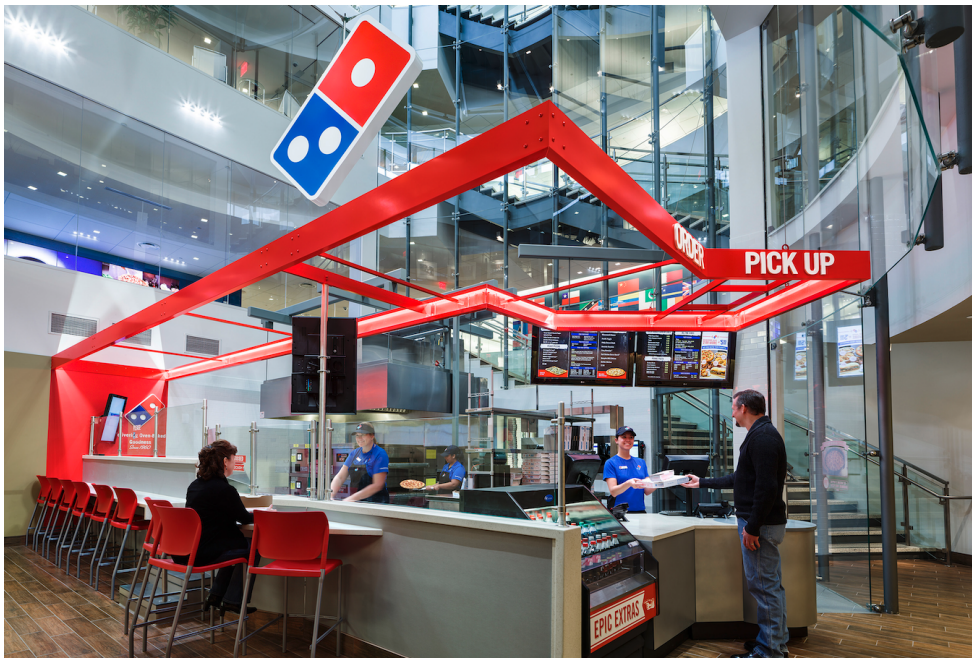
Delivering Enterprise IoT Solutions Securely: The Domino's Pizza Story

Deral Heiland - Principal Security Researcher (IoT),Rapid7
Joe VanOosterhout - Senior Manager - Store Technology Development, Domino's Pizza, Inc.



Introduction

Increasingly, businesses around the world are embracing smart technologies to underpin innovations that enhance safety and productivity in every part of our lives, from industrial systems, utilities, building management to various forms of business enablement. These technologies offer enormous benefits, and as with any new technology, they also introduce the potential for unintended consequences due to technical issues or manipulation that may not yet have been uncovered and mitigated. The very purpose of IoT technologies is to bridge the divide between our virtual and physical worlds, and as such, technical manipulation or failure has the potential to result in loss of privacy, availability of systems, and in some cases even physical harm.

Purpose

The goal of this research project was to investigate, identify, and document security practices, vulnerabilities, risk, and business processes that pertain to enterprise IoT technology solutions deployed within a business environment, covering the following three main areas of inquiry:

- Acquisition, implementation, and deployment
- Technology and functionality
- Management and support

The first part of this research project aimed to build a thorough understanding of current methodologies and best practices currently used to acquire, implement, and deploy IoT and emerging technologies within the enterprise environment.

The second part of this project was focused on gaining a solid understanding of the technology and functionality of the systems being investigated. This included testing of the technology's security to identify vulnerabilities and risk throughout the product's ecosystem.

The third and final part of this project concentrated on how this smart technology is managed and supported to reduce risk and mitigate future and unknown vulnerabilities that may be encountered.

In order to build a thorough understanding of these key areas related to enterprise IoT technology, Rapid7 partnered with Domino's Pizza, Inc.[1] Domino's is a multinational pizza restaurant chain founded in 1960, headquartered at the Domino's Farms Office Park in Ann Arbor, Michigan. Domino's is the perfect example of a large enterprise business that leverages IoT technology for business enablement on a regular basis. Domino's historically has a robust and mature IT and security program that helps facilitate the implementation of large IoT solutions. The goal of this paper is to share knowledge learned from evaluation of an IoT solution from acquisition through deployment and management across the industry to help other businesses learn from the best-in-breed processes and practices at Domino's.

The Domino's IoT-based ecosystem solution evaluated during this project is referred to as Flex. Flex is a platform-based solution that consists of various small services, which allows stores to leverage various web experiences and digital products on a variety of kiosk screens in their stores. These are purpose-built, Domino's-specific products that Domino's store team members can leverage at will. The platform powers all in-store screen technology, allowing Domino's stores and team members to be more efficient and situationally aware, so they can more efficiently and effectively run Domino's stores. The platform also provides a centralized cloud managed platform with Domino's hosted experiences, which provides Domino's stores and team members with the flexibility in technology they need to make the stores efficient and successful.

In the following sections, we will define and discuss multiple processes and procedures related to product acquisition, deployment, and management of the Flex platform solution, along with an evaluation of a product's security and methods used to evaluate the solutions IoT technologies.

[1] https://biz.dominos.com/

Acquisition Phase

What are common best practices when it comes to acquiring IoT technology for a complex deployment? To help address this question, we will discuss some of the various components related to the acquisition process executed by Domino's, with a more focused look at how information security played into a few of those processes.

Again, the acquisition phase is one of the most critical areas when thinking about IoT security, but sadly, security is not often considered during this phase, nor done effectively. The acquisition process, if done correctly, can help many organizations avoid problems down the road. Some of the key components of an effective acquisition process[2] include:

- Acquisition project plan
- Request for proposals (RFP) creation
- Vendor security risk assessment
- Solution security evaluation
- Contract legal agreement language
- Acquisition execution

Acquisition Project Planning

Any large-scale project, and especially a solution that will support new emerging technologies and cover a broad geographical area like Domino's in-store order management and tracking solution does, should embrace a comprehensive plan. This plan should start with a project team and encompass the following key areas:

- Assemble acquisition team
- Review organization acquisition process
- Develop preliminary scope, budget, resource, and schedule
- Develop supplier qualification and selection process
- Generate RFP(s)

---

2 https://us-cert.cisa.gov/bsi/articles/best-practices/acquisition/building-security-into-the-business-acquisition-proces

When this project was initially put together, a team was formed that also would oversee the acquisition process. This team consisted of the following organizations from within Domino's.

- INFOSEC
- Store Technology
- @Retail Flex Development
- Domino's System Architecture

As noted in the list above, the security team was a key component of this process from the very beginning, a reminder that any and all projects engaged by an organization should have security at the foundation. Also, it is important to note that introducing new technologies into a project may also drive the need for new security requirements that address gaps in governance. That's exactly what Domino's did.

The role of the security team within a project is to help define key security expectations and requirements and ensure these are consistent with the organization's security policies. A security team's involvement should start from the very beginning of the project and include helping develop the preliminary scope and overseeing the validation of the security requirement. Consequently, as Domino's Flex project scope was being developed, the security team determined the following minimum requirements were must-haves:

- **Stolen Device Process:** The stolen device requirement helps ensure Domino's teams can use remote management to quickly lockout and delete all content on a Flex device if they believe it has been stolen. This feature was tested during this project and is discussed later in this document.
- **Single Sign-On and Multifactor Authentication:** Single sign-on (SSO) is an important authentication service mechanism that allows the end user to only need a single set of credentials to gain all needed access. For in-store order entry and management systems such as Flex, this helps remove the complexity of authentication and reduces the risk caused by using multiple account passwords. Combining industry proven SSO solution with multi factor authentication (MFA) adds an

extra protective layer that greatly enhances the security of the in-store systems and reduces the risk of compromise via authentication attacks.

- **Operating System and Hardware Encryption:** Hardware-level encryption is one of the best methods for encrypting data that may be stored on a device. Hardware encryption separates the cryptographic process from the operating system (OS), so even if the OS is compromised, the encrypted data still may remain protected. Often, hardware encryption in IoT technology is done by using a built-in Trusted Platform Module (TPM). Later in the paper, we examine encrypted data and the use of devices that leverage a TPM module.
- **Secure Two-Way Communication:** Ensuring that all protocol communication leverages some form of industry-standard encryption for exchange of data is always a top priority. Typically, when secure communication is needed via web services and remote desktop protocol (RDP), Transport Layer Security (TLS)[3] is used to maintain this point-to-point encryption.

The above security requirements were built into the RFP and properly executed and validated to ensure the project maintained the highest level of security.

Vendor Security Risk Assessment

Any modern enterprise-level IoT initiative will likely require one or multiple vendors at some point, and the Domino's project was no exception. Whenever a project requires external vendor services — including giving the vendor access to the organization's network directly or via VPN, or contracting them to manage resources or corporate data — it becomes critical to conduct a vendor security risk assessment. These processes should also account for supply chain risk. To address these requirements at a high level so other organizations at this stage can understand the value of this process, we have listed the following key areas of this process for consideration:

- **Vendor Background Check:** In all cases where a vendor or partner is considered, a business-to-business background check should be performed.

[3] https://tools.ietf.org/html/rfc5246

- **Business/Regulatory Risk Determination:** Categorize each of your vendors based on the criticality of services they perform for the business. Are their services critical or non-critical? What is the level of risk to the organization?
- **Identify Inherent Risks:** Inherent risks are being addressed in the external vendor questionnaire. This document could serve as a guideline for many different vendor types, particularly around stability (strategic) and credibility (reputation), as detailed in the document.
- **Risk Mitigation Strategies Based on Criticality and Residual Risk:** The quantity of inherent risk the vendor carries is determined by evaluating their criticality, access to sensitive data, reliability, and scalability.
- **Review and Re-evaluate:** Re-evaluation of the vendor's status should take place on a routine basis. The nature of risk means that it must be frequently verified and reconfirmed to ensure it stays within risk tolerance thresholds. Often, vendors hold certifications, such as SOC2 or ISO27001, to help assure their customers that the required controls are in place to protect their systems and data and to show their diligence in these areas.

Solution Security Evaluation

When looking into acquiring any IoT solution, organizations should consider more than just the vendor's word and the sticker price. Once you've narrowed down a series of solutions between potential vendors based on technical and security requirements, the next step is to test the products internally or use an external service to validate that they meet the organization's base requirements and security expectations.

Expecting the hardware platform and underlying operating system to be one of the most critical components of the Flex solution and the most significant part of the acquisition process, Domino's evaluated a number of potential embedded solutions, including:

- **iOPEN:** iOpen is MAC-enabled open standards systems.[4]
- **Windows 10 IoT:** The latest Windows operating system designed to run on embedded hardware.[5]

[4] https://iopen.net/
[5] https://developer.microsoft.com/en-us/windows/iot/

- **ChromeOS:** ChromeOS is a Google-designed operating system derived from Gentoo Linux.[6]

While evaluating these various solutions, Domino's focused their evaluation across several important traits, including supportability, manageability, security, and cost. Of all of the solutions examined and tested by Domino's team, the ChromeOS by Google met all of those needs. Google Workspace allowed for this technology to be remotely manageable with ease of use. The underlying hardware requirement, ChromeBox devices, was easy to maintain and deploy. Also, ChromeOS and associated Chromebox technology provided a robust security architecture that also included hardware-based encryption. The Chrome solution and Chromebox hardware helped Domino's meet the must-haves defined in the RFP developed during project acquisition planning.

To conclude the acquisition section, we want to point to a few key security measures we learned from the process and the challenges Domino's faced during this phase of the project.

- Building security into the process starting with planning and acquisition phases is critical to any project's success and its future supportability.
- When planning out your project, always consider vendor security in your risk planning and modeling.
- When planning and defining those security must-haves, make sure they are mapped properly to your organization's security policies.

Details like right to assess and breach notification should be incorporated into the contract/legal agreement with the vendor to avoid conflicts if any related security issues or damages occur.

Design and Implementation

Beyond the planning and acquisition phase, the design and implementation of a new technology solution creates a high level of security risk for an organization. Even though we may have a complete design done early on, once we move into the implementation phase, we may need to make changes

[6] https://www.google.com/chromebook/chrome-os/

to production network infrastructure, firewall configurations, and software applications to support the new solutions.

In this section, we will address Domino's effort as it relates to the security, design, and deployment of the Flex solution, covering at a high level the components that all organizations must consider when deploying a solution.

First and foremost, an organization needs to look at their current security control processes and compliance needs and determine how the new solution will map to the security and controls plan without negatively impacting the current security posture. In Domino's case, their security control solution is a mix of both Center for Internet Security (CIS) Controls[7] and National Institute of Standard and Technology (NIST) Special Publication 800-53[8], with the main focus encompassing the CIS controls. Working through this project with Domino's, we quickly saw that the CIS controls could be mapped to the Flex project to support a solid security posture. CIS also provides a companion manual[9] to help organizations to better map CIS controls to their current IoT projects. Here are several of these controls and how they mapped out to Domino's Flex project.

CSC1 - Inventory and Control of Hardware Assets: In the case with Domino's, inventory control of the Chromebox physical hardware, which makes up the primary embedded technology for the Flex solution, is not a number one priority. This is because inventory control and purchase of the physical hardware predominately falls under the control of the individual franchised stores. With that said, tracking inventory is still maintained because of the one-for-one relationship between the physical hardware and the installed Chrome OS software, which Domino's manages via Google Workspace. Google Workspace ensures proper inventory control and management of the OS license, which helps to prevent unauthorized devices from gaining any level of information access to the Flex ecosystem.

CSC2 - Inventory and Control of Software Assets: As mentioned above, Domino's maintains a detailed inventory of all Chrome OS licensed software and manages this inventory with Google Workspace, which helps ensure the

[7] https://www.cisecurity.org/controls/
[8] https://csrc.nist.gov/publications/detail/sp/800-53/rev-4/final
[9] https://learn.cisecurity.org/iot-security-companion-201501015

deployed OS version is patched to the latest authorized and secure version at all times. Domino's manages other critical software packages — including off-the-shelf, internally generated and open-source software packages — with an internal software design and lifecycle management database.

CSC3 - Secure Configuration for Hardware and Software on Mobile Devices, Laptops, Workstations and Servers: To meet this CIS control, Domino's supports an internal software design and lifecycle database control systems, which requires tracking and management of all internal deployed software, including internally generated, open-source, and library software packages in use. Also, this detailed inventory is continually scanned for any known vulnerabilities and security issues that may arise. This allows Domino's to quickly identify issues within deployed software so they can mitigate issues as quickly as possible.

CSC4 - Continuous Vulnerability Assessment and Remediation: This security control underscores the importance of conducting regular security assessment against IoT ecosystems like Flex. Domino's understands this and conducted a number of initial security tests and product evaluations before deploying them into the production Flex ecosystem environment. Coming out of this initial solution deployment, Domino's also contracts to external services on a regular basis for more detailed follow-up and regular security testing to ensure that this CIS control is applied properly.

Besides the above discussion on the CIS controls used within Domino's, we also discussed another key area that helped deliver a well-designed IoT solution that met Domino's security expectations. When deploying an advanced and potentially complex solution, external services can sometimes help design and deploy the technology at the highest security level. Domino's took advantage of this concept by partnering with expert services from Google. Google integration partners helped the Domino's team build a baseline configuration and security-hardened it to industry best practices that also mapped to Domino's internal policies.

To conclude the implementation section, we want to point to a few key security features that we learned while designing and implementing the Flex solution.

- Any organization working on deploying an enterprise IoT project of any scale should leverage CIS's IoT companion guide to help with CIS controls as they apply them to IoT.
- Never assume you have all the answers. Often, best practice dictates taking advantage of external professional services to help refine and improve on your solutions security posture.
- Leverage internal and external services to conduct regular security audits and testing of your deployed solutions.
- Pick a security and control model, and map your project to that model to ensure you've properly covered all your bases. This will help ensure your organization maintains a solid security posture.

Technical Product Review

Product security assessment and testing are always critical parts of the acquisition process. On top of these, organizations deploying a complex enterprise IoT solution should regularly conduct assessment of that solution. This testing should be a fully comprehensive evaluation of the entire product ecosystem.

As part of this research project with Domino's, Rapid7 conducted a full ecosystem test of the deployed enterprise IoT solutions. During this testing, Rapid7 followed their current published methodology[10] for testing the IoT ecosystem.

This ecosystem consists of every interactive component that makes the IoT product solution function as designed. In the case with Domino's in-store order entry and management solution, this ecosystem included the following key components:

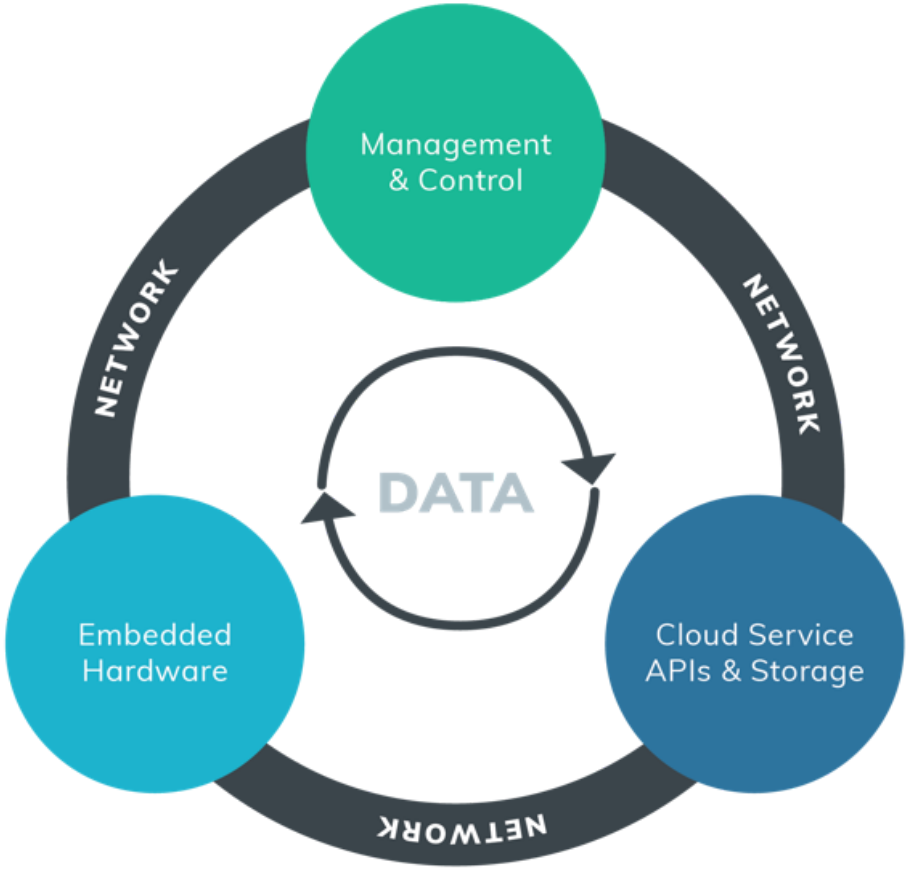
- Cloud API and web services
- Network communication protocols, used in local networks and over the internet
- Embedded hardware (Acer Chromebox CXI3)
- Radio frequency (RF) communication protocols

[10] https://www.rapid7.com/research/report/iot-ecosystem-testing-methodology/

- Backend server systems (RDP, SMB, MSSQL)
- Firewall, routers, and switches that support network VLAN segmentation

The incentive behind examining the entire ecosystem is, naturally, to ensure all components of the technology are secure (or, more realistically, secure within the constraints of a limited device). The fact is, the interconnected nature of IoT means that the security of any component in this ecosystem can, and will, affect the security of all other parts. This requires all security testing and examinations to be holistic in nature, addressing every component of the IoT products ecosystem.

The following diagram illustrates all of the logical components typically encountered within an IoT ecosystem and shows how each component interacts and converges with the others.



Embedded Hardware Testing

As part of the testing of the Domino's Flex solution, it was arranged to have a live system extracted from a Domino's store to simulate a smash-and-grab theft of one of the Flex solution Chromebox systems used within a store location. Often, this phase of testing can add a lot of insight into potential methods of access and information compromise for enterprise IoT technology. During this phase of testing, a number of critical subcomponents of the embedded technology were examined. These include the following:

- Exposed Ethernet or WiFi network services
- Interactive hardware ports (USB and HDMI)
- Extraction of configuration and application data
- Extraction of firmware and operating system information
- Non-routable RF communication
- Inter-chip communication (communications at the circuit board levels)

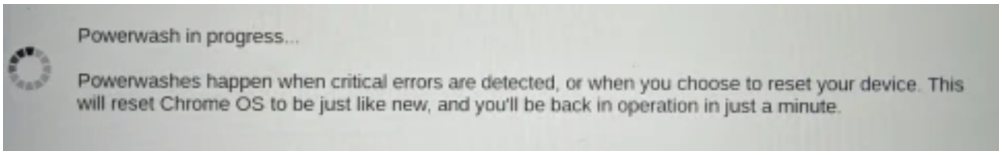
The following sections give some deeper insight into some of the tests that were conducted, along with the findings and the benefits gained from this level of examination.

The first steps taken during testing of the Flex solution Chromebox hardware that was removed from a store was to connect power to the device, but not give it network connectivity to prevent the device from phoning home and triggering a possible wipe of the data. During this initial test, without network connectivity, it was observed that the device attempted application initialization and tried to connect to the internet to update and provision the application. From this point, we were not able to gain access to any of the initially defined application functionality or configuration information. We were able to access the menu for loading other applications and the setup menu for the WiFi and Bluetooth, but no real data of value could be initially accessed, which is what we would expect from a system like this. Typically, a stolen device would be remotely wiped as soon as it was detected. This is often done by leveraging the enterprise management solution that is in place. During this series of tests, Domino's did not attempt to remotely wipe the device so that the assessment could proceed uninhibited.

With remote wipe disabled, we next connected the device to the Internet to see if further access to data was possible. Once connected, the application was able to update and provision itself. Once basic provisioning appeared to be completed, we disconnected the device from the internet, forcing the application to lose expected connectivity. Within the applications console, we were able to open a diagnostic screen that showed the current status of the application. We noticed that if the device failed to connect over the internet to its services or lost connection to the web API services or its authentication services, then at the bottom of the diagnostic window, under the 'LATEST ERROR' section, a box called 'Show more' would be displayed indicating that further system error messages were available. Clicking on this displayed more detail of the application's current error state and revealed further data, including URLS, API keys, OAuth Secrets, and hardcode password information. These types of verbose error messages are very common when testing web application services and APIs and can often be enumerated on a client host when system errors or loss of network connectivity occur. Often, these verbose error messages also appear when debug services are left enabled or system error engines are not configured to deliver generic responses or obfuscate critical data within the message.

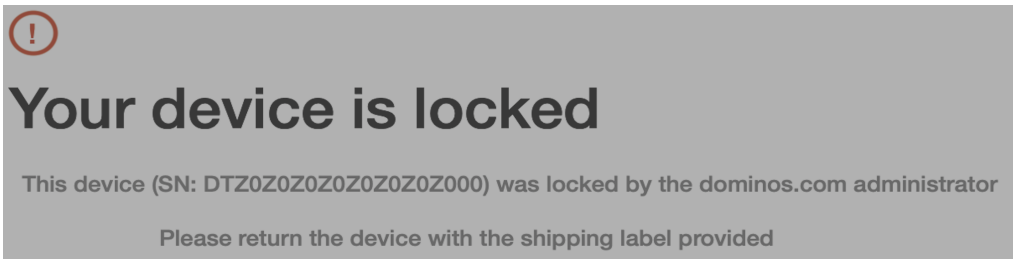
To resolve security bugs that are identified within an application, details are then fed into the Domino's software development lifecycle and analyzed to determine the proper architectural approach to resolving identified security bugs, such as those discussed above. Once the architectural approach is approved, it next enters the development phase followed by quality assurance testing. It's then deployed into small test groups to ensure no unknown issues arise during operations. Once all proper testing is complete, a solution then can be deployed. Deployment is done as rapidly as possible and necessary, and it's always weighed and balanced against other priorities based on their importance and the criticality of the identified security bugs.

Beside the verbose error messages, which Domino's was able to quickly mitigate, this test also allowed us to validate the benefits of configuring a solution, such as a store kiosk system, to be remotely manageable. Remote management allows Domino's to wipe a stolen device the minute it connects to the internet and phones home. To validate this process, we had Domino's support execute their documented stolen device procedure and trigger the remote wipe of the device via Domino's Google Workspace management console. As soon as the device authenticated to the cloud services, the remote wipe operation was triggered and wiped the device's storage, un-provisioned it, and deallocated the assigned license. An example of the Powerwash process triggered on the device to wipe data and reset the operating systems is shown below:



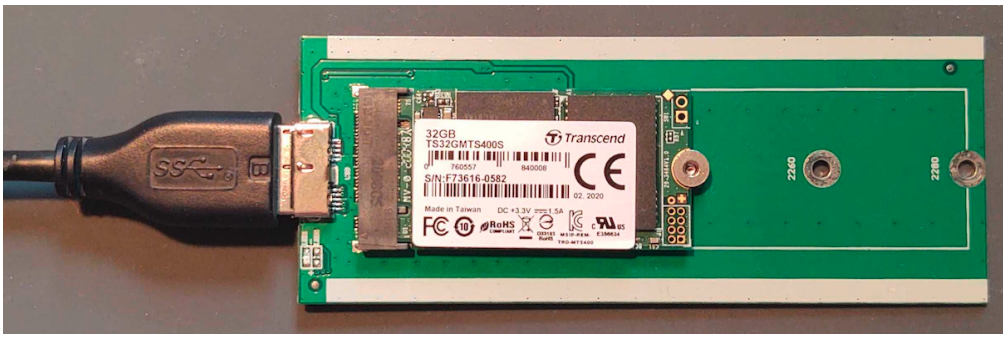
This operation proved effective in removing Domino's proprietary data from the device, but once completed, the device was left in an unlocked and unprovisioned condition. This condition allowed whoever had physical control of the device to reconfigure the device for personal use. This condition was found to be problematic. So we next conducted several other reset method tests to determine the best method for wiping the device and preventing its reuse. The final process, which we found to be most effective, was when the device authenticated to the cloud services and triggered the remote wipe

operation (Powerwash). This wiped the device's storage, un-provisioned it, and locked the device, preventing its reuse by anyone outside of Dominos. An example of the locked splash screen, presented on the device after this procedure, is shown below:



With this process tested and validated, Domino's changed their documented stolen-device procedure to use this new method moving forward. This is another example of the value of regularly retesting documented security procedures. It allows for the discovery and improvement of security processes to help maintain a best-in-class security program.

Another key area we often focus on when examining embedded hardware is the internal storage of the technology. This includes examining flash memory storage for critical data and firmware. In the case of a Chromebox, the device contained a solid-state drive (SSD). In this specific case, it was a 32GB SSD. For testing purposes, we removed the SSD and made a byte-level data copy of it for offline examination and also to protect the data in case our testing led to data corruption or system wipe. We also made a copy of the data onto a second 32GB SSD drive and mounted it in a USB reader, as shown below:

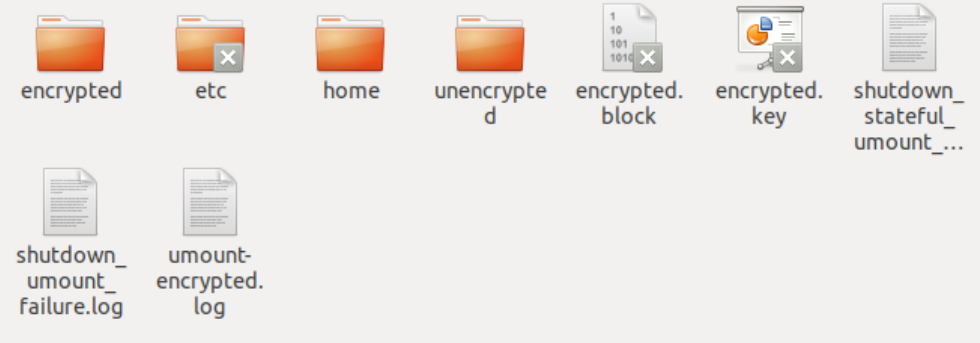


Using the SSD mounted in the USB reader, we examined the device for

formatted partitions. During this process we identified 12 partitions with the following partitions being easily mountable on a Linux operating system:

- EFI System
- OEM
- ROOT-A
- ROOT-B
- Filesystem

We mounted each of the mountable partitions individually and conducted an in-depth examination in an attempt to gain access to information that a malicious actor could use to further compromise the systems. An example of the mounted Filesystem partition we examined during these tests is shown below.



During this test, we found all user data stored within an encrypted section (encrypted.block), which is only accessible if the user's password is known. This is a well-known security method deployed within ChromeOS systems. It is also important to note that this data is also protected by a Trusted Platform Module (TPM)[11], which can only be disabled if the device is placed into developer mode. Developer mode also has a protection method, which will also wipe all data from the system when it is enabled on the device for the first time. This prevents malicious actors from using developer mode to bypass enabled system security. As an added security feature, a device that has been placed into developer mode can be prevented from being enrolled into Google Workspace. It's also possible to set a system policy that prevents a previously enrolled device from being placed into developer mode. When researching

[11] https://www.chromium.org/developers/design-documents/tpm-usage

methods to extract and/or decode the encrypted user block, we also found a documented forensic process[12] that documented a recovery process if the user's password was known. During this testing, the username and password was not known, which prevented us from using this method. This also points out the critical nature of ensuring that the local user's account password is complex and secured when leveraging Google Chrome as a managed solution.

In general, a few of the most important takeaways from conducting embedded hardware testing consist of identifying security issues within the products and validating if the deployed solution meets best-in-class security. Security issues, when found, can include everything from deployment or misconfiguration errors to product vulnerabilities. Also, once the issues are identified, an organization can then take action to mitigate and, if needed, report them to the vendor so they can take needed action.

Cloud Service and API Testing

When testing cloud web and APIs services, testing should focus on the OWASP[13] top 10 vulnerabilities, including at minimum the following areas:

- Injection vulnerabilities
- Broken authentication, session management, and access control
- Sensitive data exposure
- Security misconfiguration issues
- API business logic attacks

During API testing, besides using various open-source tools, we used the application BurpSuite[14] to assist in the testing process. This tool helped us capture, replay, fuzz, and conduct injection attacks against the APIs and web services. On a positive note, no serious issues were identified during this testing. This lack of findings can be attributed to two key areas. First, Domino's has implemented an effective security program within their organization, which requires regular assessment and security testing against services that are exposed to the Internet. This has helped Domino's maintain a very positive security posture. Second, Domino's web and API services are hosted on

[12] https://dfir.pubpub.org/pub/inkjsqrh/release/1
[13] https://owasp.org/www-project-top-ten/
[14] https://portswigger.net/burp

Akamai Technologies web hosting[15]. Akamai's services add an additional security layer, which detects and prevents a number of injection attacks from hitting Domino's web and API services.

Management and Control

To gain a broader insight and view of the Flex in-store order entry and management solution, we went on-site to a store environment to conduct a series of tests from the inside perspective. The in-store location we used was Domino's innovation garage, which is located in Ann Arbor, Michigan, and is a perfect match of a typical Domino's store location.

The purpose of going on-site was to ensure we used a fully functioning ecosystem for testing. This is a critical part of testing any IoT solution and helps avoid the fragmented nature of testing standalone IoT components. In this environment, we can test the impact of security issues found in one segment of the ecosystem with the others and also better define existing security controls.

While on site, our security testing effort had access to a fully functioning ecosystem. This allowed us to conduct a more focused test on several critical areas within the ecosystem, including:

- Ethernet Network
  - Exposed services
  - VLAN segmentation

[15] https://www.akamai.com/

- WiFi 802.11
    - Business WiFi
    - Guest WiFi
- Kiosk security protections
    - Device break-out attacks
    - Information leakage attacks
- Back-end host services
    - Exposed services
    - Access control

While examining the in-store locations, we ran a number of tests on the network that focused on understanding segmentation protections. Segmentation testing is critical to identify the touch points between in-store systems, the Internet, and any back-end systems located on Domino's internal networks. Although in-store networks rely on physical security to protect the environment from unauthorized access, it's still essential to maintain secure network segmentation. In this case, we found segmentation for the most part to be well maintained. Only systems that were part of the Flex solution ecosystem were accessible from the appropriate VLANs. However, we did identify that several TCP ports were accessible beyond what was needed for Flex solution to properly operate.

The Flex solution is reasonably recent to Domino's, and legacy systems still exist at some store locations. To avoid the confusion of having to maintain both Flex and legacy network configurations, which can become problematic, Domino's decided to maintain a single network configuration that supports all in-store solutions until any current legacy deployments are retired. At that point, Domino's will roll out an updated configuration to remove legacy network requirements.

While in the store environment, we also had the opportunity to examine various Flex solution kiosk terminals from a user's perspective. These kiosks fell into two general categories. First, was the customer-facing kiosk. These do not have access to in-store networks or back-end processing services. Instead, they only have access to cloud services for displaying current customer order

status, and they only allow for limited user input. This level of connectivity greatly reduces their attack surface.

The second class of kiosks are store processing kiosks, such as the order entry kiosks that are typically located at the store's front counter. Since the customer-facing kiosk lacks advanced access and also has limited functionality, most of our testing targeted the order entry kiosks. These kiosks typically are only accessible by store employees, not by Domino's customers. We attempted to bypass kiosk security settings and see if, given time, it would be possible for a malicious actor to gain access to the underlying operating systems and run commands.

While evaluating the order entry kiosks, there were several interesting entry-point Windows applications that we believed an attacker could use to gain deeper access to the device. During this testing, we launched the Windows applications and used common application macro creation methods to inject code, launch commands, gain access to the underlying OS, and execute commands. On a positive note, these applications were configured to prevent access that allowed the execution of macros.

To conclude this section, we want to point to a few key security measures that should always be considered when conducting a security assessment like the one we performed against Domino's Flex solution ecosystem.

- Network segmentation is a key component of a holistic security plan. Devices that have no business requirement to communicate with each other should be restricted from doing so whenever possible.
- When testing the security for a new technology, use a holistic approach that targets the entire solutions ecosystem and accounts for the interaction of the various components.
- Conduct regular testing of documented security procedures. Security is a moving target, and testing these procedures regularly can help identify deficiencies. This allows you to make needed adjustments or replace procedures to maintain a continued best-of-breed security posture.
- Don't overlook deployed technologies that may expose internal information or allow access to system resources, like a customer kiosk.

A compromised device can often have serious security implications. Take those extra steps to harden these devices and ensure they are tested properly.

Management and Support

Once a solution has been designed, acquired, and deployed it moves into the support phase, where we ensure that it continues to operate and meets the business's needs. Outages and security incidents that could impact production, loss of services, or loss of data need to be avoided. To do this successfully, an organization must build a solid, well-thought-out management and support infrastructure. While examining Domino's applied management and support solution for the Flex environment we quickly saw that Domino's enterprise approach to management and support was well-structured and capable of handling the needs for the Flex solution.

Patch Management

Patch management for many organizations often overlooks embedded appliances. But in the case of Chromebox devices used with the Flex project, patch management is ingrained into the Google Workspace management solution. This lets Domino's identify deployed version levels and deploy new updates remotely. Also, Google supplies updates on a regular cadence, allowing Domino's to structure an effective patch management cycle.

Within Domino's scheduled patching cycle, they conduct quality assurance (QA) testing, then pilot the changes within a small production test group prior to rolling out updates to the masses. QA testing followed by piloting is an imperative process when deploying changes within a large ecosystem like Domino's stores. It helps them avoid the cure being worse than the disease and potentially causing adverse impact on the production environment. Once the patches are scheduled for deployment, Google Workspace allows Domino's to track the deployment and generate status reports to monitor success. This helps them manage and make changes to improve the process for future deployments.

Another important aspect of making changes within a production environment is having a change management process, which helps an organization

effectively roll out and adopt the changes that need to be deployed. For Domino's, this process included a change management team whose purpose was to evaluate the requested changes to ensure they met business needs and didn't conflict with other scheduled changes, impact security, or cause unnecessary outages. Creating and deploying an effective change management process should be a priority for all organizations of nearly any size and complexity. Domino's change management process was well-defined and implemented, and it integrated well with their patch management process for the Flex ecosystem.

Risk and Vulnerability Management

One of the most fascinating observations I made while working with Domino's during the review of the Flex environment was the company's ability to leverage Open-Source Intelligence (OSINT) within their security program. Leveraging open-source information on the internet to determine potential risk to your organization is critical.

For example, let's say there is a vulnerability in the ChromeOS product, which is the core operating system of the Flex embedded hardware solution. OSINT can help Domino's detect and identify discussion or news on proof-of-concept exploit code, as well as real-world attacks against organizations with malware or ransomware created from the vulnerability. This lets them quickly identify an increased risk or probability that a vulnerability has been weaponized in the wild. With that data, Dominos can take a more proactive approach by using compensating controls to further mitigate a potential attack against them or leverage their structured patch management process to speed up patch deployment to fix the issue.

Logging and Monitoring

With over a decade of consulting experience, I've regularly encountered organizations that overlook key components that should be key elements of any robust management and support plan, such as logging and monitoring. I was pleased to see Domino's did not fall into that category. Leveraging their internal configured resources — including Google Workspace logging, active directory logging, and firewall logging — combined with security information and event management (SIEM) tools, Domino's constructed an effective

solution for security monitoring of the full Flex ecosystem, from hardware and internal data services to cloud, API services, and network communication. This solution helps Domino's security operation center maintain security oversight of the Flex solution, allowing them to better detect potential incidents and take action as needed.

To conclude this section, we want to point to a few key security measures we learned while examining key components of Domino's Flex solution's management and support.

- Implement or acquire an OSINT solution that gives your organization proactive insight into potential risks on the horizon. This will allow your security teams to mitigate potential issues before they impact your organization.
- When deploying new embedded technology within your enterprise environment, make sure the technology is properly integrated into your organization's patch management, logging, and monitoring processes.
- It is critical at this stage to identify what actionable security events an analyst should dig into. Typical, out-of-the-box SIEM alerting will not generally support IoT and embedded-device monitoring needs (for security or operational reliability).

Conclusion

As we close this review of our research partnership examining the acquisition, deployment, technical, and ongoing support and management of the Domino's deployed Flex solution, we are pleased to finish up with a quick discussion of the vulnerability disclosure program (VDP). Having a structured VDP in place should be the goal for all organizations and demonstrates a high level of security maturity. A properly structured and implemented VDP forms a process for security researchers, customers, and partners to report security issues and vulnerabilities to the organization. This allows the organization to effectively receive and process critical security data from any external source and remediate those security issues in a timely manner. Domino's VDP includes a guided process to submit in-scope findings, validation of the findings, and internal tracking to resolution. This is all handled via the online portal at https://dominos.responsibledisclosure.com.

Organizations looking to build out or improve their own current VDP can leverage the following respected resources: ISO 29147[16](Information technology—Security techniques—Vulnerability disclosure) and ISO 30111[17] (Information technology—Security techniques—Vulnerability handling processes).

[16] https://www.iso.org/standard/72311.html
[17] https://www.iso.org/standard/69725.html