dr0wned – Cyber-Physical Attack with Additive Manufacturing

Sofia Belikovetsky
Ben-Gurion University of the Negev

Mark Yampolskiy
University of South Alabama

Jinghui Toh
Singapore University of Technology and Design

Jacob Gatlin
University of South Alabama

Yuval Elovici
Ben-Gurion University of the Negev, Singapore University of Technology and Design

Abstract

Additive Manufacturing (AM, or 3D printing) is an emerging manufacturing technology with far-reaching implications. AM is increasingly used to produce functional parts, including components for safety-critical systems. However, AM's unique capabilities and dependence on computerization raise a concern that an AM generated part could be sabotaged by a cyber-physical attack.

In this paper, we demonstrate the validity of this concern by presenting a novel attack: reducing the fatigue life of a functional part. We develop a sabotage attack against a specific 3D-printed quadcopter propeller, causing its mid-flight failure, ultimately leading to the quadcopter's fall and destruction. The study described in this paper presents the very first full chain of attack against AM. We present all stages of the attack, beginning with a cyber-attack aimed at compromising a manufacturing environment and ending with the destruction of the target system that employs this part. Among major scientific contributions of this paper are a new category of a sabotage attack (accelerated fatigue), a novel systematic approach to identify options for such attack involving AM, and a demonstration of an empiric approach for the development and validation of an AM specific malicious manipulation.

We further demonstrate how the proposed sabotage attack can be integrated in a worm, thus enabling a wide-scale attack targeting either specific or similar enough digital design files of functional parts.

1 Introduction

Additive manufacturing (AM), often called *3D printing*, is a manufacturing method that creates objects by fusing layers of material. Compared to traditional subtractive manufacturing, which uses cutting tools to reduce a block of source material to the desired shape and size, AM has numerous socioeconomic, environmental, and technical advantages. These include shorter design-to-product time, just-in-time and on-demand manufacturing in proximity to assembly lines, reduction of source material waste, and especially the ability to produce functional parts with complex internal structure and application area optimized physical properties. A recent example of AM adoption, the U.S. Federal Aviation Administration (FAA) has certified the GE Aviation's 3D-printed fuel nozzles for the next-generation LEAP jet engine [27]. According to the Wohlers report [35], in 2016, the AM industry accounted for $6.063 billion of revenue, with 33.8% of all AM-generated objects used as functional parts. Due to the computerization involved in AM and the differences in the production environment compared to traditional manufacturing, several researchers have raised concerns regarding its security, including intellectual property violation [20, 37, 30, 22, 15, 4, 8] and sabotage [28, 38, 41, 31, 42, 43].

In this paper, we focus on the latter – presenting a novel attack that reduces the fatigue life of an AM generated functional part, thus causing its premature failure. To the best of our knowledge, such an attack has not yet been considered. Furthermore, while prior work has addressed various isolated aspects of sabotage attacks, no one has provided a holistic view and demonstrated a complete chain of attack. We present all stages of a sabotage attack, beginning with a cyber-attack aimed at compromising a manufacturing environment and ending with the destruction of the target system that employs this part.

The remainder of the paper is structured as follows. We discuss the related work in Section 2 and the considered attack scenario in Section 3. We develop a sabotage attack for this scenario and present an experimental verification of the attack in Sections 4 and 5, respectively. Then, in Section 6, we suggest a way to generalize the attack to enable a fully automated wide-scale sabotage on AM printed objects. In Section 7, we discuss the pre-

sented attack and outline approaches to mitigating similar attacks. We conclude this paper with a short overview of our findings.

The appendices provide additional theoretical background to support the analysis of the proposed attack. Appendix A provides an overview of a typical AM workflow and Appendix B contains a systematic approach for identifying possible attack chains which enable a controllable sabotage of a 3D printed functional part.

2 Related Work

To the best of our knowledge, the first experimental proof that a desktop 3D printer could be compromised was presented in 2013 at the XCon2013 conference by Xiao Zi Hang (Claud Xiao). According to his keynote presentation [36], an attack can modify "printing results," including the size of the model, position of components, integrability of components, etc.

Several publications analyze the possibility of compromising 3D printers. In [31], the authors analyzed the manufacturing process chain and found several attack vectors that can be easily exploited. They focused primarily on the aspects related to networks and communication. They examined the lack of integrity checks, particularly at the stage of receiving the design (common mechanisms that are not secure include email and USB drives), the lack of physical security on machining tools, the exposure to common network attacks, and the difficulty of relying on existing quality control processes. A recent publication [24] analyzed open-source software that is commonly used with desktop 3D printers: *Marlin* firmware, and three GUI applications that run on PCs and communicate with the 3D printer via G-code, *Cura 3D*, *ReplicatorG*, and *Repetier-Host*. In each of these programs, static analysis of the source code and dynamic analysis of the communication protocol between the 3D printer and the computer reveal numerous vulnerabilities that can be exploited.

Other publications dealing with this topic can be grouped based on the two main security threat categories associated with AM: intellectual property (IP) violation and sabotage of AM.

IP considerations will be covered briefly, as they are tangential to a sabotage attack. IP violation in AM has been considered from the legal perspective [20, 30, 8], and as the goal of an attack [15]. Several authors have addressed means of protecting AM-produced IP, including [22] and [13]. In [37], the scope of IP in AM is expanded to include part properties and AM process parameters. We are not aware of any other publications addressing IP in AM.

AM sabotage attacks have been addressed in several peer-reviewed publications. In [42], the authors propose a framework for the analysis of attacks involving AM and then discuss how certain categories of attacks can generate effects comparable with those produced by weapons (*e*.g., kinetic damage). Further, the authors argue that the targets of such an attack can be 3D manufactured objects, AM equipment, or the environment. In [41], based on an extensive survey of AM-related material science literature, the authors identified manufacturing parameters that can have a negative impact on a manufactured part's quality. The discussion focuses on AM with metals and alloys and covers a variety of AM processes, including powder bed fusion, direct energy deposition, and sheet lamination. The identified parameters include, but are not limited to, build direction, scanning strategy, heat source energy, etc.

For 3D plastic printers, several publications provide experimental proof that various manipulations can reduce the part's quality. In [28], the authors developed malware to alter the STL file defining the 3D object geometry by introducing voids (*i*.e., internal cavities) into the design. A similar approach is presented in [43]. The authors investigated two types of manufacturing modifications that impact the tensile strength: the insertion of sub-millimeter scale defects in the interior of 3D printed parts, and the modification of the orientation of a part during printing. As opposed to [28], in [43] defects are introduced by replacing the main material with a contaminant.

In [25], the authors discussed and demonstrated an attack against 3D printer firmware, where the amount of extruded source material was modified in order to compromise the printed object. Pope et al., 2016 [26], identified an indirect attack that can potentially influence the quality of the printed part through the modification of network command timing and energy supply interruptions.

3 Considered Attack Scenario

In this section we outline the considered attack scenario and the equipment used in experiments.

3.1 Victim and Adversary

We consider a realistic scenario involving a home user of a desktop 3D printer (see Figure 1) producing replacement propellers for his quadcopter UAV. The printer owner has procured a blueprint from a 3D object designer, and the 3D printer is controlled by a personal computer that sends the G-code commands via a USB connection[1].

[1]This scenario contains major elements characteristic to a general AM workflow (outlined in Appendix A).

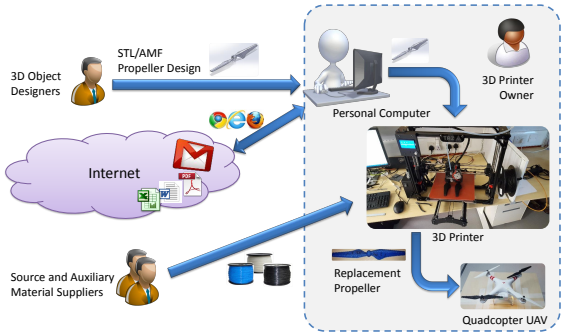

Figure 1: Scenario considered in experimental evaluation

In our study, we make several assumptions. First, we assume the 3D printer owner does not keep his software up-to-date, behavior which is representative of the majority of private users. Second, the printer owner also uses this PC to surf in the Internet, read emails, download documents, play games, etc.

In the considered scenario, an adversary wants to destroy the victim's drone. The attacker can achieve this by sabotaging the victim's 3D printed replacement propeller so that it fails at high altitude, resulting in damage to the drone. This means that the attack must pass basic visual inspections and manual mechanical tests, and not affect the propeller's integration into the drone, to ensure that the user will install it without suspicion. The defect must also be **time-delayed**, since the victim may choose to perform a basic functional test, and must fly the drone long enough for it to reach a sufficient height.

We assume that the adversary has basic to average hacking skills, and moderate AM proficiency. In addition, the adversary is capable of procuring the same desktop 3D printer and drone as the targeted user, or can build a similar test environment, in order to prepare and test the sabotage attack.

3.2 Experimental Environment

We have implemented the scenario outlined above using equipment that is reasonable for a private household (see summary in Table 1). The DJI Phantom 2 Vision+ is a four kilogram (8.8 pounds) quadcopter UAV that can fly for up to 25 minutes. The drone can be purchased for about $500.

For a 3D printer, we have selected the LulzBot Taz 5. Lulzbot uses Fused Deposition Modeling (FDM), a technology characteristic of most desktop 3D printers currently available. It can be purchased for about $1,250, placing it at the lower end of the price range for desktop 3D printers. The 3D printer operates using *Marlin* firmware; its counterpart, installed on the controller PC, is *Cura*. Both are widely used and recommended by several desktop 3D printer OEMs. The communication between the controller PC and 3D printer is established via a direct USB connection.

DRONE	DJI Phantom 2 Vision+
3D PRINTER	Lulzbot Mini TAZ 5
FIRMWARE	Marlin 2015Q2
CONTROLLER PC	Intel Xeon CPU, 32GB RAM, Windows 7
SOFTWARE	SOLIDWORKS, Cura 19.12

Table 1: Experimental Environment

The propeller is printed using Acrylonitrile Butadiene Styrene (ABS), a common material for printers of this type. ABS is durable, strong, flexible, shock absorbent, and heat resistant: a reasonable choice for producing lightweight functional parts. The latter is especially important to withstand the heat generated by the quadcopter's motors.

4 Attack Preparation

In this section we present the design and evaluation of our fatigue-based sabotage of a 3D printed CPS component. To the best of our knowledge, this is the first study that shows the entire chain of attack, beginning with infiltration and leading – over multiple stages – to the physical destruction of the targeted CPS. Furthermore, this is the first study that proposes a fatigue attack. The study also experimentally proves the feasibility of such attack. A video demonstrating the final stage of the attack is available on *YouTube* [1].

4.1 Attack Chain Selection

A general discussion about possible attack chains is presented in appendix B. In the current section we apply those concepts to identify an attack chain for our scenario (our attack path selection is presented in Figure 2).

4.1.1 Attack Target

In this scenario, the goal is for the propeller to break after a relatively brief operational time. This means that the sabotage of the propeller should introduce a defect leading to the mechanical stress concentration resulting in the rapid development of material fatigue in the propeller. It must, however, first survive a brief functional test and a reasonable amount of flight time.

4.1.2 Manipulations

Our assumptions about the adversary's skill level rule out manufacturing process manipulation; the manipulation must be selective and direct to achieve the goal.

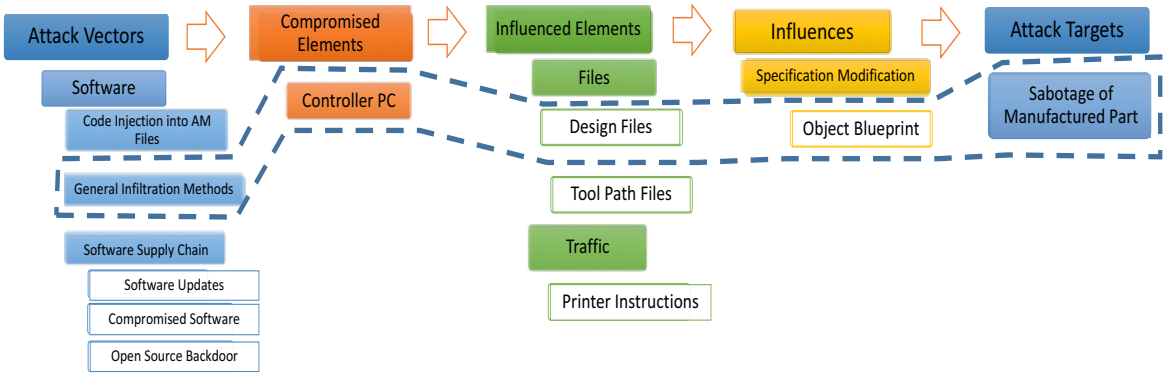

Figure 2: Case study attack chain

After consideration of the possible manipulations, we believe that the effects mentioned above can be achieved by modifying the object's blueprints, specifically by inserting an internal void, a type of sabotage that is unique to AM. We determined the exact specifications of this void by performing an experiment which is described in Section 4.2.

While the selection of the manipulation category is relatively straightforward, the exact definition of the modification is not. Even though the adversary has moderate AM proficiency, it is extremely difficult to calculate the impact of a defect on mechanical strength and fatigue, although these factors have been intensively studied and are well understood in materials science. Therefore, we assume the adversary must experiment with various defects before identifying a satisfactory choice.

4.1.3 Compromised Element

Having decided to introduce a defect into the object specification, the adversary must now target an element of the AM workflow to compromise. The object specification, in various representations, moves between different devices in the workflow. It originates with the 3D object designer and is transferred to the controller PC as an STL file. From the controller PC it is transmitted to the 3D printer as a sequence of G-code commands. In the 3D printer itself, the design may have a new, final representation. Any element along this path, if compromised, can modify the object specification.

In our scenario we aim to compromise the AM environment via the Internet; the controller PC, the element that is used for both AM and surfing the Internet, is the logical element to choose. At this stage, the specification is in an STL file.

4.1.4 Attack Vectors

Personal computers are frequent targets of attacks. We focused on the most common tools and behaviors of individual users and chose a phishing attack and a ZIP file format vulnerability in the popular WinRAR software. With well-considered social engineering, this combination could be an effective form of attack in many AM environments. It must be noted, that this cyber-attack is performed in order to gain access to the user's PC, in order to perform sabotage of the propeller, which is the main focus of our study; the infiltration method selected for demonstration purposes and is not based on novel material.

4.2 Defining Manipulation

Experiments are required to verify that the manipulation leads to the sabotage of the propeller. We assume that the adversary can obtain the resources to perform some experiments, using equipment either identical or close to identical to the victim's. The adversary has full knowledge of the printer and software in use, after gaining access to the victim's PC. This assumption is realistic: an adversary, after compromising a PC, can perform reconnaissance of the PC, and its installed software, stored files, and peripheral devices like a 3D printer.

Figure 3 shows the original propeller design, as modeled in the SOLIDWORKS software. The modifications will also be introduced using SOLIDWORKS.

4.2.1 Goals of Defect

As mentioned before, our goal for the attack is to cause the mid-flight destruction of the printed propeller, with a variety of secondary considerations. The defect should be undetectable by basic visual inspection and difficult to

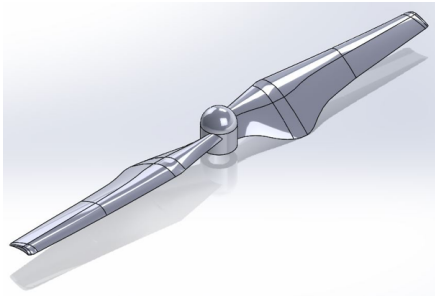

Figure 3: Original Propeller Designed using SOLIDWORKS

detect under basic manual mechanical tests. In addition, we wish the object to break after short operation time under normal conditions (*e.g.,* should be able to withstand a number of rapid ascents). The benefit of this to an attacker is that the part may pass a basic functional test, and then be deployed in normal operation. This category of cyber-physical attack, degradation fatigue time, has not been considered in the research literature before.

4.2.2 Location of Defect

Propellers convert the engine's torque force into thrust force. The original propeller design is engineered to withstand the thrust force that acts on it when the motors are spinning at the maximum supported revolution per minute (RPM). The lift force causes the propeller to bend upwards during flight, thus placing stress on the propeller.

The calculation of thrust force and stress distribution are complex topics[2] that, we assume, exceed the knowledge of an average adversary. Nevertheless, even with basic physics knowledge it is obvious that the most force is applied at the joint connecting the blades to the cap of the propeller (see Figure 4). This joint appears to be a good location to insert a defect in the design. Furthermore, if the propeller breaks at this joint, the loss of thrust will have the greatest impact compared to all other places where a defect can be introduced.

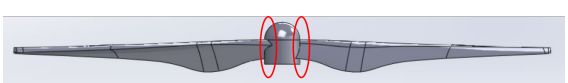

Figure 4: Red circles highlight location selected for the defect

[2]The thrust produced by a propeller depends on factors like its shape and diameter, on the motor's RPM, and on the density of the air. Stress distribution depends on the object's geometry and material properties.

4.2.3 Iteration through Manipulations

We began investigating possible manipulations by drilling holes between the blades and cap of a propeller printed using the unaltered design. A counterintuitive outcome was that doing this did not greatly affect the propeller's ability to operate at maximum speed without breaking. This was due to the circular cavity's ability to evenly distribute the stress.

Next, we modified the design file by inserting tiny gaps into the joint, between the blades and the cap. As a result, the design file contained three separate parts that were placed adjacently. The leading idea for this manipulation was that if the gaps were small enough, the three parts will still be attached during the 3D printing process and an artificial breaking point will be created. This manipulation is unique for AM, since we are creating a breaking point without changing the design of the propeller, but by slightly changing the distribution of the material.

We iterated using various gap sizes and verified their impacts empirically. We found 0.1mm length gaps to be optimal for the considered scenario and time frame, since larger gaps degraded the mechanical strength below operational conditions, and thus caused the propeller to break within seconds of normal activity. To enforce the created breakage point, we have added rectangular structures for connecting the separated parts, that are shown in Figures 5 and 6.

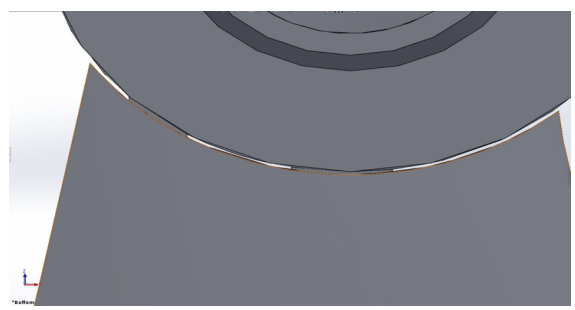

Figure 5: A sketch of the gap with the 2 support structures

This manipulation exploits a weakness in the AM process. The slicing software tries to slice the design "as is" without taking into consideration adjacent or overlapping parts. Thus, by cutting the design to parts and moving them slightly, we completely change the resulting tool path. In the original design, the created tool path printed the external wall while continuously moving from one side of the propeller to the other. However, once we broke the design into three parts, the tool path constructs the external wall of each part separately. This shift weakens the strength of the material at that point and accelerates breakage.

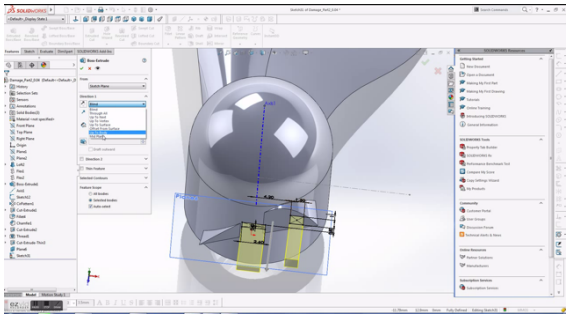

Figure 6: Propeller design with introduced defects

Through this iterative process, we identified a modification that yielded satisfactory results, *i*.e., **the propeller broke after operating at maximum thrust for several minutes**. We tuned the defect for this short time for the purpose of the experimental demonstration.

It should be noted that, because the defect is introduced at a joining point of the blades and the cap, the change in the printing pattern at this point will remain unnoticed by a simple visual inspection of the printed propeller.

Figure 7 shows two propellers side by side. The propeller on the top was printed using the original design, while the propeller on the bottom was based on our maliciously altered design. The net weight change due to the insertion of the void was less than 3%.

Figure 7: Two printed propellers side by side. The top propeller is *benign* and the bottom one has been *sabotaged*

4.2.4 Evaluation of Effects

For each altered design, we printed a propeller. We evaluated the effects of the introduced defects empirically, in a laboratory setting. During each experiment, we installed two 3D printed propellers, one unmodified and the other sabotaged.

In order to create a controlled, reproducible testing environment and to measure the effects of our manipulations, we conducted the test in an enclosed room and attached the drone to the table. We rapidly increased thrust to reach the maximum RPM, a normal operational condition under which the propeller is exposed to the maximum stress. During this experimental process, we measured two factors: the time the propeller broke and the RPM at that time.

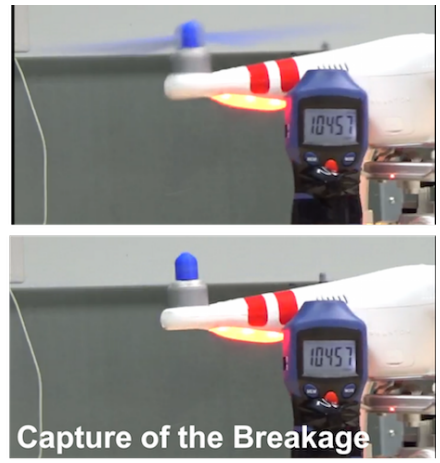

Figure 8: The damaged propeller just after it breaks in the lab

During our tests we verified that the propeller printed using the original design could operate at more than 15,000 RPM for an extended period of time without showing any signs of material fatigue. However, propellers printed with various defects introduced in the design either could not sustain operational conditions at all or were not able to sustain these conditions for an extended period of time. Figure 8 captures the moment a sabotaged propeller broke in one of the conducted experiments; in this particular experiment, the breakage point occurred at 10,457 RPM.

5 Attack Execution

In this section we describe the actual attack execution. The attack was performed according to the plan developed in Section 4.

5.1 Sabotage Attack

We executed the sabotage attack in three steps. First, we compromised the victim's controller PC. Second, we downloaded the original design file and manipulated it (as described in Section 4.2). Third, we changed the design file on the victim's PC.

Figure 9: The drone with three normal propellers and one damaged propeller

5.1.1 Compromising Controller PC

In order to infiltrate the system, we used a patched WinRAR vulnerability [2] that spoofs the file name and extension of the archived document. We created a malicious EXE file using the Metasploit framework that triggers an exploit. Using the WinRAR vulnerability, we changed the name and extension of the executable file to look like an innocent PDF file.

The victim received an email enticing him to download a .Zip file from Dropbox and double-click on the PDF file inside. Once the victim clicks on the file, a reverse shell is opened in the background and the attacker can take control of the system.

5.1.2 Manipulating Design File

The next step was to find design files that the attacker can manipulate. The attacker searches for .STL files and downloads them for further investigation. Once the files are in the attackers' possession, the adversary can modify the design in any way.

In our experiment, we modified the design using the SOLIDWORKS software, tested the effect of the manipulation experimentally (as described in Section 4.2).

5.1.3 Replacing Design File

After the modification has been developed, we used reverse shell to replace the original file with the maliciously altered one.

Figure 10: Drone in flight with the damaged propeller

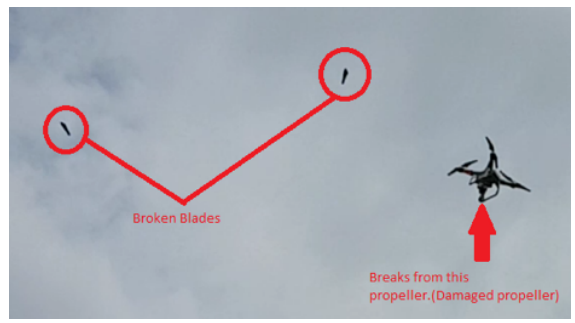

Figure 11: The damaged propeller breaks mid-flight

5.2 Field Trials

The first step of the field trials tested the original printed propellers in flight conditions. We attached four propellers that were all printed from the unaltered design. The drone was able to take off and fly normally for more than five minutes.

We then replaced a single propeller with a sabotaged version. Figure 9 shows the drone with the sabotaged propeller (pre-flight).

During the field trials, the propeller was able to withstand a flight test of 1 minute and 43 seconds. (Figure 10). During this time, the drone performed three cycles of rapid low-to-high altitude changes. During these initial cycles, the sabotaged propeller performed normally. However, during the fourth ascent the propeller broke apart.

When the propeller blades broke apart during the flight (see Figure 11), the drone was at a high altitude. Consequently, it fell from a considerable height and shattered. The fall caused damage to one of the motors; in addition,the onboard camera was completely destroyed, and the external casing of the drone cracked.

5.3 Evaluation of Trials

We achieved all three major goals stated in Section 4.2.1. The first goal, mid-flight destruction of the sabotaged propeller, was fully achieved. The second, undetectability under visual inspection and measurement, was partially achieved. As seen in Figure 7, the void is not visible; the change in the part's weight due to the void is less than 3%, which we consider tolerable. The third goal, to verify a novel attack that causes fatigue over a sustained period of operation, rather than failure based on some specified amount of mechanical stress, was met in the field trials. As stated in Section 5.2, equipped with the sabotaged propeller, the drone was able to operate normally for one minute and 43 seconds before failing. We have chosen the propeller modification with the short time for fatigue development solely for a shorter turnaround time during the field trial. Furthermore, this period of operation is sufficient for the chosen target, an experimental proof of a fatigue-based sabotage attack. A more sophisticated attack would produce a greater time delay before failure. This would enable the attack to remain undetected through a full operational range test, such as our own.

6 Scaling Up: G-Code Sabotage Worm

The presented attack can be characterized as a highly targeted because it only affects a specific user of a desktop 3D printer. Further, the effects of a sabotage can be well controlled, based on the modification developed for a specific 3D printed part and its operational conditions. This can be considered as a limiting factor by some adversaries. To scale up the attack and potentially affect multiple users, we have developed a script that introduces the described sabotage attack into G-Code files. In order to be effective, it can be integrated into a computer worm, that would manipulate G-Code files it reaches in a way that would be very difficult to detect.

G-Code files contain the instruction set for the 3D printer to execute; they represent the geometry of the design and the specific printing information like orientation, speed, amount of material, etc. Even thought STL files represent the geometry of the object, the rest of the parameters are not saved in a structured manner. They are only integrated into the G-Code file without the ability of simple retrieval (not easily reversible process). Thus, once the printing parameters were defined, they are integrated into the G-code data and present only there. This combination of dependability on storing printing parameters and the difficulty in retrieving these parameters makes the G-Code files vulnerable and a good target for a generic AM sabotage attack.

We have implemented a script that searches for a specific geometry and inserts customized gaps in it. In order to determine if the geometry is suitable, the script checks the outline of every printed layer by searching for the coordinates of the extruder's movements that build the outer frame. Then, the script iterates through the layers of the design to find the weak spots. In our case, we are looking for narrow joint points that are significantly smaller than rest of the body. In figure 12, we can see the outline of the middle layer of a benign propeller's G-Code file. The arrows represent the potential points for sabotage.

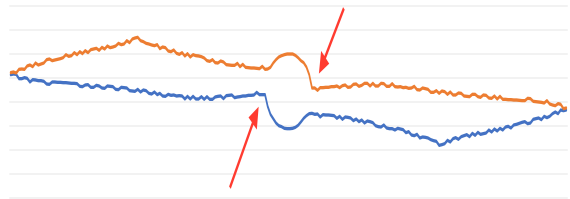

Figure 12: outline of the middle layer of the benign propeller G-Code file

If we want to target only a specific geometry, we can apply several heuristic rules to determine whether to modify the file. Those include proportions of the design, outer geometry, length of print, orientation and more. Based on those parameters, we can also adjust the size of the gaps and the support structures. The final step is determining the exact location of the gaps, their number and size, and modifying the G-Code to "skip over" those parts. Table 2 shows a comparison between the original move and the modified code. Since the second move prints over the desired gap location, it is split into two moves, "skipping" the marked location. The line that intersects the gap is commented out for demonstration by a ";" and will not be present in the resulting G-Code file.

Table 2: Insertion of gaps

Original Code	Split a printing move
G1 X103.265 Y153.205 E959.89129	G1 X103.265 Y153.205 E959.89129
G1 X107.531 Y155.147 E959.89508	**G1 X105.011 Y154.0 E959.89508**
	;G1 X106.0 Y154.450 E959.89508
	G1 X107.531 Y155.147 E959.89508

In order to increase the chances of carrying out this attack, the scripts needs to be integrated in a worm-typed malware, since the location of the G-Code files can not be known in advance and can reside on a server on the internal network. If implemented as a worm, it can have a variety of activation triggers, e.g., a specific date when an attack starts. Such a delay can ensure that numerous systems are infected, thus causing a simultaneous sabotage of numerous parts before the attack is detected. Furthermore, additional constraints can be added to make even

an automated attack more targeted. For instance, activation can be only enabled in a specified geographical regions, or vice versa exclude particular regions.

As of affected design files, we see at least three options. First, a highly targeted attack would affect exclusively parts for which a sabotage attack was developed and tested. An approach can be used that performs a global system search for design files, computes and compares files' hash sums to a specific target one, and replaces matching files through a developed sabotaged one. Obviously, the impact of an attack can be fairly well controlled, because of the dedicated and tested modification. At the same time, this attack will sabotage the least amount of manufactured parts, because neither all infected systems will contain the target design file nor it is guaranteed that it will be 3D printed in the time frame between worm is activated and a patch for this attack is developed.

Second, on the opposite site of the scale, an automated sabotage attack can insert (either predefined or random) defects at random places in all digital design files found in the system. While this attack will sabotage the largest amount of manufactured parts, the effect of such random insertion on a part's performance cannot be predicted at all.

Third, an attack can target design files that are similar to the one for which a modification was developed. Obviously, it will affect more system that in the first case, while providing somewhat similar impact on the sabotaged parts. This is the approach we have implemented.

7 Discussion

In this section, we first discuss the viability of the proposed attack and then outline mitigation strategies against it.

7.1 Viability of Attack

The viability of the proposed attack depends on several factors. Most importantly, ability to compromise AM environment, validity of the proposed approach to sabotage a part, and probability that such a sabotaged part will be identified.

The demonstrated attack can bypass standard security measures in a production environment. While there is a growing awareness of potential security issues and plans to employ cyber-security measures in industrial AM environment [5], these are not widely implemented yet.

Furthermore, since the manipulation does not interrupt the flow of the manufacturing and the compromise of the controller PC can be brief, it would be difficult for endpoint security or network security mechanisms to detect.

Notably, manipulation of the design file is possible across all AM technologies. This is not the case for other categories of manipulations, like altering the manufacturing parameters. For controlled buildup of fatigue, these manipulations should be designed under consideration of a part's geometry, source material, and mechanical stress it will be exposed to under normal operational conditions.

As of now, detectability of an attack mainly relies on ability to identify a sabotaged part after it is produced. There are significant differences between part testing, ranging from thorough testing of functional parts in industrial settings to no testing at all in consumer market. Generally, destructive testing is only used during the part design, whereas non-destructive testing (NDT) during its manufacturing. However, even NDT is very time and labor intensive, and requires expensive specialized equipment.

In order to save the time and costs associated with testing, it is common practice that in the production phase only a few samples in a batch are fully inspected. Furthermore, NDT methods used in traditional manufacturing are not fully effective for AM-produced parts and AM-specific NDT is not yet mature and can only detect relatively large defects.

7.2 Mitigation Approaches

Prevention and detection of sabotage attacks has to overcome several challenges outlined below.

A simple form of protection against the demonstrated attack would use a classical applied cryptography solution, such as a digitally signed cryptographic hash of a digital design file. While this method increases the difficulty of an attack, a compromised controller PC or 3D printer could bypass the verification.

In manufacturing environments, systems are often *air-gapped* to prevent direct exposure of the infrastructure to various attacks. While this significantly increases the difficulty of a system compromise, it is not impenetrable. Furthermore, for the 3D printing service, a connection to the Internet is required to transfer customers' blueprint files and supplementary documents, which can be used as attack vectors.

As discussed before, compromising an object does not require targeting the design files. The modification can be made to any representation of the design, at any stage. Therefore, the validity of the manufacturing process itself should be verified. If classical cyber-security solutions are implemented on 3D printer, they could become a subject of a compromise; more critical, they could impact 3D printer's timing, thus eventually degrading a manufactured part's quality.

A promising approach was shown in [10] where the

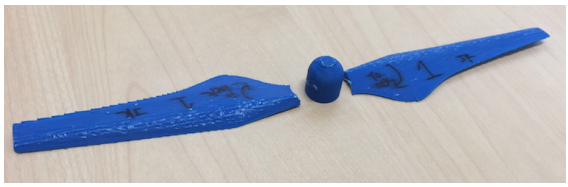

Figure 13: The broken sabotaged propeller

authors use acoustic emanations produced by a FDM 3D printer to reconstruct 3D model and compare it to the original STL file. This approach is both non-invasive and airgapped from the monitored system. Unfortunately, the researchers achieved only 77.45% accuracy in sabotage detection, what is insufficient for detection of an attack as presented in this paper.

8 Conclusion

Additive Manufacturing (AM) is emerging as a transformative manufacturing technology that is increasingly used for production of the functional parts. Due to the computerization of AM, several researchers have raised concern about its possible sabotage. Since if parts for safety-critical systems are sabotaged, such attacks can have far-reaching implications. In this paper, we demonstrate the validity of this concern.

To our best knowledge, this paper presents the first full chain of attack involving AM, beginning with a cyber-attack aimed at compromising a manufacturing environment and ending in the physical destruction of a cyber-physical system that employs a sabotaged part.

We proposed and experimentally verified a new category of a sabotage attack, an accelerated development of material fatigue. We applied the proposed approach on a realistic scenario, a desktop 3D printer used to manufacture propellers of a quadcopter UAV. During the experimental validation, the sabotaged propeller broke during the flight (see Figure 13). A video that summarizes the presented attack and shows the quadcopter's final flight is available on YouTube [1].

We also discussed several approaches to mitigate sabotage attacks. While some of these approaches are promising, their effectiveness is still insufficient to detecting the sabotage attack presented in this paper.

References

[1] dr0wned demo video on YouTube. "https://youtu.be/zUnSpT6jSys".

[2] WinRAR file extension spoofing vulnerability. "http://www.rarlab.com/vuln_zip_spoofing_4.20.html".

[3] 3MF Consortium. 3D Manufacturing Format Core Specification & Reference Guide. `http://3mf.io/wp-content/uploads/2016/03/3MFcoreSpec_1.1.pdf`, 2015.

[4] Al Faruque, M. A., Chhetri, S. R., Canedo, A., and Wan, J. Forensics of thermal side-channel in additive manufacturing systems. Tech. rep., 2016.

[5] America Makes & ANSI Additive Manufacturing Standardization Collaborative (AMSC). Standardization roadmap for additive manufacturing.

[6] Ariyapperuma, S., and Mitchell, C. J. Security vulnerabilities in DNS and DNSSEC. 335–342.

[7] ASTM International. Standard Specification for Additive Manufacturing File Format (AMF) Version 1.2, Active Standard ISO / ASTM52915-16, 2016.

[8] Brown, A., Yampolskiy, M., Gatlin, J., and Andel, T. R. Legal Aspects of Protecting Intellectual Property in Additive Manufacturing. Unpublished manuscript, 2016.

[9] Carrozza, G., Pietrantuono, R., and Russo, S. Defect analysis in mission-critical software systems: a detailed investigation. *Journal of Software: Evolution and Process 27*, 1 (2015), 22–49.

[10] Chhetri, S. R., Canedo, A., and Al Faruque, M. A. Kcad: kinetic cyber-attack detection method for cyber-physical additive manufacturing systems. In *Proceedings of the 35th International Conference on Computer-Aided Design* (2016), ACM, p. 74.

[11] Do, Q., Martini, B., and Choo, K.-K. R. A data exfiltration and remote exploitation attack on consumer 3d printers. *IEEE Transactions on Information Forensics and Security 11*, 10 (2016), 2174–2186.

[12] Electronic Industries Association and others. *Interchangeable Variable Block Data Format for Positioning, Contouring, and Contouring/Positioning Numerically Controlled Machines*. Electronic Industries Association, 1980.

[13] Fadhel, N. F., Crowder, R. M., and Wills, G. B. Provenance in the additive manufacturing process. *IFAC-PapersOnLine 48*, 3 (2015), 2345–2350.

[14] Falliere, N., Murchu, L., and Chien, E. W32. stuxnet dossier. *White paper, Symantec Corp., Security Response* (2011).

[15] Faruque, M. A., Chhetri, S. R., Canedo, A., and Wan, J. Acoustic side-channel attacks on additive manufacturing systems. In *Proceedings of the ACM/IEEE Internation Conference on Cyber-Physical Systems (ICCPS' 16)* (2016).

[16] Farzadi, A., Waran, V., Solati-Hashjin, M., Rahman, Z. A. A., Asadi, M., and Osman, N. A. A. Effect of layer printing delay on mechanical properties and dimensional accuracy of 3d printed porous prototypes in bone tissue engineering. *Ceramics International 41*, 7 (2015), 8320–8330.

[17] Frascati, J. *Effects of position, orientation, and infiltrating material on three dimensional printing models*. PhD thesis, University of Central Florida Orlando, Florida, 2007.

[18] Gibson, I., Rosen, D. W., Stucker, B., et al. *Additive manufacturing technologies*. Springer, 2010.

[19] Hiller, J. D., and Lipson, H. Stl 2.0: a proposal for a universal multi-material additive manufacturing file format. In *Proceedings of the Solid Freeform Fabrication Symposium* (2009), no. 1, Citeseer, pp. 266–278.

[20] Holbrook, T. R., and Osborn, L. Digital patent infringement in an era of 3d printing. *UC Davis Law Review, Forthcoming* (2014).

[21] Lipson, H. Amf tutorial: The basics (part 1). *3D Printing and Additive Manufacturing 1*, 2 (2014), 85–87.

[22] MACQ, B., ALFACE, P. R., AND MONTANOLA, M. Applicability of watermarking for intellectual property rights protection in a 3d printing scenario. In *Proceedings of the 20th International Conference on 3D Web Technology* (2015), ACM, pp. 89–95.

[23] MAUW, S., AND OOSTDIJK, M. Foundations of attack trees. In *International Conference on Information Security and Cryptology* (2005), Springer, pp. 186–198.

[24] MOORE, S., ARMSTRONG, P., MCDONALD, T., AND YAMPOLSKIY, M. Vulnerability analysis of desktop 3d printer software. In *Resilience Week (RWS), 2016* (2016), IEEE, pp. 46–51.

[25] MOORE, S. B., GLISSON, W. B., AND YAMPOLSKIY, M. Implications of malicious 3d printer firmware. In *Proceedings of the 50th Hawaii International Conference on System Sciences* (2017).

[26] POPE, G., AND YAMPOLSKIY, M. A Hazard Analysis Technique for Additive Manufacturing. In *Better Software East Conference* (2016).

[27] REPORTS, G. The faa cleared the first 3d printed part to fly in a commercial jet engine from ge. Tech. rep., 2015.

[28] STURM, L., WILLIAMS, C., CAMELIO, J., WHITE, J., AND PARKER, R. Cyber-physical vunerabilities in additive manufacturing systems. *Context 7* (2014), 8.

[29] TEHRANIPOOR, M., AND KOUSHANFAR, F. A survey of hardware trojan taxonomy and detection. *IEEE Design and Test of Computers 27*, 1 (2010), 10–25.

[30] TRAN, J. L. The law and 3d printing. *J. Marshall J. Computer & Info. L. 31* (2015), 505–657.

[31] TURNER, H., WHITE, J., CAMELIO, J. A., WILLIAMS, C., AMOS, B., AND PARKER, R. Bad parts: Are our manufacturing systems at risk of silent cyberattacks? *Security & Privacy, IEEE 13*, 3 (2015), 40–47.

[32] VAEZI, M., AND CHUA, C. K. Effects of layer thickness and binder saturation level parameters on 3d printing process. *The International Journal of Advanced Manufacturing Technology 53*, 1-4 (2011), 275–284.

[33] WHITEPAPER, E. ACAD/Medre.A.

[34] WOHLERS, T. Wohlers report, 2016.

[35] WOHLERS, T. *Wohlers Report 2017 3D Printing and Additive Manufacturing State of the Industry Annual Worldwide Progress Report*. Wohlers Associates, Inc., Fort Collins, Colorado, USA, 2017. www.wohlersassociates.com.

[36] XIAO ZI HANG (CLAUD XIAO). Security attack to 3d printing, 2013. Keynote at XCon2013.

[37] YAMPOLSKIY, M., ANDEL, T. R., MCDONALD, J. T., GLISSON, W. B., AND YASINSAC, A. Intellectual property protection in additive layer manufacturing: Requirements for secure outsourcing. In *Proceedings of the 4th Program Protection and Reverse Engineering Workshop* (2014), ACM, p. 7.

[38] YAMPOLSKIY, M., ANDEL, T. R., MCDONALD, J. T., GLISSON, W. B., AND YASINSAC, A. Towards Security of Additive Layer Manufacturing, 2014. WiP presented at The 30st Annual Computer Security Applications Conference (ACSAC) 2014.

[39] YAMPOLSKIY, M., HORVATH, P., KOUTSOUKOS, X. D., XUE, Y., AND SZTIPANOVITS, J. Taxonomy for description of cross-domain attacks on CPS. In *Proceedings of the 2nd ACM international conference on High confidence networked systems* (2013), ACM, pp. 135–142.

[40] YAMPOLSKIY, M., HORVÁTH, P., KOUTSOUKOS, X. D., XUE, Y., AND SZTIPANOVITS, J. A language for describing attacks on cyber-physical systems. *International Journal of Critical Infrastructure Protection 8* (2015), 40–52.

[41] YAMPOLSKIY, M., SCHUTZLE, L., VAIDYA, U., AND YASINSAC, A. Security challenges of additive manufacturing with metals and alloys. In *Critical Infrastructure Protection IX*. Springer, 2015, pp. 169–183.

[42] YAMPOLSKIY, M., SKJELLUM, A., KRETZSCHMAR, M., OVERFELT, R. A., SLOAN, K. R., AND YASINSAC, A. Using 3d printers as weapons. *International Journal of Critical Infrastructure Protection* (2016).

[43] ZELTMANN, S. E., GUPTA, N., TSOUTSOS, N. G., MANIATAKOS, M., RAJENDRAN, J., AND KARRI, R. Manufacturing and security challenges in 3d printing. *JOM* (2016), 1–10.

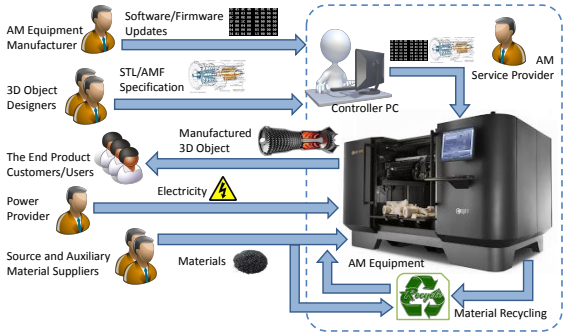

Figure 14: Additive Manufacturing workflow

A Additive Manufacturing Workflow

Figure 14 presents a high abstraction level workflow that is common in AM. This workflow represents a scenario where AM is offered as a service. In this case, multiple actors are involved in the AM process and provide/consume different services. For home use 3D printing, the workflow is reduced to a user (combining AM service provider, end customer, and possibly 3D object designer) and the various suppliers, and material recycling does not take place.

AM equipment is usually developed and provided by an original equipment manufacturer (OEM). The firmware and software updates (for AM equipment and the controller PC) that extend functionality and fix bugs are provided by the OEM or commercial software companies. For desktop 3D printers, open-source software developed by the 3D printing community is often used.

It is important to note that for equipment maintenance and repair, various mechanical, electrical, and electronic components (*e*.g., motors, filters, etc.) might be required. These items are sold by OEMs or third-party companies, and shipped via physical carriers.

The blueprint of a 3D object is provided either in STereoLithography (STL) [19], in Additive Manufacturing File (AMF) format [7, 21], or in a recently adopted 3D Manufacturing Format (3MF) [3], all of which represent the Computer-Aided Design (CAD) model of the 3D object to be manufactured. The figure depicts a scenario in which object blueprints (in STL/AMF files) are provided by external 3D object designers directly to the AM service provider. Another scenario is when the design is provided by the end product customer, who either designed the proposed object (common for enterprise customers) or acquired the design/blueprint from a designer (common for individual consumers).

At the AM service provider site, an STL/AMF file can either be directly transferred to a 3D printer (*e*.g., via computer network or USB stick), or interpreted by the controller PC. In the latter case, the controller PC sends the 3D printer either individual control commands (often in G-code [12], a language used in Computer-Aided Manufacturing, CAM, or a toolpath file containing a sequence of 3D printer-specific commands to be executed [28]. In both cases, the commands are generated by "slicing" software.

Depending on the AM process, the source material, and the part geometry, the production workflow can include several post-processing steps. These typically include removal of support structures used in the production of 3D objects with complex geometry. After all necessary production and post-production steps are accomplished, the manufactured 3D object is delivered to the customer via a physical carrier.

B Generic Attack Chain

Figure 15 outlines how attacks on or with AM can be performed [42]. A variety of attack vectors can be used to compromise one or more elements of the AM workflow. The compromised element(s), their roles in the workflow, and the degree to which an adversary can control these element(s) determine the kind of manipulations an adversary can perform. In conjunction with the type of AM equipment, source materials, and the application area of the manufactured part, these manipulations determine the potential effects. Only a subset of the resulting effects may intersect with the adversarial goals. In what follows, we refer to this intersection as the *achievable adversarial goals* or *attack targets*.

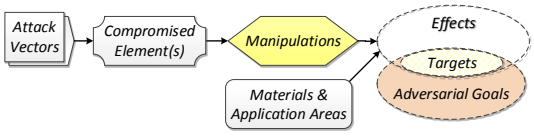

Figure 15: Attack on/with 3D printer (based on [42])

In this section we use the framework depicted in Figure 15 for the identification of possible attacks involving AM. Inspired by attack trees [23], we begin with the achievable adversarial goals (or attack targets), followed by a discussion of the manipulations that are needed to achieve the selected goals, and then we discuss which elements in the AM workflow can facilitate the manipulations listed, and address attack vectors that enable these elements to be compromised. The outcome of the analysis presented in this section is summarized in Figure 16.

It must be noted that the sets of goals, manipulations, compromised elements, and attack vectors are not wholly unique to AM. Some portions are generic to all cyber-physical systems (*e.g.*, the attack vectors), and some are

common to many CAM methods (*e.g.*, influenced elements, some compromised elements). These are included for completeness; AM shares the weaknesses of technologies with similar implementations and purposes.

B.1 Adversarial Goals

In this paper, we consider only the intentional sabotage of a 3D printed functional part.

Sabotage of Manufactured Part: Functional parts are typically designed to operate under specified conditions for an extended period of time. We can distinguish between two cases of sabotage. First, the part can be altered such that its normal operational range exceeds its (altered) mechanical strength, causing the part to break under normal conditions. Second, the part can be altered in a way that it is still operational but develops material fatigue faster, causing the part to wear out sooner than expected. The former attack category has already been discussed in the research literature [28, 41, 43]. To our best knowledge, the latter, which is the focus of this paper, has not been previously shown or discussed.

B.2 Manipulations

Sabotage can be achieved via various manipulations. In this paper, we only consider manipulations that can be executed in the cyber domain. As defined in [39, 40], manipulations can be classified by influenced elements and influences. Influenced elements describe the object that is manipulated by an attack and influences describe the modification that is done.

The research literature has identified two major categories of manipulations: modification of the object's specification [28, 41, 43] and manipulation of the manufacturing process [41]. We discuss the influences and the influenced elements that are involved in those manipulations. We further restrict our considerations to the operational phase of the manufacturing life cycle.

B.2.1 Influences

Object Specification Modification: The object's specification describes the object's geometry, orientation, and its material; the latter is only relevant for multi-material AM equipment. It should be noted that the object specification can have various representations, based on the "location" in the AM workflow (see Figure 14). It is commonly associated with the STL or AMF files, both of which are CAD formats, but it can also be represented by a toolpath file or as a series of individual G-code commands, etc.

Defects: Changing an object's exterior shape can affect a part's integrability in a system and be detected by visual inspection. Several researchers have also proposed the use of internal defects as a means of sabotage [28, 41, 43]. The negative impact of internal voids (*i*.e., cavities) and contaminant material defects on mechanical strength has been demonstrated experimentally in [28] and [43], respectively. Further, as noted in [41], this kind of attack will affect the part's weight and weight distribution – properties that can impact the performance of the system.

Orientation: It is well-known that anisotropy[3] in 3D printed parts is present in several AM processes [18, 17]. Based on this property, researchers have proposed changing the build direction as a means of sabotaging a manufactured part's mechanical properties [41]. An experimental proof of this attack was shown in [43].

Manufacturing Process Manipulation: AM defines the manufactured object's geometry, as well as its material. Various parameters of the AM manufacturing process[4] influence the microstructure of the created material, and thus its physical properties. This aspect has been investigated in material science from the quality assurance perspective [32, 16]. For metals and alloys, a security-centered qualitative analysis of manufacturing parameter manipulations was presented in [41]. Parameters like layer thickness, scanning strategy, heat source energy, etc. were identified as vulnerable to malicious manipulations.

In fused deposition modeling (FDM), an AM technology that is popular with desktop 3D printers (which we will use in the case study in Sections 4 and 5) parameters, such as nozzle temperature, print bed temperature, filament extrusion speed, distance between the extruder and the printed object, etc, can be manipulated. All of these can have an impact on interlayer bonding, thus impacting the part's mechanical properties. The effects of some of these manipulations have been shown in [25].

[3] Anisotropy means that properties vary in different directions; in the discussed case, mechanical properties like tensile strength are meant.

[4] These parameters and the impact of their manipulation vary greatly across different AM technologies.

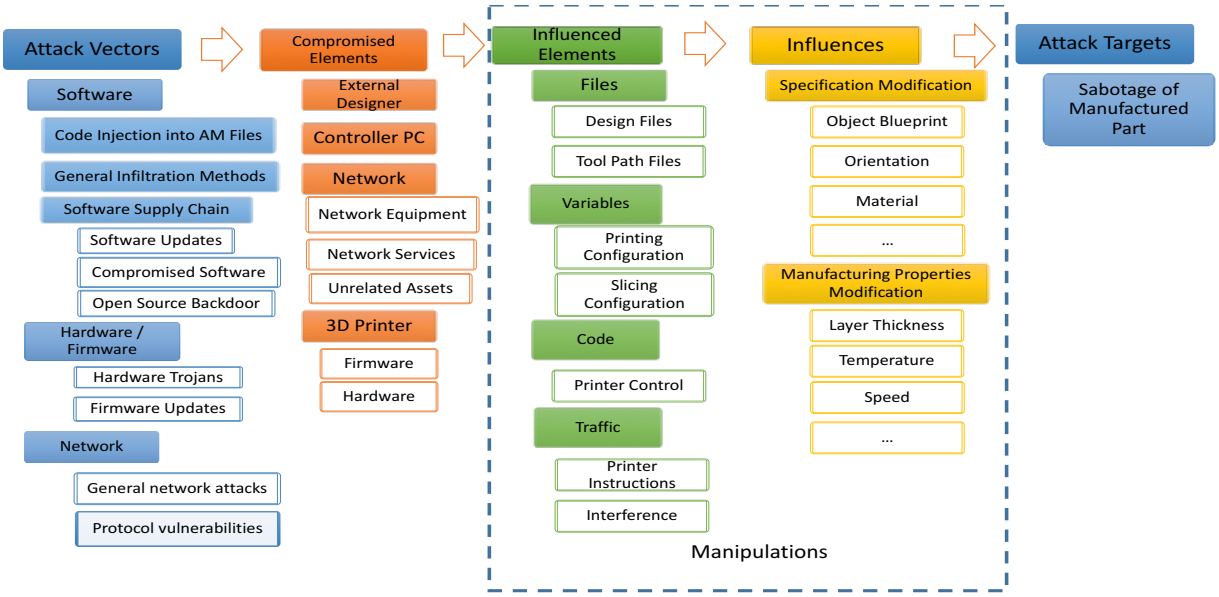

Figure 16: Attack flow

B.2.2 Influenced Elements

AM Files: STL/AMF files contain the object's specification, and any modification to these files will change the object produced. Printer instructions generated from these design files are compiled into a toolpath file. This file can be changed on the controller PC, during network communication, or on the 3D printer. The printer's instructions in the toolpath file combine the object's specification with the printing process parameters.

Configuration: Software configuration changes can impact the printed object. Crucial properties can be manipulated by slicing software, such as layer thickness, fill density, shell thickness, etc. Additionally, modifications to the printer configuration, including print bed temperature, nozzle travel speed, etc, can impact the manufacturing process.

Code: Compromising the software enables fine-grain control over the outcome. By altering the software that runs on the controller PC or the 3D printer firmware, the adversary can modify the object specification or the printing process.

Traffic: Modifying traffic from the controller PC to the printer in order to alter the timing or content transmissions can change the printed object. In [16], the authors show that even slight printing delays between each printed layer can affect the compressive strength, durability, and tangent modulus of the printed samples.

B.2.3 Characteristics of Manipulations

Individual manipulations can have different characteristics. We propose the following categorization:

Indiscriminate/Selective: Manipulations such as changing the object's orientation are indiscriminate, affecting the entire object. Similarly, configuration of the slicing tool will lead to changes that affect the processing of the entire design file.

Other manipulations can be selective, depending on the subject of manipulation and the compromised element. For example, layer thickness can be modified by changing a subset of G-code commands, impacting only selected layers during the manufacturing process.

Static/Dynamic: Another characteristic is whether manipulations are statically present or dynamically introduced. For instance, modifications of STL/AMF files are static and will be replicated every time the file is used. Compromising the software or firmware involved in the AM process will enable dynamic manipulations. Modifications to the toolpath or G-code commands[5] can then be performed dynamically and triggered by various events. As the Stuxnet attack [14] illustrated, dynamic attacks can be significantly harder to detect; therefore, such attacks can remain active for a longer period of time.

[5]Both toolpath and G-code commends can contain or specify information about the manufactured object as well as instructions for the 3D printer configuration; the latter will affect the manufacturing process.

Direct/Indirect: Attackers can directly compromise an object by manipulating the object's specification (*e*.g., in the STL/AMF file), by changing the configuration values of the AM tools, or by modifying 3D printer instructions.

In addition, as discussed in [26, 16], indirect manipulations are possible. The timing of particular commands and status information, as well as the power supply of the 3D printer, can have a significant impact on the manufactured part's quality. Various classical network attacks (DoS, etc.). can also cause indirect manipulations, *e*.g., by causing G-code commands to arrive too late or out of order.

It should be noted that while the impact of direct manipulations on a part's quality can be predicted with a high level of certainty (and/or tested in a laboratory environment), the impact of indirect manipulations is rather stochastic.

B.3 Compromised Elements

The abovementioned manipulations are only possible if at least one of the following elements of the AM workflow is compromised. Note that not all compromised elements can exercise all manipulations and that manipulations available to different compromised elements might have different characteristics.

External Designer: In the given AM workflow, the blueprint files are provided to the AM service provider by external designers or are downloaded from the Internet. The designer is likely located outside of the trusted environment, and can be compromised by a cyber-attack or impersonated by a malicious actor. In this case, the STL/AMF files, originating outside the trusted environment, may contain an altered design, and both the introduction of defects and change of the orientation are possible. This represents a direct, static manipulation; whether it can be selective or not depends on the file format.

Controller PC: The controller PC operates on the STL/AMF design files, converts the design to G-code commands, and issues those commands to the 3D printer. If the controller PC is also used for non-AM activities such as web browsing, etc., it can be compromised. Exploitation of the controller PC can target software that is either directly related to the AM process, or unrelated software components. If malicious code is running on the PC, it can manipulate the design files, the generated tool path files, individual G-code commands, or the 3D printer's configuration, and ultimately also manipulate firmware updates for the 3D printer.

Network: In the AM workflow, network communication is used to transfer files from the object designer to the controller PC (external network communication) and between the controller PC and the 3D printer (internal network communication). Both can be compromised, enabling manipulation of the transmitted data. Depending on whether an external or internal network connection is compromised, a variety of manipulations can occur.

3D Printer: Control of the 3D printer can compromise the integrity of the printed object's design and the manufacturing process. Attackers can exert control through the 3D printer's firmware or hardware to achieve their goals. If the 3D printer is compromised, the whole spectrum of manipulations is possible.

B.4 Attack Vectors

An attack vector is a means by which an attacker can compromise and gain control over an element in the AM workflow. In this paper, we only consider cyber-attack vectors that can be used to compromise one or more of the elements described above. We distinguish between the following attack vectors:

B.4.1 Software Attacks

The software that is used on the controller PC or the 3D printer firmware can be compromised to execute malicious code. It has been well demonstrated that programs are full of vulnerabilities and exposed to arbitrary code execution. According to [9], the defect per KLOC (1000 lines of code) stands at 6.1 for the mission-critical software systems examined. Also, the Time-to-Fix can be anywhere from days to months. 3D printing software is no exception. Since there is a variety of software involved in the 3D printing process, there is a wide range of potential software vulnerabilities that can be exploited.

Code Injection using AM Files: The attacker can change the original design file at several stages, starting with the external designer, up to its representation in the 3D printer itself. A specially crafted AM file might be able to inject code into the software used at any stage.

There are two types of AM files: design files and toolpath files. Compromising the design files can be done either before their arrival, on the controller PC itself by malicious software, or during network transmission. A well-known example of a malicious code embedded in a CAD file is the MEDRE.A

worm [33]. As a recent study shows [24], the *Merlin* firmware frequently used in desktop 3D printers has numerous vulnerabilities. Therefore, it may also be possible to compromise the 3D printer itself.

General Infiltration Methods: Software attacks aim to compromise software running on one of the compromised elements. Compromise of a single device in the AM network can lead to the compromise of other components via lateral movement. An adversary will search for the easiest and least protected point of entry by leveraging common infiltration methods, such as external devices, brute force hacking, stolen credentials, or spear phishing via emails, fraudulent websites, malicious attachments, etc.

Software Supply Chain: Software vulnerabilities can be accidentally or intentionally inserted into the software at any point in its development, distribution, or use process. Software end users have limited ways of finding and correcting these defects to avoid exploitation. This vector is purely a cyber security concern, and full coverage of it is out of the scope of this paper. The areas of special concern for AM are compromised updates, such as the famous Flame malware [29], and backdoors in open source software, since open source is a vital part of the 3D printer software ecosystem.

B.4.2 Hardware/Firmware Attacks

As stated in our discussion of compromised elements, the focal point of the AM process is the 3D printer. By controlling the printer, the adversary can fully control the outcome of the manufacturing process. Most 3D printers are not directly connected to the Internet, and thus the adversary needs to gain a foothold inside the network and leverage it for a secondary attack on the AM equipment.

Firmware/Hardware Vulnerabilities: As with software, the existence of bugs in the firmware or hardware components is inevitable. They are less common and more difficult to exploit than software vulnerabilities, but they are equally hard to detect and correct.

Hardware Trojans: Hardware attacks pose a major security threat in the electronics industry. An adversary can mount such an attack by introducing a hardware Trojan into the system, which can manipulate AM process-related data.

Firmware Supply Chain: The firmware, being device-specific software, is vulnerable in much the same way as other AM-related software. In particular, malicious firmware updates would allow total compromise of a 3D printer.

B.4.3 Network Attacks

Since the AM equipment can reside on the same network as the controller PC, traditional network attacks can impact the 3D printing process.

General Network Attacks: Network attacks can lead to full control of communication by the attacker. AM environments are vulnerable to most forms of network attack, such as traffic relay, payload modification, sniffing, and hijacking. An attacker can also cause delays in the manufacturing process with a denial-of-service attack. Any interference in the timing of the printing process can have significant effects on the quality of the resulting object [26].

Protocol Vulnerabilities: There may be logic bugs in the protocols themselves. These kinds of bugs can be extremely harmful, since changing the protocol can be expensive and time-consuming. One well-known oversight is the Transaction ID Guessing attack on the DNS protocol [6]. In the AM context, researchers have shown that communication protocols employed by desktop 3D printers can be exploited. This enables retrieving current and previously printed 3D models, halting an active printing job, or submitting a new one [11].