International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2024: 7.101

Securing 3D Printing: Understanding the Hidden Risks of G-Code Manipulation in Additive Manufacturing

Vismit Sudhir Rakhecha (Druk)

Principal Information Security Engineer
Email: *rvismit[at]gmail.com*

Abstract: ***3D printing has become a cornerstone of modern manufacturing, enabling rapid prototyping, distributed production, and innovative applications across various industries. However, as additive manufacturing (AM) systems become more interconnected, they are increasingly vulnerable to cyber threats. One of the most critical yet underexplored threats is G-code injection attacks—malicious modifications to the machine-readable instructions that control 3D printers. Attackers can alter G-code to introduce structural weaknesses, embed hidden malware, or sabotage printed components without detection. This whitepaper examines the risks posed by G-code manipulation, presents real-world case studies, and explores mitigation strategies to enhance cybersecurity in additive manufacturing.***

Keywords: G-code security, Secure 3D printing, 3D printer cybersecurity, Additive manufacturing security, Cyber-physical security in 3D printing

1. Introduction

G-code is the standard language used to control 3D printers, dictating movement, extrusion, temperature, and other critical parameters. When a 3D model (usually in STL format) is processed by slicing software, it generates a G-code file that instructs the printer on how to create the object layer by layer. While G-code is essential for precision in 3D printing, its **lack of built-in security mechanisms** makes it a prime target for cyberattacks. Unlike traditional IT systems, where encryption and authentication are standard, most 3D printers operate in **open environments** without strong safeguards against unauthorized code modifications.

2. What Does G-Code Do?

2.1 Translates Designs into Actions: Slicing software (e.g., Cura, PrusaSlicer) converts 3D models (STL/OBJ files) into G-Code, breaking the model into layers and generating commands for:

1) Nozzle movement along X, Y, and Z axes.
2) Extrusion of filament (how much, how fast).
3) Temperature control (hotend, bed).
4) Fan speed, pauses, and other auxiliary functions

2.2 Controls Hardware: Every motor, heater, and sensor in the printer responds to G-Code directives. For example:

1) G1 X100 Y100 E5 moves the nozzle to (100,100) while extruding 5mm of filament.
2) M140 S60 sets the print bed temperature to 60°C.

3. Key Components of G-Code

1) G-Commands: Control motion and positioning.
2) G0/G1: Rapid or controlled linear movement.
3) G28: Auto-home the printer.
4) G90/G91: Switch between absolute and relative positioning.
5) M-Commands: Manage hardware functions.
6) M104: Set extruder temperature.
7) M106: Turn on cooling fans.
8) M84: Disable stepper motors.
9) Parameters: Variables like X, Y, Z (coordinates), E (extrusion), and F (speed) refine actions.

4. Basic G-code Commands (CNC Machining)

1) G0 – Rapid Positioning
   Move the tool rapidly to a position (no cutting):
   G0 X10 Y20 Z5
   Moves to (X=10, Y=20, Z=5) at maximum speed.
2) G1 – Linear Interpolation (Cutting Motion)
   Move in a straight line at a specified feed rate:
   G1 X30 Y40 Z-2 F100
   Cuts to (X=30, Y=40, Z=-2) at 100 mm/min feed rate.
3) G2/G3 – Clockwise/Counterclockwise Arc
   Cut an arc with a defined radius:
   G2 X50 Y50 I10 J0 F200
   Creates a clockwise arc from the current position to (50,50) with a radius offset (I=10, J=0).
4) G20/G21 – Units
   Set units to inches (G20) or millimeters (G21):
   G21
   Switches to millimeter mode.
5) G28 – Return to Home Position
   G28 X0 Y0 Z0
   Returns the machine to its home position.
6) G90/G91 – Positioning Modes
   G90: Absolute positioning (coordinates are relative to origin).
   G91: Incremental positioning (coordinates are relative to current position)

5. Why is G-Code Important?

1) **Universal Compatibility:** Most 3D printers use G-Code, ensuring cross-platform workflows.

2) **Customization:** Advanced users tweak G-Code to fix slicing errors, optimize print times, or enable unique techniques (e.g., non-planar printing).
3) **Troubleshooting:** Reading G-Code helps diagnose issues like layer shifts, overheating, or extrusion problems.

6. The Emerging Threat of G-Code Attack

1) Modifying G-Code to Introduce Print Defects
2) Layer Manipulation: Attackers can alter layer heights or infill percentages to create weak spots in the object.
3) Extrusion Changes: By modifying extrusion rates, they can cause under-extrusion (weak prints) or over-extrusion (clogging and defects).
4) Temperature Manipulation: Adjusting extruder or bed temperatures can result in poor adhesion or overheating, leading to failure.
5) Uncontrolled Movements: Attackers can add G0 or G1 commands to move the print head erratically, causing print failures or even hardware damage.
6) Spurious Heating Commands: By modifying M104 (extruder temp) or M140 (bed temp), they can overheat components, potentially leading to fire hazards.
7) E-Stop Injection: Adding M112 (Emergency Stop) can shut down the printer unexpectedly.

7. Case Studies

7.1 Cyber-Physical Attack on 3D-Printed Drone Components (2016)

Researchers at the University of California, Irvine demonstrated how malicious modifications in G-code could compromise structural integrity in critical 3D-printed components.

1) Attack Method
- Researchers modified the G-code of a drone propeller by introducing microscopic voids within the layers.
- The attack was subtle enough to avoid immediate detection but weakened the part significantly.

2) Impact
- The drone propeller failed mid-flight, causing the drone to crash.
- This attack highlighted the risks of supply chain tampering, where attackers modify files before printing.

7.2 'SABOT' Attack on 3D Printing in Military Applications (2019)

A research team from Ben-Gurion University, Israel, conducted a proof-of-concept attack called SABOT (Sabotage via Additive Manufacturing).

1) Attack Method
- The team developed malware that infected a network-connected 3D printer used in military applications.
- The malware modified the G-code before printing, introducing undetectable internal defects in mission-critical parts.

2) Impact
- The manipulated parts passed initial quality inspections but failed under operational stress.
- This research demonstrated how cyber-attacks on 3D printing could be a national security threat.

3) Fire Hazard Attack via G-Code Modification (2021)
Security researchers demonstrated how overriding temperature settings in G-code could cause thermal runaway and potential fires in 3D printers.

a) Attack Method
- The attacker modified the G-code to:
- Disable thermal safety limits (M104/M140 commands).
- Set extruder and bed temperature beyond safe operating levels (e.g., 300°C on non-metal hotends).

b) Impact
- The printer overheated, causing component failure and potential fire risks.
- This type of attack is particularly dangerous for network-connected 3D printers in industrial or home environments.

4) Intellectual Property Theft via G-Code Extraction (2022)
In a case involving corporate espionage, attackers extracted proprietary designs from G-code files stored on cloud-based 3D printing services.

a) Attack Method
- Attackers intercepted G-code being sent to cloud-based 3D printers.
- By reverse-engineering the G-code, they reconstructed original CAD designs, bypassing IP protection measures.

b) Impact
- Companies lost trade secrets and proprietary designs to competitors.
- This case raised concerns about data security in cloud-based additive manufacturing.

8. Attack Execution: Methods of G-Code Injection

Once an attacker has access to a G-code file, they can introduce various types of malicious modifications:

1) Structural Sabotage
a) **Objective:** Introduce hidden weaknesses that cause premature part failure.
b) **Method:**
- Modify infill density (M220 S50 → reduces density by 50%).
- Weaken layer adhesion by altering extrusion temperatures (M104 S180 → lowers temp below material requirement).
- Skip critical layers (G92 E0 without extrusion moves).

c) **Outcome:**
- Parts may pass visual inspections but fail under stress.
- Dangerous in aerospace, automotive, and medical implants.

2) Thermal Overload and Material Degradation

a) Objective: Destroy printer components or compromise material properties.

b) Method:

- Increase extruder temperature beyond safe limits (M104 S350 for PLA).
- Modify cooling fan settings (M106 S0 disables cooling, leading to warping).
- Change bed adhesion settings to cause print detachment (M140 S1000).

c) Outcome:

- Can physically damage the printer (e.g., overheating can burn components).
- Objects become brittle or warp, reducing reliability.

3) Toolpath Manipulation for Hidden Defects

a) Objective: Introduce invisible micro-defects to compromise integrity.

b) Method:

- Shift layer alignment slightly (G1 X10.2 Y10.2 instead of G1 X10 Y10).
- Remove critical support structures (M98 P"disable_supports.gcode").
- Adjust layer height inconsistencies (G1 Z+0.02 on every 5th layer).

c) Outcome:

- Objects may fail weeks or months later, making attribution difficult.
- Particularly effective against high-performance mechanical parts.

4) Malware Execution via G-Code Commands

a) Objective: Use G-code commands to execute unauthorized actions on a networked system.

b) Method:

- Inject command display hacks (M117 "Update Required: Visit X.X.X.X" to trick users into downloading malware).
- Enable unauthorized remote access (M530 Enable_Remote_Control).
- Create infinite print loops (G0 X0 Y0 without stopping).

c) Outcome:

- Disruptions: Continuous printing wastes materials and disrupts production.

9. Mitigations

1) Input Validation & Sanitization

- Use parsers to block or flag unsafe commands (e.g., M104 S300 for overheating, G0 X99999 for out-of-bounds movements).
- Restrict execution of non-essential commands (e.g., firmware update codes like M997).

2) Trusted File Sources

- Only use G-Code from verified repositories or trusted slicer software.
- Avoid pre-sliced files from unvetted third parties (e.g., hobbyist forums).

3) Firmware Updates & Code Signing

- Regularly patch firmware to fix vulnerabilities (e.g., Marlin, RepRap).
- Enforce code signing for firmware updates to prevent malicious flashes.

4) Runtime Safety Checks

- Enable built-in firmware safeguards (e.g., thermal runaway protection, motion boundaries).
- Lock critical settings (e.g., max temperature, axis limits) to prevent override via G-Code.

5) Sandboxed Execution

- Run G-Code interpreters in isolated environments to limit system access.

6) Network Segmentation

- Isolate CNC/3D printer networks from enterprise IT systems to limit lateral movement.
- Use firewalls to block unnecessary inbound/outbound traffic.

7) Physical Port Security

- Disable unused USB/SD card ports to prevent local G-Code injection.

8) Monitoring & Anomaly Detection

- Monitor for abnormal machine behavior (e.g., rapid temperature spikes, axis collisions).
- Log G-Code execution and flag suspicious patterns (e.g., loops, M112 emergency stops).

9) User Training & Awareness

- Train users to recognize risks of untrusted G-Code (e.g., phishing via "free" 3D models).
- Promote use of slicer software from official sources (e.g., Cura, PrusaSlicer).
- Developers should follow secure coding standards for slicers and firmware (e.g., input validation, memory-safe languages).

10) Industry Collaboration & Standards

- Follow guidelines like NIST SP 1800-30 (cybersecurity for additive manufacturing) or ISO/ASTM 52943.
- Encourage coordinated reporting of firmware/slicer vulnerabilities.

10. Conclusion

G-code injection attacks represent a serious and evolving threat in the field of cyber-physical security. As 3D printing becomes a core technology for industrial, defense, and medical applications, adversaries will increasingly target the weak security mechanisms in additive manufacturing workflows.

By implementing strong encryption, anomaly detection, and secure G-code handling, organizations can mitigate these threats and ensure the trustworthiness of 3D-printed components.

Securing G-code today means securing the future of additive manufacturing.

References

[1] **Sturm, Y., Formby, D., & Krishna, A.** (2018). *Dr0wned – Cyber-Physical Attack with Additive Manufacturing.*
[2] **Nahar, P., Xu, X., Zhou, C., & Qu, G.** (2020). *S3D: A Tool to Detect Sabotage Attacks on 3D Printers.*
[3] **Trend Micro Research.** (2018). *3D Printing Security: Risks and Opportunities.*
[4] **ISO/ASTM 52943:2023.** *Additive manufacturing — Qualification principles — Security aspects.*
[5] **University of Cambridge. (2016**). *3D Printing and Intellectual Property: Security Challenges.*