inexpensive and easily replaceable, the continuous cycle of troubleshooting and repair can lead to prolonged printer downtime and wasted human effort. This disruption ultimately results in reduced availability of printing services and operational inefficiency.

Attack Category. Sabotage → Ecosystem → Hardware → Physical Disruption

Implementation. The malicious firmware executes the attack by causing the print bed to collide with the nozzle during the printing process. To achieve this, the firmware monitors the printing sequence to identify when the nozzle is actively extruding and moving. Once this condition is detected, the attack is triggered at the start of the third print layer. This is done to achieve maximum disruption from the attack, where, along with mechanical damage, the printing process is also disrupted, resulting in biomaterial wastage. Once the third layer is detected instead of lowering the print bed as per normal operation, the firmware commands the print bed to move upward, forcing a collision with the nozzle. The magnitude of upward bed displacement is kept at twice the current standoff distance, making the attack magnitude enough to deform the nozzle. Once the attack is concluded, the firmware halts the printing process, giving an endpoint hit error. The attack execution details are provided in Appendix B (Algorithm 1).

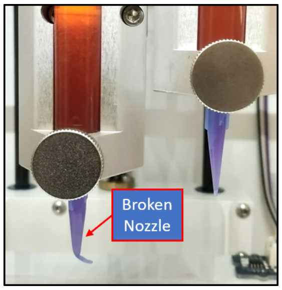


Fig. 8: Nozzle Damage during the printing process

Evaluation. The impact of the attack is illustrated in Figure 8, which shows the resulting damage to the printer nozzle. The attack not only damages the hardware but also results in wasted biomaterial and extended printer downtime.

D. Cartridge Burning

Attack Motive. Similar to the nozzle breaking, this attack targets the printer hardware to cause physical disruption resulting in the unavailability of the printing services to the legitimate user until the new part has been installed.

Attack Category. Sabotage → Ecosystem → Hardware → Physical Disruption

Implementation. The attack is implemented by modifying the firmware to increase the maximum allowable temperature limits, causing the hot-end temperature to exceed safe operating thresholds. The temperature is chosen and set beyond the melting point of the polymer cartridge. The attack is triggered when two conditions are met: the printer is in an inactivity state, and the adversary sends a custom G-code command (M199). The printer's inactivity state is monitored using Marlin's built-in idle detection routine. A custom flag is introduced in this routine to indicate when the printer is idle, which the firmware checks upon receiving the custom G-code command. Once both conditions are satisfied, the firmware invokes the *setTargetHotend()* function to set the hot-end temperature to the maliciously increased limit. The elevated temperature is maintained until the cartridge melts, causing physical disruption and rendering the printer unusable.

Evaluation. The malicious firmware maintains the elevated temperature (260°C) for 3 minutes, leading to the cartridge melting and causing irreversible damage. Once the attack is completed, the firmware restores the temperature to its normal state. The resulting melted cartridge is illustrated in Figure 9.

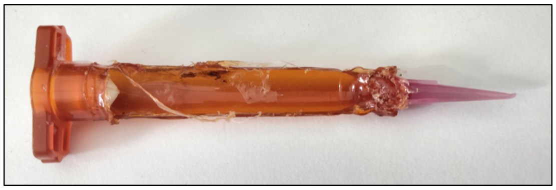
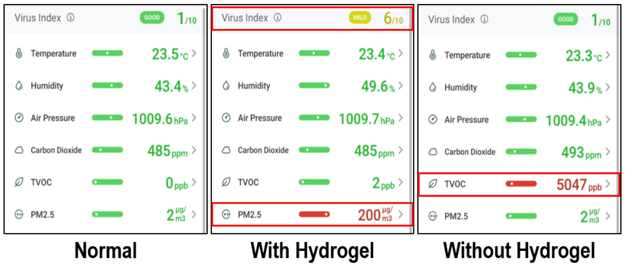


Fig. 9: Impact of high temperature on cartridge



Normal

Virus Index ⓘ GOOD 1/10
Temperature 23.5 °C
Humidity 43.4 %
Air Pressure 1009.6 hPa
Carbon Dioxide 485 ppm
TVOC 0 ppb
PM2.5 2 µg/m3

With Hydrogel

Virus Index ⓘ MILD 6/10
Temperature 23.4 °C
Humidity 49.6 %
Air Pressure 1009.7 hPa
Carbon Dioxide 485 ppm
TVOC 2 ppb
PM2.5 200 µg/m3

Without Hydrogel

Virus Index ⓘ GOOD 1/10
Temperature 23.3 °C
Humidity 43.9 %
Air Pressure 1009.4 hPa
Carbon Dioxide 493 ppm
TVOC 5047 ppb
PM2.5 2 µg/m3

Fig. 10: Air quality stats for contamination attack

E. Facility Air Contamination

Attack Motive. The attack targets the printing facility and the human operator working in the facility by emitting health-injurious volatile organic compounds (VOCs) and microparticles in the environment.

Attack Category. Sabotage → Ecosystem → Environment → Facility

Implementation. Similarly to cartridge burning attack, the firmware looks for the idle state of the printer and uses the printer heating functionality to heat the plastic cartridge to the point where the fumes start to come out of it (210 °C). The temperature is maintained for 2 minutes, cooled down, and increased again. This cycle helps ensure that the cartridge is not noticeably damaged while the VOCs and microparticles contaminate the facility. Due to the odorless nature of these contaminants, the attack can go undetected, resulting in potential health hazards to the workers.

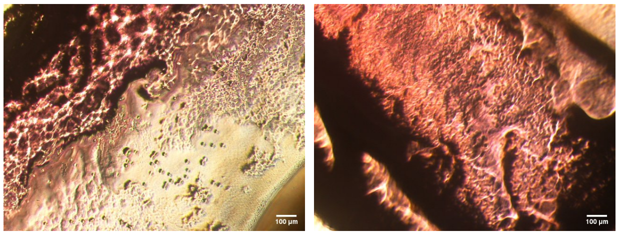
Evaluation. The attack was evaluated by monitoring air quality statistics within the printing chamber during its execution. The results, presented in Figure 10, compare scenarios with and without hydrogel in the cartridge. When hydrogel was present, the PM2.5 levels increased due to the hydrogel boiling and vaporizing at elevated temperatures. Conversely, in the absence of hydrogel, Total VOC levels rose, attributed to the release of fumes from the heated plastic cartridge.

F. Nozzle Switching

Attack Motive. This attack manipulates the composition of the bioprinted construct by maliciously switching between different material types during the printing process. For example, changing the deposition from material type A to material type B results in an incorrect material being used, thereby altering the biochemical properties and potentially compromising the construct's functionality.

Attack Category. Sabotage → Construct → Biochemical

Implementation. During the printing process, the malicious firmware intercepts nozzle-switching commands (e.g., M712 or M721), identifying constructs that utilize multiple material types. Upon detecting such a command, the firmware sets an attack flag and temporarily disables the switching operation (Type B). The firmware then continues printing the current layer with material type A (using nozzle 1). The attack ensures that one or more commands intended for nozzle 2 are executed using nozzle 1, resulting in the deposition of material type A in regions designated for material type B. After command completion, the firmware checks the attack flag and switches to the initial intended nozzle (Type B). Details of attack implementation are provided in Appendix B (Algorithm 2).



(a) Construct with cells (b) Construct without cells

Fig. 11: Nozzle Switching Attack: Construct with one side printed without cells.

Evaluation. The attack was evaluated using two bioinks:

- Type A: A simple hydrogel with no embedded cells.
- Type B: A bioink embedded with A549 cells.

Two Square constructs were printed, where the normal construct consisted of two layers of Type A and a middle layer of Type B. However, in the attacked construct, part of the middle layer was printed with Type A instead of Type B, disrupting the intended composition. The impact of this attack was analyzed under a microscope to assess biochemical disruption. Figure 11a depicts the normal construct composition, where black dots represent A549 cells, while Figure 11b illustrates the disrupted construct composition without cells.

G. Thermal Attack

Attack Motive. This attack targets cell viability and printability by manipulating temperature conditions. Elevated temperatures induce hyperthermia, compromising cell viability by causing cellular damage or death. Conversely, maintaining temperatures below the optimal range leads to premature bioink gelation, resulting in nozzle clogging and a poor printability ratio.

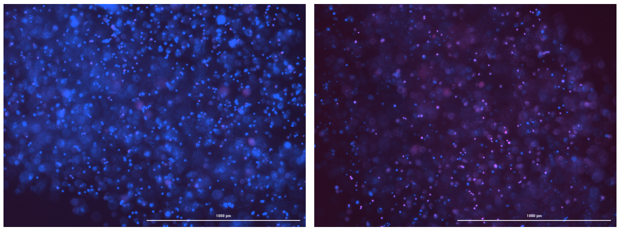
Attack Category. Sabotage → Construct → Hybrid

Implementation. The attack is initiated during the active printing process. Upon verifying that printing is in progress, the firmware introduces a trigger to modify the nozzle's current temperature by a predefined factor:

- **Increase temperature:** Add a positive offset to the current temperature settings (e.g. +15 °C), inducing hyperthermia conditions and causing cell death.
- **Decrease temperature:** Add a negative offset to the current temperature settings (e.g. -15 °C), resulting in poor Pr value.

To avoid detection, the malicious firmware continues to display the unaltered temperature settings on the user interface.

Evaluation. To evaluate the impact of the attack, A549 cells were embedded within a gelatin-agarose hydrogel at 37 °C. Two sets of square constructs were printed: one using normal temperature settings and another at an elevated temperature of



(a) Control Construct with NB and PI (b) Attacked Construct with NB and PI

Fig. 12: Cell viability after 24 hrs for normal and attacked construct

Day #	Print Construct	DAPI count (Total)	TR (Dead)	Live cells	% Viability
Control Day 0	Construct 1	1153	34	1119	97
	Construct 2	589	8	581	98.6
Control Day 1	Construct 1	3235	402	2833	87.6
	Construct 2	3539	344	3195	90.3
Attack Day 0	Construct 1	2796	52	2744	98.1
	Construct 2	2533	37	2496	98.5
Attack Day 1	Construct 1	1997	562	1435	71.8
	Construct 2	2780	993	1787	64.3

TABLE II: Cell viability of control (37°C) and attacked (60°C) print constructs.

60 °C, manipulated by malicious firmware. Extrusion pressure was manually adjusted to ensure comparable print quality metrics across both sets. Cell viability was assessed using a live/dead assay at 0 and 24 hours post-print. The results are tabulated in Table II and visualized in Figure 12, wherein the blue cells indicate live cells and the red represents dead cells. For the control construct, cell viability was 97.8% at Day 0 and decreased slightly to 88.9% at Day 1, consistent with expected post-printing conditions. However, the attacked construct exhibited a significant reduction in viability, with initial measurements of 98.3% on Day 0 dropping to 68.1% on Day 1. This marked decline indicates substantial cell death due to hyperthermia conditions induced by the malicious firmware.

H. Construct Contamination

Attack Motive. The attack targets the biochemical properties of the print by potentially making the printing process vulnerable to external contaminants.

Attack Category. Sabotage → Construct → Biochemical

Implementation. To maintain a sterile printing environment, the printer uses a HEPA filtration unit in the printing chamber. If the functionality of the filtration unit is compromised, it can expose the construct to environmental pathogens. The malicious firmware achieves this by targeting the fan attached to the filtration unit controlled by the $manage\_cleanchamber()$ function. The function maintains the sterile conditions by keeping the filtration unit active even during idle state. The firmware manipulates the function behavior by reducing the fan speed to 1/4 of the original value. The malicious firmware also captures and manipulates the fan speed command (M780 S_x) communicated during printing.

Evaluation. To evaluate the attack's effect on the bioprinted construct, we used A549 cells to print a square-shaped bioconstruct. However, due to the unavailability of advanced microbial detection techniques such as fluorescence microscopy or real-time PCR, the specific effects of contaminants on the printed construct could not be conclusively determined.

I. Print Speed

Attack Motive. The objective of this attack is to compromise the structural integrity of the print construct by manipulating the print speed parameter, leading to conditions such as over-extrusion or under-extrusion.

Attack Category. Sabotage → Construct → Mechanical

Implementation. The attack is implemented with minimal changes to the firmware by intercepting G-code move commands (G0/G1). For every move command containing a speed parameter (F), the malicious firmware adjusts the feed rate by applying a predefined multiplication factor:

- **Increase Feed Rate:** Multiply by a factor greater than 1 (e.g., 2 to double the speed).
- **Decrease Feed Rate:** Multiply by a factor less than 1 (e.g., 0.5 to halve the speed).

The modified firmware parses the F parameter in each move command and updates the feed rate accordingly. Algorithm 3 provides the implementation details of the attack.

Evaluation. The manipulation of print speed directly affects material deposition, leading to under- or over-extrusion, which negatively impacts the printability ratio (Pr). To investigate this, we conducted three instances of attacks where the print speed parameter was multiplied by factors of 1.25, 1.5, and 1.75. The outcomes of these manipulations are presented in Figure 13, showcasing the variations in print quality. Each experiment was repeated four times to ensure reproducibility, and the average Pr and POI values for both normal and attacked print constructs are summarized in Table III. It is important to note that along with decreased Pr and POI values at higher speeds, we get more broken constructs.

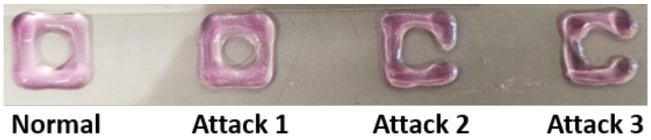


Fig. 13: Constructs at different magnitude of print speed attack

Construct	Speed (mm/s)	Broken Constructs	Pr Avg.	POI Avg.
Normal	600	0	0.83	0.305
Attack 1	750	2	0.81	0.275
Attack 2	900	2	0.82	0.272
Attack 3	1050	3	0.82	0.15

TABLE III: Pr and POI values for print speed attack averaged over four runs

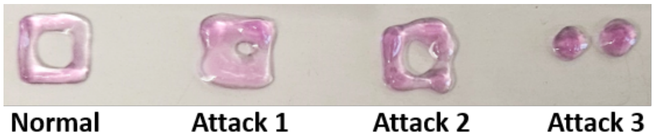


Fig. 14: Constructs at different magnitude of standoff attack

Construct	Attack Magnitude (mm)	Broken Constructs	Pr Avg.	POI Avg.
Normal	0	0	0.831	0.305
Attack 1	0.5	1	0.82	0.240
Attack 2	1	4	-	-
Attack 3	1.5	4	-	-

TABLE IV: Pr and POI values for standoff distance attack averaged over four runs

J. Standoff Distance

Attack Motive. This attack compromises the structural integrity of the print construct by manipulating the standoff distance (distance between nozzle and print bed). Such manipulations could lead to weakened inter-layer adhesion, voids, and under/over-extrusion conditions.

Attack Category. Sabotage → Construct → Mechanical

Implementation. Similarly to print speed attack, the firmware intercepts the G-code move commands (G0/G1), and for every move command containing the z-axis parameter (Z), the malicious firmware adjusts the z-coordinate by applying a predefined factor:

- **Increase standoff distance:** Add a positive factor to the Z value (e.g., +0.5 mm), resulting in weaker inter-layer adhesion and voids.
- **Increase standoff distance:** Add a negative factor to the Z value (e.g., -0.5 mm), causing over-extrusion and potential collisions with the printed construct.

The modified firmware parses the Z parameter in each move command and updates the z-coordinate accordingly. See Appendix B (Algorithm 4) for attack details.

Evaluation. The manipulation of standoff distance affects material deposition, causing material dragging and leading to under/over-extrusion, negatively impacting the Pr value. To evaluate this, we conducted three instances of attacks in which the standoff distance was increased by factors of 0.5, 1.0, and 1.5. The results of these manipulations are presented in Figure 14. It can be seen that with the increasing attack magnitude, the construct deformity becomes more obvious. Each experiment was repeated four times, and the average Pr values for both normal and attacked constructs are summarized in Table IV.

VII. Firmware Attack Countermeasures

To mitigate firmware attacks on 3D bioprinters, a robust and integrated approach combining prevention, detection, and mitigation strategies is essential. The following countermeasures outline best practices and measures to safeguard against firmware attacks:

- **Firmware Authentication** To prevent the installation of malicious firmware, techniques such as secure boot mechanisms, cryptographic signing of firmware files, hardware-based authentication, and the integration of security modules like Trusted Platform Modules (TPM) or Hardware Security Modules (HSM) should be implemented [48], [49].
- **Secure Supply Chain:** Establishing a trusted supply chain is essential and can be achieved by sourcing hardware and software components from trusted vendors, conducting regular audits, and using secure distribution channels. Blockchain technology can further enhance supply chain security by enabling real-time tracking, vendor authentication, and preventing counterfeit components through tamper-proof, decentralized records [50].
- **Runtime Monitoring:** In-situ physical process monitoring techniques should be implemented to verify firmware integrity during operation and detect anomalies in the printing process [51]. These techniques can serve as an additional layer of defense by identifying signs of adversarial manipulations in real-time.
- **Access Control:** Strict authentication and authorization mechanisms should be enforced on control software used to install firmware updates. Physical access to the bioprinter hardware should also be secured using access control policies to limit exposure to unauthorized individuals.
- **Security Policies:** As 3D bioprinting continues to evolve, it is imperative to establish industry-wide best practices and security standards. These policies should address firmware development, hardware security, forensics readiness [52], and operational procedures to minimize vulnerabilities and thwart adversarial attempts.

These multifaceted measures can significantly enhance the security of 3D bioprinting systems, fostering a safe and reliable printing environment.

VIII. Conclusion

With the increasing globalization of the supply chain, adversarial access to hardware components for installing malware and trojans has become increasingly practical. 3D bioprinters, due to their use in security-critical applications, are particularly susceptible to such threats. This study presented a novel approach to categorizing firmware attacks on 3D bioprinters and proposed a four-step persistent threat model. We implemented ten unique firmware attacks targeting the printer and the printed construct, evaluating them on a real-world extrusion-based bioprinter using human lung cancer cells (A549) to fabricate bioconstructs. By assessing key quality assurance parameters, such as printability ratio (Pr) and cell viability, we demonstrated the immediate and long-term impacts of these attacks. This work provides a foundational guide for researchers to understand the adversarial effects of firmware-based threats and emphasizes the need to develop robust security solutions for safeguarding 3D bioprinting systems.

References

[1] C.-Y. Liaw and M. Guvendiren, "Current and emerging applications of 3d printing in medicine," *Biofabrication*, vol. 9, no. 2, p. 024102, 2017.

[2] S. Vijayavenkataraman, W.-C. Yan, W. F. Lu, C.-H. Wang, and J. Y. H. Fuh, "3d bioprinting of tissues and organs for regenerative medicine," *Advanced Drug Delivery Reviews*, vol. 132, pp. 296–332, 2018, 3D-Bioprinting and Micro-/Nano-Technology: Emerging Technologies in Biomedical Sciences.

[3] Y. S. Zhang, K. Yue, J. Aleman *et al.*, "3D Bioprinting for Tissue and Organ Fabrication," *Annals of biomedical engineering*, vol. 45, pp. 148–163, 2017.

[4] S. V. Murphy and A. Atala, "3d bioprinting of tissues and organs," *Nature biotechnology*, vol. 32, no. 8, pp. 773–785, 2014.

[5] R. Augustine, S. N. Kalva, R. Ahmad, A. A. Zahid, S. Hasan, A. Nayeem, L. McClements, and A. Hasan, "3d bioprinted cancer models: Revolutionizing personalized cancer therapy," *Translational Oncology*, vol. 14, no. 4, p. 101015, 2021.

[6] B. A. de Melo, J. C. Benincasa, E. M. Cruz, J. T. Maricato, and M. A. Porcionatto, "3d culture models to study sars-cov-2 infectivity and antiviral candidates: From spheroids to bioprinting," *Biomedical Journal*, vol. 44, no. 1, pp. 31–42, 2021.

[7] "Global 3D Bioprinting Market Size," *RootsAnalysis*, 2024, Latest Accessed: Dec, 2024. [Online]. Available: https://www.rootsanalysis.com/reports/3d-bioprinting-market/182.html

[8] J. C. Isichei, S. Khorsandroo, and S. Desai, "Cybersecurity and privacy in smart bioprinting," *Bioprinting*, p. e00321, 2023.

[9] R. Alkhadra, J. Abuzaid, M. AlShammari, and N. Mohammad, "Solar winds hack: In-depth analysis and countermeasures," in *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. IEEE, 2021, pp. 1–7.

[10] C. Sheng, W. Ma, Q.-L. Han, W. Zhou, X. Zhu, S. Wen, Y. Xiang, and F.-Y. Wang, "Pager Explosion: Cybersecurity Insights and Afterthoughts," *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 12, pp. 2359–2362, 2024.

[11] TechTarget, "SolarWinds hack explained: Everything you need to know," 2023, Latest Accessed: Dec, 2024. [Online]. Available: https://www.techtarget.com/whatis/feature/SolarWinds-hack-explained-Everything-you-need-to-know

[12] H. Alam, M. S. Yaqub, and I. Nadir, "Detecting iot attacks using multi-layer data through machine learning," in *2022 Second International Conference on Distributed Computing and High Performance Computing (DCHPC)*, 2022, pp. 52–59.

[13] B. Imran, B. Afzal, A. H. Akbar, M. Ahsan, and G. A. Shah, "MISA: Minimalist Implementation of oneM2M Security Architecture for Constrained IoT Devices," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[14] M. S. Yaqub, H. Mahmood, I. Nadir, and G. A. Shah, "An ensemble approach for iot firmware strength analysis using stride threat modeling and reverse engineering," in *2022 24th International Multitopic Conference (INMIC)*, 2022, pp. 1–6.

[15] B. Imran, M. Ahsan, A. H. Akbar, and G. A. Shah, "D4GW: DTLS for gateway multiplexed application to secure MQTT(SN)-based pub/sub architecture," *Internet of Things*, vol. 26, p. 101172, 2024.

[16] Muhammad, Ahsan and Afzal, Bilal and Imran, Bilal and Tanwir, Asim and Akbar, Ali Hammad and Shah, Ghalib, "oneM2M Architecture Based Secure MQTT Binding in Mbed OS," in *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2019, pp. 48–56.

[17] M. Ahsan and M. Ali, "Lsstk: Lightweight solution to preventing stack from buffer overflow vulnerability," in *2023 17th International Conference on Open Source Systems and Technologies (ICOSST)*, 2023, pp. 1–7.

[18] M. Ahsan and I. Ahmed, "WattShield: A Power Side-Channel Framework for Detecting Malicious Firmware in Fused Filament Fabrication," in *2025 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 2025.

[19] S. Tian, H. Zhao, and N. Lewinski, "Key parameters and applications of extrusion-based bioprinting," *Bioprinting*, vol. 23, p. e00156, 2021.

[20] T. Zandrini, S. Florczak, R. Levato, and A. Ovsianikov, "Breaking the resolution limits of 3d bioprinting: future opportunities and present challenges," *Trends in Biotechnology*, 2022.

[21] J. Adhikari, A. Roy, A. Das, M. Ghosh, S. Thomas, A. Sinha, J. Kim, and P. Saha, "Effects of processing parameters of 3d bioprinting on the cellular activity of bioinks," *Macromolecular bioscience*, vol. 21, no. 1, p. 2000179, 2021.

[22] Z. Fu, V. Angeline, and W. Sun, "Evaluation of printing parameters on 3d extrusion printing of pluronic hydrogels and machine learning guided parameter recommendation," *International journal of bioprinting*, vol. 7, no. 4, 2021.

[23] G. Speit and P. Schütz, "Hyperthermia-induced genotoxic effects in human a549 cells," *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis*, vol. 747, pp. 1–5, 2013.

[24] Y. Zhao, Y. Li, S. Mao, W. Sun, and R. Yao, "The influence of printing parameters on cell survival rate and printability in microextrusion-based 3d cell printing technology," *Biofabrication*, vol. 7, no. 4, p. 045002, nov 2015.

[25] A. D. Rouillard, C. M. Berglund, J. Y. Lee, W. J. Polacheck, Y. Tsui, L. J. Bonassar, and B. J. Kirby, "Methods for photocrosslinking alginate hydrogel scaffolds with high cell viability," *Tissue Engineering Part C: Methods*, vol. 17, no. 2, pp. 173–179, 2011, pMID: 20704471.

[26] J. Shorrocks, N. D. Paul, and T. J. McMillan, "The dose rate of uva treatment influences the cellular response of hacat keratinocytes," *Journal of investigative dermatology*, vol. 128, no. 3, pp. 685–693, 2008.

[27] T. Billiet, E. Gevaert, T. De Schryver, M. Cornelissen, and P. Dubruel, "The 3d printing of gelatin methacrylamide cell-laden tissue-engineered constructs with high cell viability," *Biomaterials*, vol. 35, no. 1, pp. 49–62, 2014.

[28] I. Mironi-Harpaz, D. Y. Wang, S. Venkatraman, and D. Seliktar, "Photopolymerization of cell-encapsulating hydrogels: Crosslinking efficiency versus cytotoxicity," *Acta Biomaterialia*, vol. 8, no. 5, pp. 1838–1848, 2012.

[29] B. Webb and B. J. Doyle, "Parameter optimization for 3d bioprinting of hydrogels," *Bioprinting*, vol. 8, pp. 8–12, 2017.

[30] E. Kelly, "FDA regulation of 3D-printed organs and associated ethical challenges," *U. Pa. L. Rev.*, vol. 166, p. 515, 2017.

[31] J. M. Bliley, D. J. Shiwarski, and A. W. Feinberg, "3D-bioprinted human tissue and the path toward clinical translation," *Science Translational Medicine*, vol. 14, no. 666, p. eabo7047, 2022.

[32] C. Xiao, "Security attack to 3d printing," *xFocus Information Security Conference*, 2013.

[33] S. B. Moore, W. B. Glisson, and M. Yampolskiy, "Implications of malicious 3d printer firmware," in *Proceedings of the 50th Hawaii International Conference on System Sciences*, 2017.

[34] S. R. Chhetri, A. Canedo, and M. A. Al Faruque, "Kcad: kinetic cyber-attack detection method for cyber-physical additive manufacturing systems," in *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2016, pp. 1–8.

[35] H. Pearce, K. Yanamandra, N. Gupta, and R. Karri, "Flaw3d: A trojan-based cyber attack on the physical outcomes of additive manufacturing," *IEEE/ASME Transactions on Mechatronics*, 2022.

[36] M. H. Rais, M. Ahsan, and I. Ahmed, "{SOK}: 3D Printer Firmware Attacks on Fused Filament Fabrication," in *18th USENIX WOOT Conference on Offensive Technologies (WOOT 24)*, 2024, pp. 263–282.

[37] Y. Gao, B. Li, W. Wang, W. Xu, C. Zhou, and Z. Jin, "Watching and safeguarding your 3d printer: Online process monitoring against cyber-physical attacks," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 3, pp. 1–27, 2018.

[38] M. Ahsan, M. H. Rais, and I. Ahmed, "SOK: Side Channel Monitoring for Additive Manufacturing - Bridging Cybersecurity and Quality Assurance Communities," in *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, 2023, pp. 1160–1178.

[39] M. H. Rais, M. Ahsan, V. Sharma, R. Barua, R. Prins, and I. Ahmed, "Low-magnitude infill structure manipulation attacks on fused filament fabrication 3d printers," in *International Conference on Critical Infrastructure Protection*, 2022, pp. 205–232.

[40] Y. Forihat, C. Taylor, R. A. Awad, and I. Ahmed, "Security Assessment of an LBP16-Protocol-Based Computer Numerical Control Machine," in *International Conference on Critical Infrastructure Protection*. Springer, 2024, pp. 65–84.

[41] M. H. Rais, M. Ahsan, and I. Ahmed, "Sabotaging material extrusion-based 3D printed parts through low-magnitude kinetic manipulation attacks," *ACM Transactions on Cyber-Physical Systems*, vol. 9, no. 1, pp. 1–26, 2025.

[42] M. Yampolskiy, A. Skjellum, M. Kretzschmar, R. A. Overfelt, K. R. Sloan, and A. Yasinsac, "Using 3d printers as weapons," *International Journal of Critical Infrastructure Protection*, vol. 14, pp. 58–71, 2016.

[43] R. Tsang, D. Joseph, Q. Wu, S. Salehi, N. Carreon, P. Mohapatra, and H. Homayoun, “Fandemic: Firmware attack construction and deployment on power management integrated circuit and impacts on iot applications.” in *NDSS*, 2022.

[44] “Ultimachine — RAMBo v1.3,” Latest Accessed: Dec, 2024. [Online]. Available: https://ultimachine.com/products/rambo-1-3

[45] S. Lahteine, R. Neufeld, C. Pepper, B. Kuhn, and E. v. d. Zalm, “Home — Marlin Open Source Firmware,” 2011, Latest Accessed: Dec, 2024. [Online]. Available: https://marlinfw.org/

[46] Atmel, “AVR068: STK500 Communication Protocol,” Latest Accessed: Dec, 2024. [Online]. Available: https://www.diericx.net/downloads/STK500v2.pdf

[47] S. Belikovetsky, M. Yampolskiy, J. Toh, J. Gatlin, and Y. Elovici, “dr0wned–{Cyber-Physical} attack with additive manufacturing,” in *11th USENIX Workshop on Offensive Technologies (WOOT 17)*, 2017.

[48] P. H. Coppock, M. K. Yacoub, B. L. Qin, A. J. Daftardar, Z. Tolaymat, and V. J. Mooney, “Hardware root-of-trust-based integrity for shared library function pointers in embedded systems,” *Microprocessors and Microsystems*, vol. 79, p. 103270, 2020.

[49] J. Frazelle, “Securing the boot process: The hardware root of trust,” *Queue*, vol. 17, no. 6, p. 5–21, feb 2020.

[50] V. Hassija, V. Chamola, V. Gupta, S. Jain, and N. Guizani, “A survey on supply chain security: Application areas, security threats, and solution architectures,” *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6222–6246, 2020.

[51] M. Ahsan, E. Pak, K. Jackson, M. H. Rais, B. Najarro-Blancas, N. Lewinski, and I. Ahmed, “BioSaFe: Bioprinting Security Framework for Detecting Sabotage Attacks on Printability and Cell Viability,” in *40th Annual Computer Security Applications Conference (ACSAC’24)*. IEEE, 2024.

[52] M. H. Rais, M. Ahsan, and I. Ahmed, “Fromepp: Digital forensic readiness framework for material extrusion based 3d printing process,” *Forensic Science International: Digital Investigation*, vol. 44, p. 301510, 2023.

Appendix

A. Bioink Preparation

For the preparation of the bioink, the first step is to prepare a hydrogel. After testing multiple hydrogel formulations, we selected a gelatin-agarose solution due to its superior strength, relatively shorter preparation time, and the advantage of not requiring a chemical crosslinking step. A polymer concentration of 2% agarose and 2% gelatin weight per volume was used because of its smaller swelling ratio.

To prepare the hydrogel, a two-step process was employed, starting with the preparation of agarose and gelatin solutions. For the agarose solution, 2% (w/v) agarose was dissolved in Complete Dulbecco’s Modified Eagle Medium (DMEM), supplemented with 5% Fetal Bovine Serum (FBS) and 1% Penicillin-Streptomycin (Pen-Strep), and heated to 80°C with continuous stirring for 1 hour with a magnetic stir bar stirring at 1000 rpm. Simultaneously, a 2% (w/v) gelatin solution was prepared in Complete DMEM, heated to 55°C, and stirred continuously for 1 hour. The two heated solutions were then combined in a beaker, maintained at 55°C, and stirred for an additional hour to ensure homogeneity. The resulting hydrogel was carefully extracted using a sterile syringe, ensuring air bubbles were removed, and transferred to a bioprinting cartridge for subsequent use.

The A549 cells, shown in Figure 15, were cultured in Complete DMEM under incubation conditions of 37°C with 5% CO_2. With a doubling time of 24 hours, the cells were fed daily with fresh media and harvested at 80% confluency. After splitting and counting, a cell pellet of the desired quantity was prepared for bioink mixing. The pellet was suspended in 0.2 ml of sterile media and transferred to a 3 ml syringe, which was then connected to a 2 ml cartridge prefilled with pre-warmed hydrogel using a syringe coupler. A549 cells were embedded in the hydrogel at a concentration of 1 million cells per ml of hydrogel. The suspension and hydrogel were mixed by gently pushing and pulling the contents 3–5 times to achieve an even distribution while avoiding air bubbles, preparing the bioink for printing.

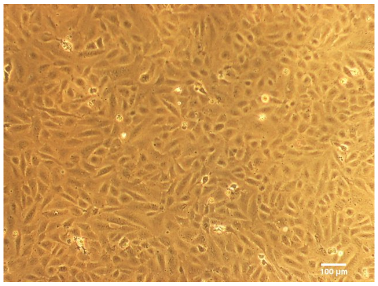


Fig. 15: A549 human lung cancer cells

B. Firmware attacks algorithms

Algorithm 1 Nozzle Break Attack

```
Input: G-code Commands
Output: Broken printer nozzle
Procedure:
if cmd.start(G0)||cmd.start(G1) then          ▷ G-code G0/G1
    if cmd.has(Z) then                         ▷ Parse Z parameter
        $Flag \leftarrow Flag + 1$
        $Z_{cord} \leftarrow extract.value(cmd)$
        $S_d = S_d + Z_{cord}$
    end if
    if Flag == 3 then                          ▷ Third layer detected
        $Z_{cord} \leftarrow -2 * S_d$
        Pass modified $Z_{cord}$ to motion planner
    end if
end if
```

Algorithm 2 Nozzle Switching Attack

```
Input: G-code Commands
Output: Change construct composition
Procedure:
if cmd.start(M721)||cmd.start(M712) then
    Flag ← $True$                               ▷ Enable attack flag
    continue                                   ▷ Skip switch cmd execution
end if
if cmd.start(G0)||cmd.start(G1) then          ▷ G-code G0/G1
    if cmd.has(G1) then                        ▷ Extrusion cmd
        Execute move command
        if Flag == $True$ then
            Execute Nozzle switch
            Flag ← $False$                      ▷ Disable attack flag
        end if
    end if
end if
```

Algorithm 3 Print Speed Attack

```
Input: Attack Factor ($A_s$)
Output: Modified feed_rate
Procedure:
if cmd.start(G0)||cmd.start(G1) then        ▷ G-code G0/G1
    if cmd.has(F) then                      ▷ Parse F parameter
        feed_rate ← extract.value(cmd) $*\ A_s$
    end if
end if
```

Algorithm 4 Standoff Distance Attack

```
Input: Attack Factor ($A_z$)
Output: Modified $Z_{cord}$
Procedure:
if cmd.start(G0)||cmd.start(G1) then        ▷ G-code G0/G1
    if cmd.has(Z) then                      ▷ Parse Z parameter
        $Z_{cord}$ ← extract.value(cmd) $+ A_z$
    end if
    Pass modified $Z_{cord}$ to motion planner
end if
```

C. Appendix: Attacks Description

A comprehensive list of potential firmware attacks on 3D bioprinters along with the category and brief attack descriptions is provided in Table V.

No.	Attack Name	Attack Goal	Explanation
1	Object Geometry	Construct Data Exfiltration	Stealing the geometry information of the printed object
2	Object Metadata	Construct Data Exfiltration	Stealing the object metadata including print speed, number of layers, composition, layer height, temperature, etc.
3	Facility information	Ecosystem Data Exfiltration	Exfiltrating facility information such as ambient temperature, camera pictures, etc.
4	Networked devices	Ecosystem Data Exfiltration	Exfiltrating networked devices information including OS, open Ports, communication protocols, etc.
5	Print Bed Damage	Physical Disruption	Causing damage\breaking the printer bed to disrupt physical process
6	Limit Switches Breaking	Physical Disruption	Hitting the print bed with limit switches to mechanically break it
7	Cartridge Burning	Physical Disruption	Increasing the nozzle temperature beyond the melting point of polymer cartridge to cause burning.
8	Solenoid	Physical Disruption	Maliciously switching the solenoids at high speed to cause mechanical issues
9	Nozzle Breaking	Physical Disruption	Moving the print bed beyond hardware limits to hit and break the printer nozzle
10	Motor Belt Wear out	Latent Disruption	Moving the printer nozzle at high speed during non-activity periods to cause mechanical wear out of belts
11	Motor Wearout	Latent Disruption	Keeping the motor in an active state during printer inactivity to increase its temperature resulting in performance degradation or potential burnout
12	HEPA filter life cycle	Latent Disruption	Decreasing the HEPA filter performance by continuously turning it on during non-activity periods
13	Facility Air contamination	Environment Sabotage	Melting the polymer-based cartridge to increase VOCs in the printing facility
14	Operator Injury/Harm	Environment Sabotage	Maliciously increasing the nozzle temperature to cause burns to unsuspecting printer operator
15	DDoS	Network Sabotage	Compromising the networked entities to launch DDoS attacks
16	Voids	Mechanical Properties	Intentionally removing the material to cause voids in the print geometry
17	Layer Height	Mechanical Properties	Changing layer height to compromise the structural integrity of the print construct
18	Standoff Distance	Mechanical Properties	Changing the distance between nozzle and print bed to compromise print structural integrity
19	Print Speed	Mechanical Properties	Increasing/decreasing the nozzle print speed to cause under/over-extrusion of the material
20	Geometry Scaling	Mechanical Properties	Scaling up or down the print construct geometry
21	Trajectory unsync.	Mechanical Properties	Unsynchronize the nozzle trajectory by maliciously changing the x and y motor speed profiles
22	Nozzle Switching	Biochemical Properties	Changing the print material composition by switching between nozzles
23	Construct Contamination	Biochemical Properties	Manipulating HEPA filtration unit fan speed to cause pathogens contaminating the construct
24	Nozzle Temperature	Hybrid	Changing nozzle temperature to compromise cell viability and structural integrity
25	UV Curing	Hybrid	Changing the intensity or duration of the UV curing process
26	Multistage	Hybrid	Maliciously changing multiple printing parameters to affect cell viability and structural integrity

TABLE V: Attack description and their categories based on adversarial goals