Cybersecurity Considerations for Digital Twin Implementations

Authors:
Mark Hearn
Sr. Director, Strategic Market Business Development
Irdeto
mark.hearn@irdeto.com

Simon Rix
Technical Evangelist
Irdeto
Simon.rix@irdeto.com

INTRODUCTION

Digital twins are a prime example of how Industry 4.0 is transforming the industrial and manufacturing industries and creating a vast number of opportunities and efficiencies within these sectors. Previously, only the domain of large enterprises due to the expense of the computing resources required for development, digital twins are now rising in popularity across a range of market segments. As digital twins increase in availability and operation, many organizations are looking to digital twin technology as a means to improve efficiency, prevent or manage downtime and, potentially, to monitor for attack against the real system. Therefore, it is not surprising that Gartner identified digital twins as one of its "Top 10 Strategic Technology Trends for 2019."[1]

The rise of digital twins is part of a wider smart technology revolution in the industrial and manufacturing sectors, with a recent Smart Factory Market report by Markets and Markets projecting that the smart factory market will be valued at USD 205.42 Billion by 2022.[2] Although the benefits of the wider use of digital twins and smart technology are clear, as with any technological advancement based on connectivity, it also increases the number of vulnerabilities (or attack surface) for software, risk of Intellectual Property (IP) theft and exposure of critical processes.

As a result, companies innovating with Industry 4.0 have become extremely viable targets for nefarious actors. Recent research of 220 security decision makers in industrial and manufacturing organizations, conducted by Irdeto and Vanson Bourne, found that 79% of those surveyed have experienced an IoT-focused cyberattack in the past year.[3] With this in mind, it is clear that connected systems and software can be exploited (sometimes easily) for ill intent, if control falls into the wrong hands. Organizations must therefore seriously consider the security implications of a digital twin and take a new approach to security.

[1] Gartner, Gartner Identifies the Top 10 Strategic Technology Trends for 2019, October 2018, https://www.gartner.com/en/newsroom/press-releases/2018-10-15-gartner-identifies-the-top-10-strategic-technology-trends-for-2019

[2] Markets & Markets, Smart Factory Market worth $244.8 billion by 2024, March 2019, https://www.marketsandmarkets.com/PressReleases/smart-factory.asp

[3] Irdeto, Irdeto Global Connected Industries Cybersecurity Survey, May 2019, https://go.irdeto.com/connected-industries-cybersecurity-survey-report/

Digital Twin Threats

There are a multitude of use cases for digital twins in industrial and manufacturing environments for everything from production to safety purposes. While the definition of a digital twin may refer to a critical data model in one context or software containing IP in another (or even both), in all cases, digital twins touch production assets that may be business critical—and herein lies the risk of digital twins that must be considered and mitigated. A Wall Street Journal article from Deloitte cites an example of an industrial manufacturer using a digital twin to bring down liabilities and maintenance costs in the field.[4] When it comes to IoT and IIoT, digital twins could actually be used for security, as outlined by Gerald Glocker on the Bosch ConnectedWorld blog in 2018.[5] However, as Michal Cobb of SearchSecurity points out, "While digital twins can improve the security of IoT devices and processes, it is critical to consider the security of the twins themselves when implementing one."[6]

While the definition and implementations of digital twin vary, this article focuses on software implementations, particularly in cases where the digital twin implementation uses both existing intellectual property and new innovations for the purposes of best matching the functionality and performance of the real system the twin is mirroring. However, the original security models such as hardware security, air gapping, etc. used to protect software in the real system may not be applicable on the twin itself, as the twin is often deployed on standard platforms such as an industrialized PC. This article focuses on the criticality of securing the digital twin's platform and hardening its software—both for the safety of the digital twin and for the real system it is monitoring.

Indeed, the security threats associated with the digital twin may be a risk to the physical systems they represent. The key for any digital twin is that to effectively assist the ecosystem, it must be as accurate a representation of a real system (or selected aspects of a real system, depending on the purpose of the digital twin) as possible. However, inevitably in the design process, gaps between the digital twin and the actual system will likely exist. This is not a problem if the gaps are fully understood and considered in the security strategy. Unfortunately, we have seen that the security gaps between the twin and the physical system can often be poorly understood. For example, physical hardware may frequently have advanced security

[4] Mussomeli, Parrott and Warshaw, Meet Manufacturing's Digital Twin, Deloitte & Wall Street Journal, September 2017, https://deloitte.wsj.com/cio/2017/08/09/meet-manufacturings-digital-twin/

[5] Glocker, A primer on digital twins in the IoT, Bosch Connected World Blog, October 2018, https://blog.bosch-si.com/bosch-iot-suite/a-primer-on-digital-twins-in-the-iot/

[6] Cobb, With an IoT digital twin, security cannot be forgotten, Internet of Things Agenda, February 2019, https://internetofthingsagenda.techtarget.com/tip/With-an-IoT-digital-twin-security-cannot-be-forgotten

features such as secure micro-controllers which may not be available on the platform running the twin. Such gaps, if not addressed, can lead to threats to the system, as well as the manufacturer's business model. These issues are also alluded to in a 2017 article by Enterprise IoT Insights which identifies insufficient security configurability and insecure software or firmware as two of the top ten IIoT security vulnerabilities.[7]

A significant security concern with a digital twin being such a close representation of an actual system is that if the twin is obtained by a hacker, it can then serve as a blueprint to the real system, identifying components, their behaviors and their interfaces. This immediately gives the hacker an internal view of the system to be attacked and will help them to identify vulnerable attack points. In this scenario, one can assume that prior to a physical system attack, a hacker would have an entire script mapped out using the compromised digital twin, allowing penetration of the actual system with minimum detection or disruption. Digital twins can also be used for penetration testing of a physical system's interfaces, thereby allowing the attacker to fine-tune their attack mechanisms.

If a digital twin is compromised by a hacker, it also has the potential to expose the organization to backend system attacks, as these systems may be called directly by the twin—this threat opens a map of backend systems to a hacker. Code analysis by a hacker can quickly identify the API calls required for these backend systems and possibly expose the twin's access credentials during authentication functions. It should be noted that while code analysis (often called reverse engineering) is a viable attack on all platforms, software designed for Windows or Linux platforms can often be more easily reverse-engineered due to the common availability and low price of reverse engineering tools. Once these access points are obtained, the hacker can then easily spoof the behavior of the twin or even access the physical system (spoofing the twin), potentially providing access to system-wide data.

IP is another area for diligence. Frequently, a digital twin includes critical IP, often innovative but also repurposed legacy, which may be damaging to the IP owner if it is reverse engineered. In today's geopolitical situation, adequate security must be put into place to protect the investment that an innovative company makes to gain market share—market share that is at risk from companies from other regions of the world. For example, a company with a digital twin of their industrial control system (ICS) may utilize real-system code in their digital twin to offer a more accurate twin experience. However, such a situation increases business risk further, as any theft of damage to the twin would impact both the new twin as well as the existing ICS components themselves.

[7] Blackman, IIoT security: The top 10 security vulnerabilities, Enterprise IoT Insights, November 2017, https://enterpriseiotinsights.com/20171127/security/iiot-security-top-10-security-vulnerabilities-tag40-tag99

Steps to Securing a Digital Twin Implementation

Earlier, we discussed the increased amount of digital twin development in Industry 4.0 activities. This, combined with the identified availability of general-purpose twin development environments, is a major reason for the growth in digital twin usage, and the providers of these environments also have a vested interest in ensuring security. For example, Microsoft publishes a variety of best practice tips for digital twins based on Azure.[8]

To minimize the risk associated with the development and operation of a digital twin or any system within the organization, the involved parties must consider some basic guidelines during design and implementation.

The first place to start may sound odd, but security only flourishes when the organizational culture actively enables it in an ongoing manner. Clearly there is a difficult and delicate economic balance to be found in a competitive marketplace where time to market, solution features and profit compete with quality and security. In the modern world, it is imperative that corporate leadership enables and empowers healthy ecosystems—and that must include secure design as part of regular operations.

Organizations must look to implement security in their systems from the ground up, fully understanding and planning for the security measures which are put in place. This begins with a clear and well-defined secure software development lifecycle (SDLC) management process that includes all aspects of the lifecycle, from inception to system retirement. Once in place, the SDLC must become a key part of product development.

As there is no prescriptive SDLC formula, this paper will discuss security concepts from a general, secure product development viewpoint. A clear set of high-level requirements or goals is essential to begin any project. They need to be specific and measurable. Once the security requirements are understood and committed, it is imperative that attention be given to the software design process. This step is often rushed and can lead to severe problems later in the process. Good software design must take security and testing into account at the earliest point as these items often impact solution design. The design phase should only be considered as complete when the design, test plan and security requirements are met.

The software development phase then seeks to implement the agreed design, test and security specifications. Ideally, security testing should be included in regular product testing and automated to allow for iterative testing through the software life cycle.

To achieve good quality and security of the software source code, it is helpful if one institutes automated processes to scan source code for language conformance, style, flaws and known vulnerabilities, as well as open source compliance to company

[8] Microsoft, Security best practices, August 2019, https://docs.microsoft.com/en-us/azure/digital-twins/security-best-practices

policy (whether that is to avoid it, make sure the latest versions are used or to ensure compliance to license requirements). These automated processes need to be enhanced by best practices such as secure coding methodology, peer code review and good repository control. Where applicable, specific security testing techniques such as fuzz and penetration testing should be applied.

The activities discussed to this point focus on creating and implementing a software design and development that meets the clear requirements of both quality and security.

Software protection, sometimes referred to as "software hardening," has a rich set of techniques to draw on that make the resultant binary executable hack resistant. These techniques include data and software transformation that effectively protect the "data in use" in the design, as well as enhance the level of effort required to reverse engineer the executable. Simultaneously merging functions together or in-lining functions to break up the modular code and then entangle transformed data with the altered control flow of the software render the reverse engineered binary very hard to understand.

This transformative technology offers multiple benefits. Not only is it very hard to understand—thwarting the adversary's efforts to attack a system—it is equally difficult to modify the protected binary to introduce the desired nefarious functionality and still have the software operate in a reliable manner.

Software protection can also inject more active defenses which detect that the binary executable has been modified, debuggers have been attacked or environments have been rooted. Software protection creates a safe zone within which to repair weaknesses and defects in software. When patches are released, adversaries rush to perform differential analysis to compare the new release with the previous version, often being able to pinpoint security updates in minutes. Fixes reveal weaknesses in the earlier code that can be exploited. Preventing differential analysis resets the "effort clock" for the adversary, providing time for the new release to safely roll out upgrades across the operational system.

As each solution is unique, so too are the exact defensive blend of software protection techniques that can be applied to harden each design. The application of software protection technologies, specifically to sensitive areas, hardens the software in the twin and makes it exceedingly difficult for a hacker to use as a blueprint, as well as making the twin software more difficult to modify without being caught.

Finally, there are techniques that can lock both the software and data to specific devices (computers) by using various types of data and copy protection technologies (such as whitebox cryptography) and hardened APIs. The end goal is to render the software inoperable and/or to ensure that the data is inaccessible if the software and/or data is copied to another machine, thereby preventing propagation of the twin implementations between devices. Technologies such as these necessarily have additional management overhead but

manage attack risks against digital twins effectively.

Software protection techniques ensure that your software is not the target of an attack and enables digital twins to operate in higher-risk edge environments where they can operate closer to the real system they are monitoring. To ensure they implement these stages effectively, industrial and manufacturing organizations must understand the scope of their current risk, ask hard cybersecurity-centric questions to vendors and work with trusted advisors to safely embrace connectivity in their manufacturing process.

Securing the Future

While these steps are specifically targeted at securing systems from the ground up, it is not too late for industrial and manufacturing organizations to implement good software lifecycle management, design, test and software protection to secure their existing software and systems. For organizations which are increasingly implementing connectivity into their infrastructure and supply chains, it is crucial to have a cybersecurity strategy in place to protect critical software and data. The hack resistant properties that software protection offers can provide an important defense-in-depth component to the overall security solution. Even further, with a modular approach, software protection techniques can be added in over time, minimizing single-release impact and ensuring that the security "bar" is continually being reset and advanced against the hacker.

Understanding and planning for software protection threats associated with digital twins will help ensure you transform the business risks of perimeter security into a more extensive defense-in-depth security strategy which targets cybercriminals where it hurts them most: by breaking their business models and leaving you to reap the benefits of your digital twin innovation.

➢ Return to IIC Journal of Innovation landing page for more articles and past editions