

CYBER-PHYSICAL VUNERABILITIES IN ADDITIVE MANUFACTURING SYSTEMS

L. D. Sturm, C. B. Williams, J. A. Camelio, J. White, R. Parker

Design, Research, and Education for Additive Manufacturing Systems Laboratory

Department of Mechanical Engineering

Virginia Polytechnic Institute and State University

REVIEWED

Abstract

One of the key advantages of additive manufacturing (AM) is its digital thread, which allows for rapid communication, iteration, and sharing of a design model and its corresponding physical representation. While this enables a more efficient design process, it also presents opportunities for cyber-attacks to impact the physical world. In this paper the authors examine potential attack vectors along the Additive Manufacturing process chain. Specifically, the effects of cyber-physical attacks, and potential means for detecting them, are explored. Based on the results of this study, recommendations are presented for preventing and detecting cyber-physical attacks on AM processes.

1. Cyber-Physical Vulnerabilities in Manufacturing Processes

1.1 Attacks on Cyber-Physical Systems

Cyber-physical systems (CPS) are systems that integrate physical hardware with software systems, often with the use of a network. With the growth of the Internet of Things (IoT) the number of CPS systems on networks continues to increase [1]. Concurrently, cyber-attacks have become more prevalent in recent years, increasing in maliciousness and decreasing in visibility [1-3]. This poses a significant issue, as cyber-attacks on cyber-physical systems could result in damage to the machines themselves or the humans who interact with them.

A prominent example of a cyber-physical attack was the Stuxnet worm that targeted Iranian centrifuges used for refining uranium. In this attack the worm was able to infect the software system and affect the physical hardware, causing damage to the centrifuges. By sending false data back to the operators, Stuxnet was able to make it appear as though the centrifuges were operating correctly, while it caused them to damage themselves. The ability of Stuxnet both to cause damage to physical systems and to hide itself illustrates the ability of a cyber-physical attack to disrupt manufacturing systems and the need for physical methods of detection [4].

Another example of a cyber-physical attack is the hijacking of insulin pumps. In this case a hacker is able to connect to a Bluetooth enabled insulin pump to control the dose of insulin given to the wearer. By increasing or decreasing the dosage of insulin, it is possible to cause serious injury or even death in the user. The currently security system for these pumps is insufficient to prevent a cyber-physical attack that could have potentially lethal consequences [5].

1.2 Cyber-Attacks on Manufacturing

The previously mentioned examples demonstrate the ability for cyber-attacks to cross over into the physical world. Attacks on CPS are even more alarming when considering the ever-increasing amount of networked devices that are being connected to machines in the manufacturing world. A cyber-attack on these machines could cause injury to plant workers and damage to the machine itself. More insidiously, an attack could be designed to cause a process to produce faulty parts that might find their way into end-user products [6]. For example, an attack could be designed to affect the production of a jet turbine part such that it would pass inspection but fail during operation and cause significant damage.

With the rise in both the number of CPS connected to networks and in malicious cyber-attacks, there is a clear need for research to understand the vulnerabilities of cyber-physical systems. While methods exist for detecting cyber-attacks on computer systems, no such research has been done on detecting an attack from the physical parts created by the attack. As such, the authors have begun an investigation on cyber-physical vulnerabilities in manufacturing systems [7-8].

1.2 Context: Cyber-Physical Vulnerabilities in Additive Manufacturing

In this paper, the authors scope their research solely on Additive Manufacturing (AM) systems. The process chain of these networked machines has unique vulnerabilities that warrant a detailed investigation due to their ability to fabricate parts in a layer-wise fashion. For example, voids can be placed inside of a part and the material properties of internal layers can be changed without affect the exterior layers. Because of the potential damage from a cyber-physical attack, there is a need to look at AM systems to determine what vulnerabilities exist and how to prevent and mitigate the threat of cyber-attacks.

An overview of the AM process chain is presented along with a description of potential vulnerabilities in Section 2. Based on this evaluation, a cyber-physical attack case study is presented in Section 3 in which the .STL file structure is attacked. The effectiveness of this attack is evaluated through part testing (Section 4). Finally, in Section 5, the results of the attack are analyzed to identify ways of preventing and mitigating future attacks.

2. Cyber-Vulnerabilities in the Additive Manufacturing Process

To be able to prevent a cyber-attack, one must first understand the vulnerabilities and weaknesses of the system. To do this, it is necessary to follow a cyber-attack through the process chain, from conception to simulated deployment. In this section, the AM process chain is examined for potential vulnerabilities to cyber-attacks.

The digital nature of the additive manufacturing process chain, shown in Figure 1, provides an opportunity for a cyber-attack to cross into the physical world. There are four main steps on the process chain where an attack could take place: the CAD model, the .STL file, the toolpath file, and the physical machine itself.

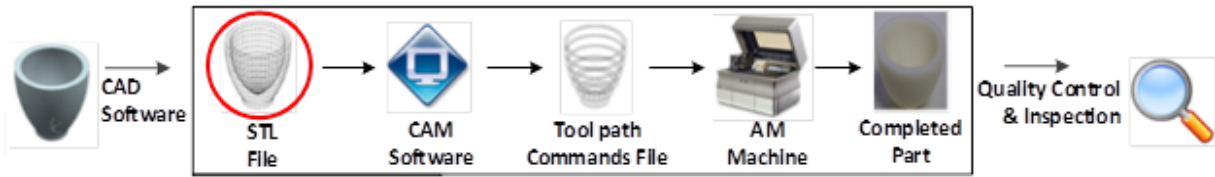


Figure 1. Additive Manufacturing Process Chain

2.1 CAD Model

The CAD model is the first step in the additive manufacturing chain and is common both to additive and subtractive manufacturing. This step in the process is the most valuable in terms of information, as it contains all of a part's geometric data. If connected to a PLM software, the data could also include information related to simulated performance (e.g., results from finite element analysis, computational fluid dynamics, multi-physics simulations, etc.) and the associated parameters of the part's intended use.

Attacks at this point in the process may focus on stealing or corrupting files. One example of this is ACAD/Medre.A worm, which was designed to infect and steal AutoCAD drawings [8]. Another example of this is the CryptoLocker malware, which encrypts a user's files and then demands a ransom for the encryption key to unlock these files [9]. Because CAD files are the basis from which other drawings and files are generated, the theft or loss of these files can be costly. Furthermore, a competitor who stole these files would have access to all of the design steps that went into making the CAD model and could easily reverse engineer or modify the part.

The complex nature of most CAD files makes it more difficult to directly alter the part; however an attack could be designed to do so. For example, a CAD file for a crankshaft could be altered to reduce the area of the load bearing member, resulting in premature failure. Any corruption at this phase would propagate through the entire process chain, resulting in a part that is bad from start to finish. However, because parts may still be edited during the CAD phase the chances of detection of an attack increase.

2.2 .STL/.AMF File

The second step in the process chain is to convert the CAD model into a .STL file, the current defacto file standard in additive manufacturing. Upon conversion, most of the model data is lost, and only the surface information of the part remains. At this stage the model is no longer represented by complex mathematical equations, but by a simple surface geometry composed of triangular elements, called facets. Each facet is defined by three vertices, specified by a set of x,y, and z coordinates. Each closed set of facets comprises a shell, and multiple parts can be represented inside a single .STL file by including multiple shells.

As with the CAD file, the part could be altered to reduce its performance. By changing the coordinates of the vertices it is possible to edit the model. Additional vertices can also be added to create new features. Additive manufacturing is unique in that the interior of a part can be altered without affecting the exterior of the part, resulting in a part that looks and feels strong, but is weak on the inside.

Section 3 gives more details on the potential presented in .STL attacks and on why this attack vector was chosen as a focus.. Both the .STL and .AMF file (the current ISO/ASTM

standard) are simple file formats that contain vertices of facets and so are both susceptible to the same types of attacks. The .AMF file contains more information that makes it vulnerable to a longer list of attacks than the .STL. For this reason the focus will be on the .STL file, with the understanding that these techniques are also valid on the .AMF file. The CSG history of how the geometry was created is no longer present in the file, so it becomes harder to reverse engineer the information needed to create a similar part. Despite this loss of information, a theft of a .STL file is still costly as it (i) contains all of the information needed to fabricate the geometry of the part (which could result in the production of counterfeit copies) and (ii) the surface geometry data can be attacked to nefariously change part geometry.

2.3 Toolpath File

Upon receiving a .STL file, each additive process converts the model into layers and generates a toolpath from these layers. Conceptually similar to GCODE, this toolpath file contains the commands for the controllers that move the AM systems' coordinate axes and deposition mechanisms (e.g., extrusion federate, laser power, inkjet pulses, etc.).

Potential attacks on the toolpath are to place/remove material in the wrong location, to cause layers to be placed too close/far away from each other, and to damage the part/machine by driving the tool into the part/machine. Toolpath attacks offer the most freedom in what an attack can achieve because any operation the machine could normally use when creating a part can be altered in the toolpath. A toolpath file could be intercepted by a virus on the machine's computer or when the file is sent from one computer to the machine. For example, with a wireless printer the information might be sent over a network where it could be intercepted.

Theft of the toolpath file is less valuable than that of the .STL file, since it is machine specific. While it could still be reverse engineered or used on another machine of the same type to produce the part, it requires more work to do so than an .STL file. Attacking the model through the toolpath does allow any of the changes that can be made in the .STL file to be made to the part, however the implementation of such changes is more difficult in the toolpath.

2.4 Physical Machine

The final step in the AM process chain is the machine itself. This stage of the process is vulnerable to Stuxnet type attacks, in which a worm alters the firmware of the system's controller PC. It is noted that such a sophisticated attack vector is not needed, as most AM systems have open USB ports for maintain the machine's software.

Using this as an attack vector, the machine's process parameters (e.g., nozzle temperature, laser intensity, etc.) could be altered to affect the material properties of the part. For example, inkjet nozzles might be turned off at one point to prevent material from being placed, or turned on at another point to place excess material. Temperatures of extrusion nozzles could be altered to change the mechanical properties of the extruder or to clog the nozzle all together. Laser intensity could be altered to provide lower energy density to the bed to change the final part properties. Figure 2 shows configuration data for a layer being printed on a Connex 350 printer that was intercepted in plaintext using WireShark [7]. By intercepting and altering this configuration file it would be possible to remotely change the material properties by altering the UV intensity. This type of attack can be difficult to detect without some type of independent monitoring.

The two limitations on this attack method. First, no model data is available to be stolen or altered. Second, any attack must be specifically targeted to each machine. The difficulty of attacking the machine can also vary significantly from one machine to another. While one machine might be particularly vulnerable to an attack, another machine could be well secured and hard to infect.

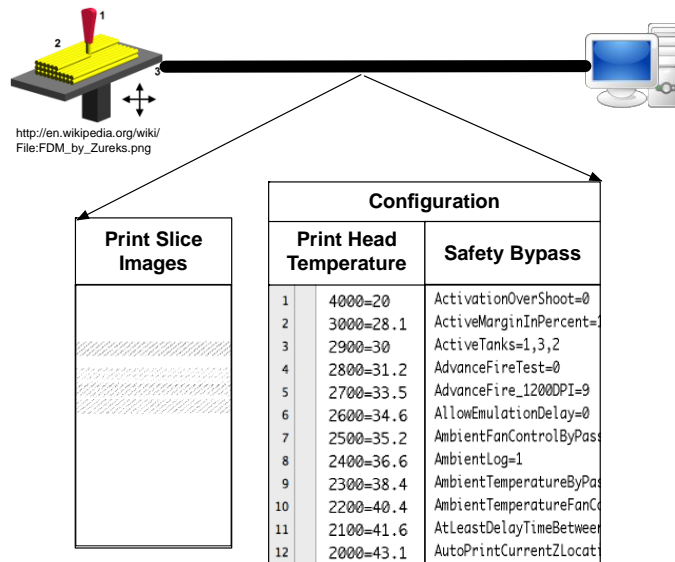


Figure 2. Object configuration data intercepted from WiFi using WireShark [7].

3. Case Study: Cyber-Attacks on the .STL File

This section will cover the development of a cyber-physical attack by examining some of the considerations that go into attacking a .STL file. The .STL is the one potential attack that does not require specific modification for each AM machine as it is an attack on the standardized file format that every AM machine takes as input. As STL file creation occurs at the beginning of the process chain, and is general to every AM machine, a focused attack on the file could have severe implications and could cripple an AM production line. To prevent replication of this attack, specific details of the final attack algorithm have been omitted.

3.1 Types of Attacks

As described in Section 2.2, while a .STL file only has surface data in the form of a list of triangle vertices, there are several different types of file attacks that can affect the final part geometry. The effects of these attacks on a sample part geometry (an ASTM tensile test specimen) are illustrated in Figure 3.

- *Corruption/Encryption* – A traditional cyber-attack where the file is damaged or encrypted, this renders it inaccessible to the user. A corruption or encryption attack is a straightforward attempt to damage or extort the owner of the file. By rendering the file unusable it makes it very evident that an attack has taken place. However, if backups are

not available, this type of attack can cause a lot of damage before it is detected. (Figure 3a)

- *Scaling* – The part is scaled up or down in one or more axes resulting in a changed form that may affect the fit or strength of the part. For example, a tensile test specimen that is 10% thinner may not be noticeably different to the eye, but will cause a measurable change to performance. This type of attack is easily detected in software or by measuring the completed part. (Figure 3b)
- *Indents/Protrusions* – Small protrusions or indents may be added to a part to affect the fit, surface finish, or strength of the part. For example, the inside of a printed duct or pipe might contain indents that negatively affect the flow. These attacks are visibly detectable, but may be placed in locations where they are difficult to see or measure. (Figures 3c and 3d)
- *Vertex Movement* – One or more vertices in the part is moved, resulting in a changed form that may affect the fit or strength of the part. This has the advantage of manipulating the shape of the file without changing its size. Unlike scaling, vertex manipulation allows almost any section of a part to be sized both up and down, as well as warped or altered. Measurements can detect this attack, but by altering areas that are difficult to measure, the presence of this attack can be hidden. (Figure 3e and f)
- *Voids* – Similar to an indent, a void is an area where material has been removed from the model. A void differs from an indent in that it is completely enclosed inside the existing model. Because voids are completely enclosed they are undetectable by dimensional measurements and may be difficult or impossible to find visually. The use of supporting material in many processes also renders the void undetectable by weighing, since the void is filled with a structurally deficient, but equivalently dense material. The presence of a void weakens the strength of a part, and if placed in a loadbearing location, may cause part failure. This attack does increase the file size slightly, but the small change is unlikely to be noticed.

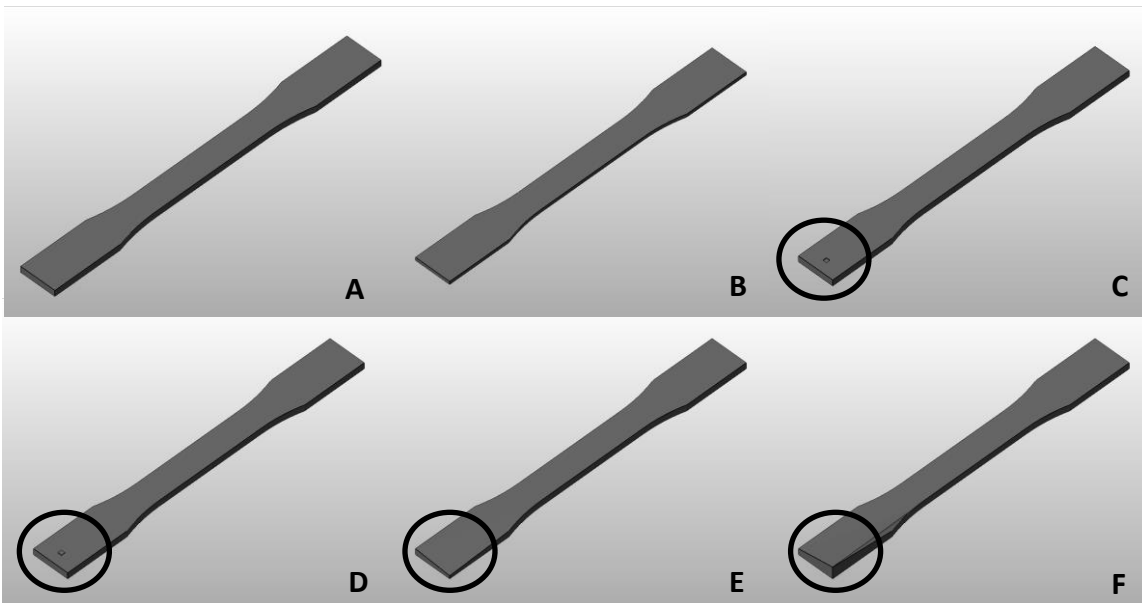


Figure 3. .STL Attacks on an ASTM dogbone. A) An unaffected dogbone B) A scaled down dogbone C) An indentation D) A protrusion E) A vertex moved inward F) A vertex moved outward

The first four types of attacks are shared by both additive and subtractive manufacturing and can have a demonstrable effect on the part being produced [8]. The last attack, the creation of voids inside of a part is unique to additive manufacturing due to the layer based fabrication method. Because voids are difficult to detect once a part has been created and because they are an attack that is unique to AM, this specific attack type was chosen for further investigation in this study.

3.2 Void Attack Considerations

There are several factors that must be considered when attacking a .STL file using a void:

- *File Size* – When altering an .STL file a change in file size can may make it easier to detect an attack. Some attacks, such as scaling or moving a vertex, do not change the file size. Other attacks, such as adding voids or other features, will change the file size depending on the complexity of the feature added. It is especially concerning that a void can be added by inserting as few as four additional facets, an increase of only 200 bytes. This size increase is negligible in all but the simplest files, as .STL files can easily be in the megabyte range.
- *Location* – The location of a void in a part has a significant impact on the effect of the attack on a part. A void is not a threat if it does not result in any reduction in part strength. A well place void is a concern, as it can lead to a critical part failure.
- *Void Shape* – The shape of the void affects the amount of information needed to represent it in a .STL file. The shape can be as simple as a tetrahedron – adding only four facets and 200 bytes to the file.
- *Void Size* – Void size is important for affecting the mechanical strength of the part. A void that is too small will not be fabricated, due to process resolution limits. Large voids increase the chance of part failure, but may be easier to detect.
- *Void Number* – The number of voids placed can be varied based on the desired effect. A larger void might be made less noticeable by replacing it with several smaller aligned voids. The addition of more voids will add more to the file size and may make the change more noticeable.
- *Full Enclosure* – It is essential that any void placed is fully enclosed in the original part. If a void is not fully enclosed inside of a single shell it will cause printing errors that can alert the user.
- *Run Time* – The time it takes for the attacking software to analyze and place a void is directly related to how easy it is to detect. A several millisecond delay is likely to go unnoticed by a user, while a several second delay is likely to draw attention.
- *Silent* – The software does not set off any warnings. For example if the void contained a facet that had the normal facing in the wrong direction Netfabb would display a warning to the user alerting them that there was a problem with the file.

The most dangerous void attack would be one where the void was located in a structurally important location and sized such that it was difficult to detect while still causing a critical structural failure in the part. Such an attack would run quickly, add a minimal amount of data to the part file, and would not cause any warning messages when inspecting or printing the part.

3.3 Attack Embodiment

As mentioned, it was decided that a void attack would be used for the study. This attack was chosen for the following reasons:

- *Uniqueness* – The ability to place voids inside of a part is unique to additive manufacturing.
- *Visibility* – Unlike a protrusion, indentation, or other external change, a void cannot be physically measured once it is enclosed inside of the part. Because of this, it was desirable to determine what would or would not make an internal void visible. Most quality control processes rely on dimensional checks or visual inspection methods, such as operators or computer vision systems. By making the void invisible, these quality control processes can be subverted.
- *Threat* – Because voids are not readily visible when looking at a model on a computer screen, and are not measurable once the part has been made, they present a higher likelihood of passing undetected.

The attack software featured three key pieces of functionality: (i) the ability to automatically determine a location for the void, (ii) the ability to ensure that the void is located completely inside the part, and (iii) the ability to automatically scale the void based on the part. In addition the voids placed did not set off warning messages in Netfabb, Objet Studio, or any of the other software we worked with the parts in. We developed this software in order to assess its potential impact, to gain an understanding of what the effects of such an attack were on a part, to see if it could be detected through existing quality control measure, and if not, to see what new control measures would be needed to detect it.

Void Placement

In the attack software presented there are two methods for locating voids, random placement and targeted placement. In the random placement mode, a random location inside of the part is chosen and the void is placed there. The targeted mode is a semi-intelligent method for locating the void. Because .STL files are not solid models it is difficult to attempt any type of structural analysis of the part. Additionally, since the parts infected could vary significantly, and the purpose of each part is not known, it is hard to place a void in any meaningful way. To solve this problem the software uses a semi-intelligent method to place voids in locations that are likely to have stress concentrations. Areas with holes or sharp changes in curvature are often located at important locations in a part. To detect these areas in a .STL file the density of the mesh can be estimated. Similarly to FEA, more complex areas will be represented by a denser mesh than flat areas. By placing voids near locations where the mesh is dense the likelihood of placing them somewhere that will cause a critical failure increases significantly.

The ability to place a void completely inside of the part is a key part of a void attack, both for the functionality of the void and for avoiding detection. If a void is placed completely outside of the existing shell of the part it will serve no function and is not a concern. Similarly, a void that extends partially outside of a part will cause errors that will draw attention. The only dangerous void attacks are those where the void is completely enclosed. The task of placing a void inside of an arbitrary .STL file poses several challenges. First, .STL files may contain multiple shells. A void may be fully enclosed inside of the part, but still intersect multiple shells causing an error. Second, because an .STL file may contain almost any geometry, they can be convex or concave.

Because of this techniques for determining collisions based on convex geometries cannot be used. Other approaches, such as determining the center of mass, meet with similar difficulties as the center of mass may be outside of the object. In analyzing this problem a two-step method was found that could be used by an attacker to ensure that a void is placed completely inside of a part.

Void Scaling

Another feature that makes a void attack more dangerous is the ability to scale the size of the void to better fit the part. The user has little concern from the placement of a void that will not have any harmful effects. An attacker can increase the chance of failure by making the void larger, while still keeping it small enough to avoid detection. An attacker can easily modify the size of a void, however this increases the risk of it extending outside of the shell it is enclosed by. By combining scaling techniques with the previously discovered method for ensuring the void is completely enclosed, the chance that the void placed by the software would cause a failure was increased.

The final attack software analyzes an .STL file, places a tetrahedron shaped void at either a random or semi-intelligent location, scales the void to a more idea size, and attempts to ensure that the void is fully enclosed inside of a single shell. Adding the void adds a total of four new facets to the file and creates a new negative volume shell consisting of those facets into an existing shell. For a binary .STL file the total change in file size is 200 bytes since each facet requires 50 bytes to represent. Though this might be a discernable change in a small, simple part, it will be nearly invisible in a larger more complex part that has a file size of several megabytes.

4. Effects of .STL Void Attack

To ascertain the potential impact of this specific attack, two experiments were performed. First the authors evaluated the effect of a “printed void” on the mechanical strength of a printed specimen.

4.1 Effect of Voids on Part Strength

The goal of the attack was to evaluate the impact of a void placed inside of a part for measurable degradation of part quality. To evaluate the effectiveness of a void at causing a failure an ASTM Standard D638-10 tensile test specimen, or dogbone, was used as a sample part. This part was chosen because it provided a straightforward way of measuring the quality of a part with and without a void.

Figure 4 shows a cross section slice of a part with a void placed inside of it using the software. To quantify the effects of the void placement, finite element analysis was run to determine if the void was likely to cause a failure. Unsurprisingly, the Von Mises stress of the part is highest at the location of the void (Figure 5).

In order to verify this result for parts manufactured using an AM process, several dogbones with and without voids were printed on via Powder Bed Fusion (a Sinterstation 2500 Plus machine) using Nylon 12 powder. Upon testing, all of the dogbones containing voids fractured at the void location, while the dogbones without voids failed normally, as shown in Figure 6. The average reduction in yield load was 14%, from 1085N to 930N, and the strain at failure was reduced from 10.4% to 5.8% as shown in Figure 7.

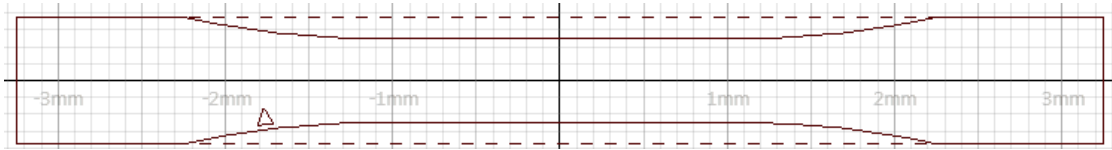


Figure 4. Cross sectional slice of a dogbone infected with a void.



Figure 5. Von Mises Stress of a dogbone infected with a void.

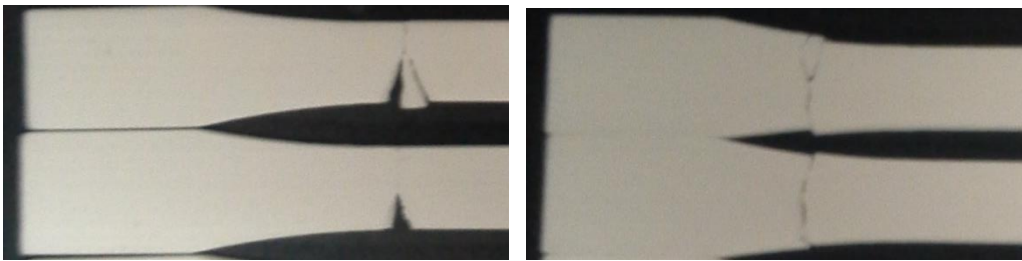


Figure 6. On Left: Uninfected dogbones breaking at the neck. On Right: Infected dogbones breaking at the void location.

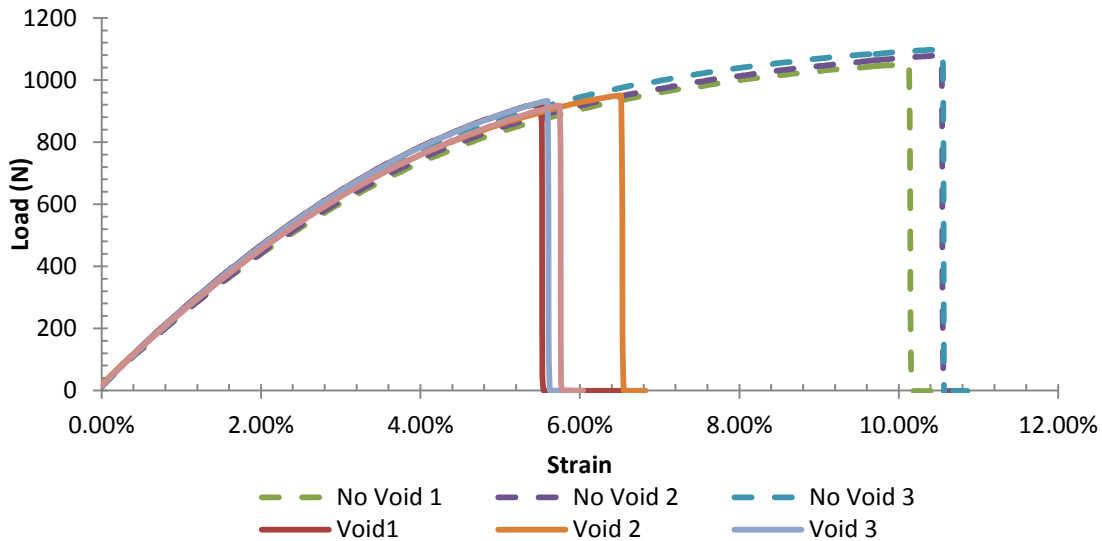


Figure 7. Load and strain data of parts with and without voids.

In addition to the reduction in mechanical strength, the dogbones containing voids were tested to see if the voids were detectable through common inspection techniques. Physical dimensions of the part revealed no differences, since the void was full enclosed. Weighing of the part did not reveal any noticeable differences since the void was small and filled with loose nylon powder, which had a similar density to the finished model material. The only effective

way of detecting the presence of a void was to hold the dogbone up to a light source, revealing a darker area where the void was located; of course this technique is only useful for this thin (3.2 mm) part.

4.2 Recommendations

The results of this study show that more work is needed to protect AM systems against cyber-physical attacks. One of the great benefits of the .STL or .AMF file format is the ease in which files and designs can be freely shared. This is a great benefit to AM and would be lost if an attempt was made to move to a more closed “secure” format. Trying to create a “secure” replacement for the .STL would not only harm the field of AM as a whole, it would also fail to address the root of the problem. Any security system can be broken eventually, and replacing the .STL file would do little more than provide a false sense of security. A better way of addressing the threat of cyber-physical attacks is to develop better monitoring systems and procedures and to better train the workforce to detect these attacks.

- *Improved Software Checks* – Detection of shells with negative volumes, detection of very small shells, and detection of small features.
- *Hashing/Secure Signing* – Allow users to check that the file they receive is the same as the one that was sent.
- *Improved Process Monitoring* – Monitoring process parameters through indirect “side channel” methods. By observing physical parameters such as the melt pool temperature indirectly from machine settings such as laser power, attacks are more likely to be discovered.
- *Operator Training* – Better education of the workforce on the threats of cyber-physical attacks and on how to detect when an attack may be occurring.

The first step that can be taken to prevent cyber-attacks is to improve the default software checks that are run in common software programs used for checking .STL files. Existing software, such as Netfabb Pro, is easily capable of detecting shells with negative volumes. However, its default error check does not present the user with an alert when a negative volume shell is detected, even though in most parts negative volume shells represent an error or attack on the file.

A second check that can be performed is a test for shells with very small volumes since they are unlikely to be important features, and are often unneeded artifacts left over from another process like 3D scanning. Testing for these shells increases the chance of detecting a void based cyber-attack in addition to alerting the user of potentially unwanted or unneeded data.

Finally, more research is needed into detecting small features that are not independent shells. Very small features that cannot be built by an AM process may indicate the presence of a cyber-attack. Even if the features are not malicious, the user should be alerted that they will not be printed correctly. The inclusion of these three checks would not only make detecting an attack easier, but would also help the user in identifying potential problems in their models.

Hashing is a technique commonly used in security to ensure the validity of a file. The file is run into the hashing function, which generates a string of character called a hash. The hash is then posted along with the file. When a user downloads a file they can run it through the same hashing function and compare the resulting hash with the posted hash. If the two hashes match,

the file can be assumed to be identical to the original. What makes hashes effective is their ability to convert a large file into a simple string that can be easily shared. Any small change in the file generates a large change in the hash. The simple adding of a single character to a text file will completely change the hash that is generated.

At the most basic level hashes can be (and are) used to ensure that the .STL file received has not been tampered with. While this does increase security it adds some additional work to the process. Additionally, a file could potentially be attacked after it has been received and hashed, or before it was hashed to begin with. This problem could be addressed in part by including the hash function at the time of file creation, within the CAD software itself, and generating a hash function at the last step in the process, where the .STL file is loaded in to the printer software to be converted to a toolpath. One area of further research is the creation of a physical hash, a hash that incorporates elements from both the physical and the cyber side to give added resilience against cyber-attacks.

Improved process monitoring is an ongoing goal and area of research for additive manufacturing in an effort to better control and improve them. Some solutions have already been explored, such as using optical sensors to provide closed loop control for layerwise laser melting (LLM) [10]. Indirect measuring, or “side channel” measurements, such as measuring the temperature of the melt pool to determine the laser power instead of simply asking the machine can be effective at detecting cyber-attacks effecting the machine parameters, since, as Stuxnet demonstrated, a clever attack can cause a system to report false data.

While feedback systems provide valuable information for process control, used alone they are insufficient to detect cyber-attacks. If the .STL file for a part is infected with a void a closed loop system will notice the void, compare it to the part file, and conclude that it is an intended design feature. For this reason more research needs to be done to establish statistical process control (SPC) for AM systems that works to detect the effects of cyber-attacks. By combining side channel measurements and using SPC to establish baseline operating parameters, systems can be made more robust and also more resistant to cyber-attacks.

The final area that needs improvement is in education and training. The vast majority of malicious software is installed or transmitted unwittingly by uneducated users. Educated workers are more likely to detect and prevent a cyber-attack. Informing students and workers about the potential risks of cyber-attacks can prevent these attacks from occurring and help diagnose them faster when they do occur.

5. Conclusions

With the increasing number of manufacturing systems connected to networks, more work needs to be done to ensure the safety of these systems. Additive manufacturing systems in particular have unique vulnerabilities presented by the ability to affect the internal layers without affecting the exterior. An overview of the AM process chain (Section 2) showed that the .STL file was the most vulnerable attack vector due to its universality and ease of editing. Further investigation into the .STL file revealed a method by which a void could be placed inside a part, while avoiding detection by common process checks (Section 3). This void placement was demonstrated to cause a 14% reduction in yield load in a tensile test specimen. Based on the results of this study, it appears that a real threat from cyber-attacks exists and that further research needs to be done on how to mitigate such attacks. The inclusion of software checks,

hashing, process monitoring, and worker training are proposed as methods of reducing these threats. Future work includes the development of physical hashing techniques and of improved side channel process monitoring and control

References

- [1] Evans D. The Internet of Things: How the Next Evolution of the Internet is Changing Everything. CISCO White paper; 2011. Available from: https://www.cisco.com/web/about/ac79/docs/innov/Io-T_IBSG_0411FINAL.pdf [Retrieved March 13, 2014].
- [2] Bayuk J, Cavit D, Guerrino E, Mahony J, McDowell B, Nelson W, Snelvel R. Malware Risks and Mitigation Report. Washington, DC: BITS Financial Services Roundtable; 2011.
- [3] Watin-Augouard M. Prospective Analysis on Trends in Cybercrime from 2011 to 2020. National Gendarmerie; 2011
- [4] Falliere, N., Murchu L. O., Chien, E., “W32.Stuxnet Dossier”, Symantec, (2011) Available from: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf [Retrieved August 9, 2013]
- [5] C. Li, A. Raghunathan, and N. K. Jha, “Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system,” in Proc. IEEE Int. Conf. Health Netw. Appl. Services, Jun. 2011, pp. 150–156. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6026732>
- [6] National Defense Industrial Association, “Cybersecurity for Advanced Manufacturing,” Retrieved May 28, 2014 from <http://www.ndia.org/Advocacy/LegislativeandFederalIssuesUpdate>
- [7] H. Turner, J. White, B. Amos, J. Camelio, C. Williams, and R. Parker, “Bad Parts: Are our Manufacturing Systems at Risk of Silent Cyber-attacks?,” *IEEE Secur. Priv. Appear.*
- [8] ESET, “ACAD/Medre.A”, ESET Whitepaper: June 2012 Available from: http://www.welivesecurity.com/media_files/white-papers/ESET_ACAD_Medre_A_whitepaper.pdf [Retrieved January 7, 2014]
- [9] M. Sorin, “Are your IT professionals prepared for the challenges to come?”, Computer Fraud & Security, Volume 2014, Issue 3, March 2014, pp. 18-20, <http://www.sciencedirect.com/science/article/pii/S1361372314704725>
- [10] Craeghs, T., Bechmann, F., Berumen, S., Kruth, J.-P., “Feedback control of layerwise laser melting”, Physics Procedia, 5, 505 - 510 (2010).