Secure Camera Platform Architecture Review & Threat Model

ExampleCorp

August 17, 2021

Prepared for
Eugene Belford

Prepared by
Dade Murphy
Kate Libby

Executive Summary

ExampleCorp engaged NCC Group to conduct a security architecture review of the Secure Camera platform. The objectives of this project were as follows:

- Document all of the individual components (even third-party components/partners) that make up the system
- Learn how those components work together to make the application function in as much technical detail as possible in the time allotted
- Enumerate assets of the system (e.g. valuable data or access to sensitive components)
- Enumerate and evaluate existing security controls that are in place to protect those assets
- Identify potential threats by legitimate users, employees, related third parties, and external attackers
- Uncover any design defects or security gaps that exist with the system design
- Make specific recommendations for defect remediation or product design security improvement

For this project, NCC Group reviewed documentation provided by ExampleCorp, and interviewed several subject matter experts onsite at ExampleCorp's facility in Dallas, TX. The topics covered during this review included:

- Overall system architecture
- Mobile application design
- Secure Camera Cloud overview
- QA testing
- SDLC gates and processes
- Backups, including retention and access control
- System patching and updates
- Usage of third-party libraries
- Logging and monitoring
- Business risks

After gaining an understanding of how the system works, NCC Group documented the application message flow, and identified specific assets, threat agents, and attack vectors. This process is documented in Methodology on page 4.

Based on this, NCC Group identified the following potential design or process flaws:

- Sensitive Data Exposure Through Storage API (finding NCC-ExampleCorp001-003 on page 12, finding NCC-ExampleCorp001-008 on page 20)
- Sensitive Production Data Used in Staging (finding NCC-ExampleCorp001-001 on page 15)
- Revenue Generating Advertisement Delivery can be Bypassed (finding NCC-ExampleCorp001-006 on page 23)
- Auth Token Passed Over Unencrypted Communication Channel (Web Services) (finding NCC-ExampleCorp001-007 on page 19)
- Use of Shared Administrator Credentials (finding NCC-ExampleCorp001-002 on page 16)
- Firmware or Bootloader Downgrades Not Prevented (finding NCC-ExampleCorp001-004 on page 17)
- Firmware Integrity Validation Flaw (finding NCC-ExampleCorp001-005 on page 18)
- Application Uses Weak Hash to Store Passwords (finding NCC-ExampleCorp001-010 on page 21)
- Sensitive Information Stored in Logs (finding NCC-ExampleCorp001-012 on page 14)

These potential vulnerabilities are discussed in more detail in Table of Findings on page 11. It is important to note that this information is a result of interviews conducted with employees of Secure Camera, and has not been explicitly confirmed by NCC Group through manual testing or source code review.

NOTE: This sample report is intended to provide a generic illustration of NCC Group deliverables. Every assessment of this type is tailored to the customer, and specific project parameters. The product, system architecture, interview notes, and vulnerabilities mentioned throughout this document are fictitious. Technical references and inter-relationships may or may not be consistent.

Dashboard

Target Metadata

Name	Secure Camera
Type	Security Architecture Review
Platforms	Java
Environment	Production

Engagement Data

Type	Security Architecture Review
Method	Security Architecture Review
Dates	2019-08-04 to 2019-08-23
Consultants	2
Level of Effort	30 person-days

Finding Breakdown

Critical issues	0
High issues	3
Medium issues	8
Low issues	2
Informational issues	0
Total issues	**13**

Category Breakdown

Access Controls	2
Auditing and Logging	1
Authentication	2
Configuration	1
Cryptography	2
Data Exposure	2
Data Validation	1
Other	1
Patching	1

Component Breakdown

Administration API	2
Camera	3
Camera Repository	2
Device Ownership Manager	1
Mobile Application	1
Processes/Procedures	1
User Management	3

Key

Critical | High | Medium | Low | Informational

Methodology

NCC Group performed a security architecture review of the in-scope applications and systems, tailored to the project timeframe, and focused on the following items:

- NCC Group interviewed ExampleCorp personnel to understand the target's functionality, components, assets, and security controls. Full interview notes are included in Appendix B on page 27. Stakeholders interviewed included:
  - Lead Developers
  - System Architects
  - IT Operations (for the infrastructure in which the system operates)
  - Business Line Personnel
  - Internal Security Team
- NCC Group reviewed relevant system documentation covering design, specifications, security, and deployment scenarios. Documents reviewed are listed in Appendix C on page 29, and included the following:
  - System architecture diagrams
  - Data flow diagrams
  - Database entity diagrams and data dictionaries
  - Functional specifications
  - Technical specifications
  - Use cases
  - User stories
  - Test cases
- Where possible, NCC Group conducted joint whiteboard diagramming and table-top exercises to understand expected outcomes and corner-cases.
- After completing the initial interview and discovery phase, NCC Group analyzed the system's security model to:
  - Review security assumptions and identify any gaps between them and those actually provided by the system
  - Uncover possible weaknesses in the existing design as a result of misunderstood assumptions, missing security guarantees in the underlying networks, applications, or operating systems, or insufficient specifications
  - Identify future weaknesses that could occur due to changing assumptions or deployment scenarios, and documented features and functionality that must be maintained for the system to mitigate intended risks over time
  - Describe additional features or functionality that could increase the security of the system, such as compensating controls that can provide defense-in-depth
- Finally, NCC Group documented all analysis, observations, and findings in this report, in addition to the architecture diagram and threat model. The threat model covers:
  - Key system assets to be protected
  - Significant threats to the confidentiality, integrity, or availability of identified assets
  - Most likely attack vectors that NCC Group believes would be used by attackers
  - Recommended mitigations and security features for addressing identified threats

After a thorough review of the client provided documentation listed in Appendix C on page 29, and in person interviews documented in Appendix B on page 27, NCC Group created the following architecture diagram, which details the components, trust boundaries and communication paths that make up the Secure Camera system. This diagram is based loosely on recognized conventions such as those described in The Security Development Lifecycle, Chapter 9, (M. Howard and S. Lipner, 2006). Other conventions may be used depending on project-specific requirements.

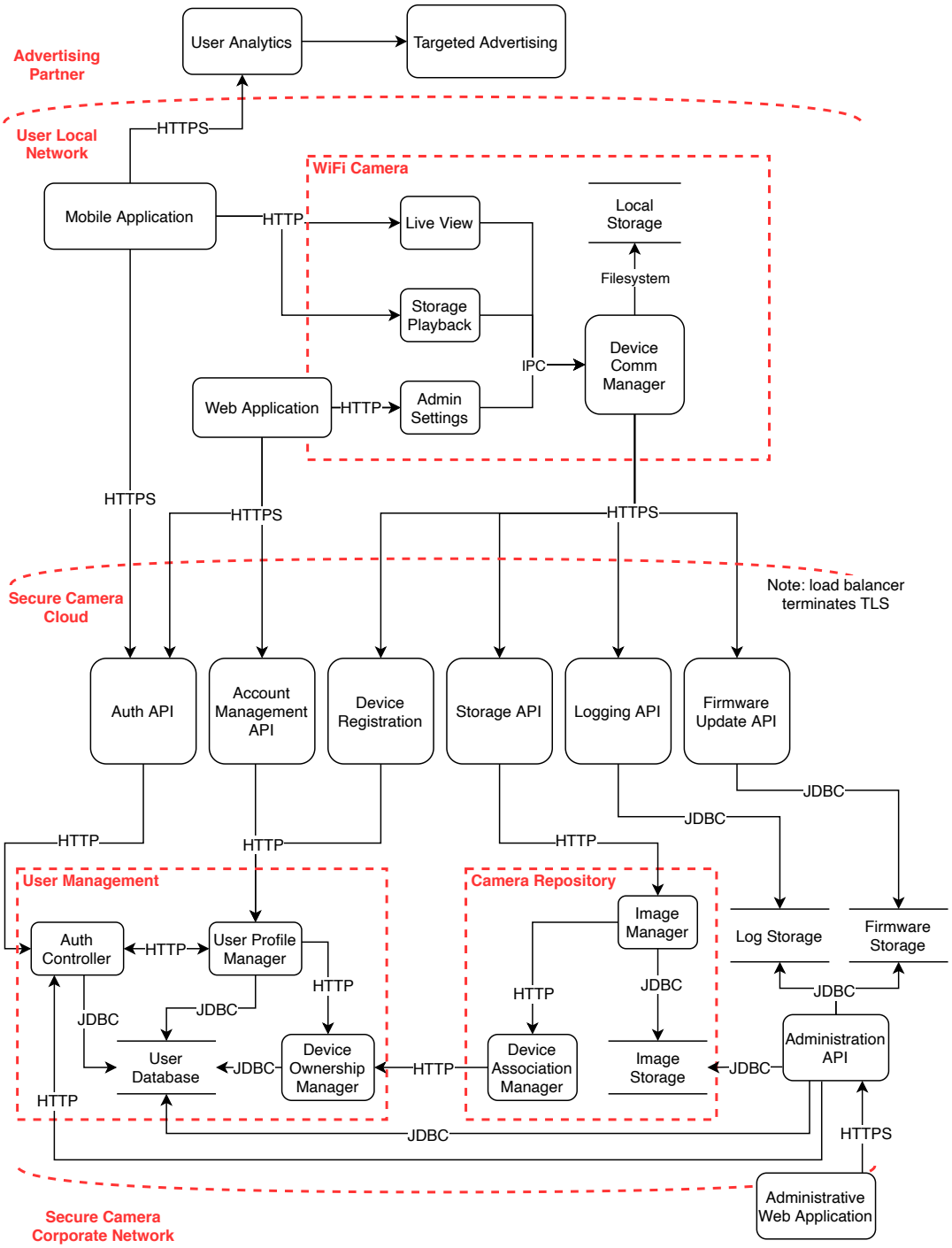

Figure 1: This is a made up system that may or may not make technical sense – for illustration purposes only

A description of each in-scope component is listed below:

User-Facing Applications

- **Mobile Application**: iOS or Android application that allows the end user to view their recorded images, or perform a live view of the camera. This is also where revenue-generating advertisements are displayed to the user.
- **Web Application**: Single Page Application (SPA) that an end user uses to create their account and register their camera with Secure Camera Cloud.

Wi-Fi Camera Hardware and Software

- **Wi-Fi Camera Package**: The following components make up the Wi-Fi Camera:
  - *Live View*: Internal LAN-only web service responsible for initiating a live view of the camera, to allow a mobile application user to view the camera live feed.
  - *Storage Playback*: Internal LAN-only web service responsible for initiating playback of recorded images, to allow a mobile application user to view previously taken photos stored locally on the camera, or saved to Secure Camera Cloud.
  - *Administration*: Internal LAN-only web service responsible for allowing an authenticated web application user to take ownership of the camera, and register the device with Secure Camera Cloud.
  - *Device Comm Manager*: Central communication component of the camera that routes user requests to local storage, or Secure Camera Cloud.
  - *Camera Local Storage*: Local 1GB storage container for storing photos and configuration data.

Internet-Accessible APIs

- **Auth API**: Externally accessible web service that handles authentication requests from the web and mobile applications.
- **Account Management API**: Externally accessible web service that handles account management requests sent by the web application.
- **Device Registration**: Externally accessible web service that handles device registration requests sent by the camera.
- **Storage API**: Externally accessible web service that handles image upload and download requests.
- **Logging API**: Externally accessible web service that handles log write requests sent by the camera.
- **Firmware API**: Externally accessible web service that handles firmware update requests sent by the camera.

Internal Systems

- **User Management**: The collection of components that make up user account management operations are:
  - *Auth Controller*: Responsible for user login, and session management operations.
  - *User Profile Manager*: Responsible for managing user account data stored in the user database.
  - *Device Ownership Manager*: Responsible for managing user to device ownership mappings in the user database.
  - *User DB*: mySQL database that is used to store user credentials, session information, and device ownership data.
- **Camera Repository Package**: The collection of components that make up the image cloud storage and playback functionality are:
  - *Image Manager*: Responsible for storing and retrieving images from the image storage database for requested devices and users.
  - *Image Storage*: mySQL database that contains camera images for all cameras.
  - *Device Association Mapper*: Responsible for mapping camera images to the appropriate devices and users.
- **Log Storage**: mySQL database that contains logs for all cameras.
- **Firmware Storage**: Filesystem that contains all firmware versions available for the Secure Camera devices.
- **Administration API**: Web service used by Admin Web Application to authenticate ExampleCorp employees, and allow them to manage the cloud data storage components. Exposed to the ExampleCorp corporate network.
- **Admin Web Application**: Web application used by ExampleCorp employees to perform administration functionality on Secure Camera Cloud components.

NCC Group has taken the architecture diagram produced in Architecture Overview on page 5 and overlaid assets, security controls, and threat agents that were identified during the interviews and document review. The resulting threat model was then used to develop specific potential attack scenarios in Threat Matrix on page 10. This sample diagram is based loosely on recognized conventions such as those described in The Security Development Lifecycle, Chapter 9, (M. Howard and S. Lipner, 2006). Other conventions may be used depending on project-specific requirements.

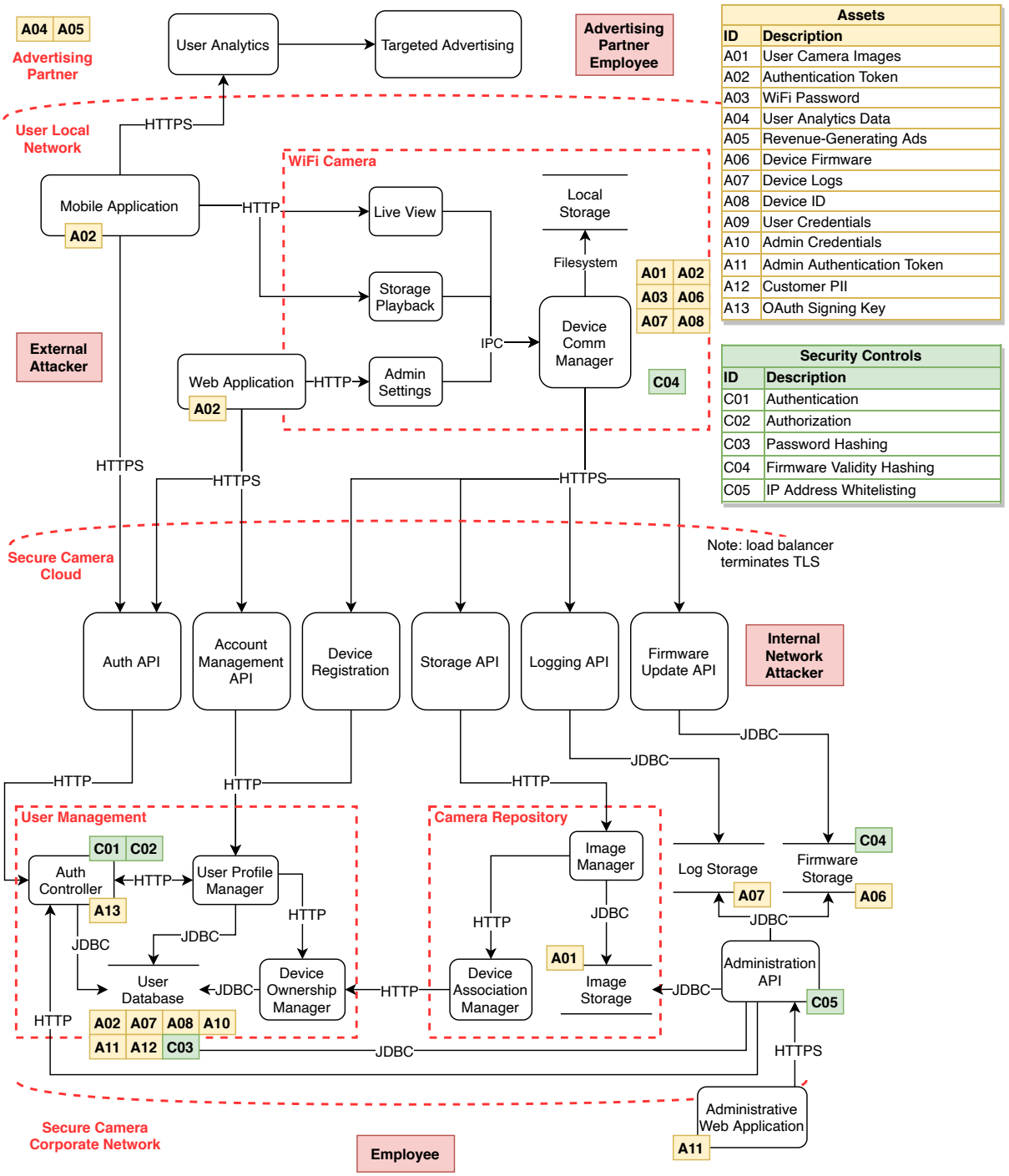

Figure 2: This is a made up system that may or may not make technical sense – for illustration purposes only

Attack Goals

The following are goals of the threat agents in this specific system.

- **AG01** - Steal user data (camera images, audio recordings, personal information) from user's local network, advertising partners, or ExampleCorp's infrastructure.
- **AG02** - Access user's local network to conduct further network attacks.
- **AG03** - Compromise ExampleCorp's infrastructure to conduct further network attacks.

Assets

Assets represent data, functionality, or an attribute of the system that a threat agent is interested in acquiring. Assets are noted in red on the diagram.

- **A1: User Camera Images**: Pictures taken by customer's cameras.
- **A2: Auth Token**: The session ID of an authenticated user or camera.
- **A3: Wi-Fi Password**: The password to the user's WiFi network.
- **A4: User Analytics Data**: User browsing history data provided to the analytics partner to help deliver relevant advertisements.
- **A5: Revenue Generating Ads**: Advertisements delivered to the user that are the sole income source for Secure Camera.
- **A6: Device Firmware**: Camera firmware packages that are installed to the device.
- **A7: Device Logs**: Debug and trace logs that the camera uploads to the Secure Camera Cloud.
- **A8: Device ID**: The identifier used to map users to specific devices.
- **A9: User Credentials**: The username and password hash of the user.
- **A10: Admin Credentials**: The username and password hash of the ExampleCorp employees/administrators.
- **A11: Admin Auth Token**: The session ID of an authenticated administrator.
- **A12: Customer PII**: The user's name, address, and email.
- **A13: OAuth Signing Key**: The key used to generate valid session IDs.

Controls

Controls are used to protect an asset from one or more threat agents. An individual control may completely protect an asset from a threat agent or the control may only partially protect an asset from a threat agent. In some cases a control will only slow down a threat agent from accessing the asset or may only deter lower skilled threat agents. Controls designed to protect an asset may fail due to vulnerabilities in their implementation. Therefore as a best practice, multiple controls should be used to protect each asset.

Controls are noted in green on the diagram.

- **C1: Authentication**: Authentication is handled by the auth controller through username and password login, and session ID validation on every API request.
- **C2: Authorization**: The auth controller is used to ensure users only access assets they have been granted permissions to view.
- **C3: SHA256 Hashed Passwords**: User and administrator passwords are not stored in the User DB in plaintext. Instead, a SHA256 hash of the user's password is stored.
- **C4: Firmware SHA256 Hash**: To help validate the integrity of the firmware images, a SHA256 hash of the image is provided to the WiFi camera along with the firmware image after a firmware update request has been received.
- **C5: IP Address Whitelisting**: Only requests originating from the Secure Cloud corporate IP range are allowed to access the Administration API.

Threat Agents

Threat agents are individuals that attack the system to either gain access to assets or disrupt the system's normal behavior. Threat agents overlaid on the diagram are defined below:

- **External Attackers**

  - **TA1: Authorized External User**: Authorized users of the system who have valid user session IDs.
  - **TA2: Unauthorized External User**: Unauthorized users of the system who do not have valid user session IDs.
- **Advertising Partner Employees**
  - **TA3: Authorized Advertising Partner Employees**: Employees or contractors of the advertising partner who are authorized to view the user analytics data and targeted advertisement data.
  - **TA4: Unauthorized Advertising Partner Employees**: Employees or contractors of the advertising partner who are not authorized to view the user analytics data and targeted advertisement data.
- **Internal Network Attackers**
  - **TA5: Authorized Hosting Employees**: Employees or contractors of the Secure Camera Cloud hosting provider who are authorized to view the environment and configuration data.
  - **TA6: Unauthorized Hosting Employees**: Employees or contractors of the Secure Camera Cloud hosting provider who are not authorized to view the environment and configuration data.
- **Employees**
  - **TA7: Authorized ExampleCorp Employees**: Employees or contractors of ExampleCorp who have valid administrator credentials.
  - **TA8: Unauthorized ExampleCorp Employees**: Employees or contractors of ExampleCorp who do not have valid administrator credentials.

Threat Matrix

Using the assets, threat agents, and controls identified in Threat Model on page 7, NCC Group has created the following threat matrix, which maps potential attackers to specific attack vectors and indicates the presence, or lack, of security control in place to prevent the attack. Attack vectors that appear to represent vulnerabilities in the system have been called out in Table of Findings on the next page.

Description	Threat Agent	Asset	Control	Note
Tamper with an authorized request to the Storage API to view images, and modify the device ID value.	TA1: Authorized External User	A1: User Camera Images	NONE	This appears to be a vulnerability, as any user with a valid session token appears to be able to modify the device ID in the request to fetch other images.
Send an unauthorized request to the Storage API to view device images.	TA2: Unauthorized External User	A1: User Camera Images	C1: Authentication, C2: Authorization	There appears to be controls in place to prevent this. However, NCC Group recommends this is thoroughly tested via dynamic analysis.
Pull the camera SD card to view the images.	TA2: Unauthorized External User	A1: User Camera Images	NONE	User images are not encrypted on the camera's local storage.
View images stored in the hosting environment.	TA5: Authorized Hosting Employee	A1: User Camera Images	NONE	The user images are not encrypted, and appear visible to anyone with access to the hosting environment data.
View the content of the Image Storage container.	TA7: Authorized ExampleCorp Employee	A1: User Camera Images	NONE	ExampleCorp administrators have access to the unencrypted user images.
View the content of the Image Storage container.	TA8: Unauthorized ExampleCorp Employee	A1: User Camera Images	C1: Authentication, C2: Authorization	Although there appears to be controls in place to prevent this, the use of shared administrator credentials poses a risk to this asset.
Session fixation permits user escalation.	TA1: Authorized External User	A2: Auth Token	C1: Authentication, C2: Authorization	The user should have access to their auth token, but not others.
Malicious employee of ExampleCorp's hosting provider accesses ExampleCorp's servers.	TA5: Authorized Hosting Employee	A2: Auth Token	NONE	The User DB is not encrypted, and appear visible to anyone with access to the hosting environment data.
An ExampleCorp employee steals the shared administrator credentials for the database.	TA8: Unauthorized ExampleCorp Employee	A2: Auth Token	C1: Authentication, C2: Authorization	Although there appears to be controls in place to prevent this, the use of shared administrator credentials poses a risk to this asset.
[example attack scenario]	TA1: Authorized External User	A3: WiFi Password	[example control]	[example note]

NOTE: NCC Group's methodology typically evaluates all valid attack scenarios to create a complete picture of all threats in the environment. This example is for illustration purposes only, and stops short of covering all threats in the above matrix.

Table of Findings

For each finding, NCC Group uses a composite risk score that takes into account the severity of the risk, application's exposure and user population, technical difficulty of exploitation, and other factors. For an explanation of NCC Group's risk rating and finding categorization, see Appendix A on page 25.

Title	ID	Risk
Lack of Authorization Controls for Storage API	003	High
Weak Administrative Credentials	011	High
Plaintext Passwords Written to Logs	012	High
Sensitive Production Data Used in Staging	001	Medium
Use of Shared Administrator Credentials	002	Medium
Firmware or Bootloader Downgrades Not Prevented	004	Medium
Ineffective Firmware Integrity Validation	005	Medium
Unencrypted Communication Channel (Camera Web Services)	007	Medium
Horizontal Authorization Bypass in Storage API	008	Medium
Application Uses Weak Hash to Store Passwords	010	Medium
Insecure Random Number Generation	013	Medium
Revenue Generating Advertisement Delivery Can Be Bypassed	006	Low
Weak Password Complexity Requirements	009	Low

Finding Details

Finding	**Lack of Authorization Controls for Storage API**
Risk	**High** Impact: High, Exploitability: High
Identifier	NCC-ExampleCorp001-003
Category	Access Controls
Component	Camera Repository
Location	Device Association Mapper
Impact	A successful exploit would result in the compromise of A1 (user camera images), and lead to a loss of user trust in the Secure Camera brand.
Description	During interviews, NCC Group was notified that the Camera Repository and Storage API use the device ID sent in the request to fetch the images that belong to the specified camera. As a result of those interviews, NCC Group believes that no validation is performed to ensure the specified device ID is associated with the logged in user's account or session ID. It appears that a malicious authorized external user could modify the device ID sent in their camera's request to the Storage API, and potentially view images from the Camera Repository that they are not authorized to view. Exploitation of this issue would result in severe privacy violations for ExampleCorp's users. If it is true that any user could modify their device ID in order to read images from other users' cameras, it would result in significant harms to those users (whose cameras have likely recorded many sensitive images) and to ExampleCorp's reputation.
Recommendation	NCC Group recommends this attack vector be thoroughly assessed during the upcoming manual penetration testing effort. The Device Association Mapper component should confirm with the Device Ownership Manager component that the device ID sent in the request belongs to the authenticated user whose session ID was presented in the original request.

Finding	**Weak Administrative Credentials**
Risk	**High** Impact: High, Exploitability: Medium
Identifier	NCC-ExampleCorp001-011
Category	Configuration
Component	Administration API
Impact	If a malicious user is able to guess or brute force the administrator password, they would get unauthorized access to all data in the Secure Camera system (A1, A2, A6, A7, A8, A9, A10, A11, A12 and A13) from the assets identified in Threat Model on page 7.
Description	Administrative accounts and internal infrastructure must take extra care to ensure attackers cannot gain access. If the interface allows access using credentials (for example, using a username and password), those credentials should be randomly generated and unguessable. Credentials that are simple, widely known, or not changed from the default values represent a single point of compromise for the application. The application has an administrative interface that is protected with weak or default credentials. An attacker with access to this interface may be able to gain access by guessing or brute-forcing those credentials, resulting in full compromise of the application.
Recommendation	Change the password to a strong, random value. Do not store the password in source code; ensure that only a small number of trusted users have access to the primary admin password. Do not use this account for day-to-day access. Instead, set up individual accounts for each user with tightly-scoped permissions, and only use the high-privileged account in exceptional scenarios. When employees leave the company, all accounts they have access to should rotate credentials.

Finding	**Plaintext Passwords Written to Logs**
Risk	**High** Impact: High, Exploitability: Medium
Identifier	NCC-ExampleCorp001-012
Category	Auditing and Logging
Component	User Management
Impact	An attacker who obtains access to the application's log files would be able to view sensitive information such as user credentials.
Description	Applications make use of logging functions to ease development, monitor application activity, or audit actions after the fact. Access controls on the log files are often more lax than those on the application itself. For example, if information that is normally stored in an encrypted form is written to a log, an attacker could access it without obtaining the decryption key. Additionally, an organization may permit internal access to log files with less strict controls than to the application's database. When creating log entries, therefore, care should be taken to ensure that no sensitive information is written, since this could provide attackers an easier path to obtaining that data. While reviewing the application's design documentation, NCC Group identified a code snippet that appears to log sensitive information. The code snippet is listed below: `logger.log("User entered username " + username + " and password " + password);` Listing 1: authentication_proof_of_concept.java line 55
Recommendation	Either remove the noted log entries entirely, or carefully sanitize them to remove any sensitive information. Once that is done, clear or sanitize the existing logs to remove sensitive data stored within.

Finding	**Sensitive Production Data Used in Staging**
Risk	**Medium** Impact: High, Exploitability: Medium
Identifier	NCC-ExampleCorp001-001
Category	Data Exposure
Component	Processes/Procedures
Impact	Cloning sensitive data into the staging environment reveals several critical assets of the Secure Camera system to Unauthorized Secure Camera Employees (TA8), and could lead to a loss of user trust in the Secure Camera brand.
Description	During interviews, NCC Group was informed that developers routinely clone data from the production databases for testing purposes into the staging environment. Generally, a staging environment is a non-production environment and should not be considered secure. NCC Group's experience has shown that non-production environments often have more lax security controls, including: • Less-strict access control policies for administration • High-privileged accounts accessible to testing staff • Shared accounts and credentials, which prevent traceability and increase the risk of password leakage • Deployment processes allowing for lower-privileged users to deploy code and enable untested, vulnerable code to be deployed
Recommendation	Non-production environments should not be considered secure and there should be minimal cross-over between a non-production environment and production data and systems. Developers should not have access to sensitive customer information stored in the production environment, including user camera images (A1), user authentication tokens (A2), user credentials (A9), customer PII (A12), or the OAuth signing key (A13). If data is needed in the staging environment databases for testing purposes, NCC Group recommends ExampleCorp develop fake or masked test data for developers to use. This data should contain no customer identifiable information, or reveal any sensitive user data.

Finding	**Use of Shared Administrator Credentials**
Risk	**Medium** Impact: High, Exploitability: Medium
Identifier	NCC-ExampleCorp001-002
Category	Authentication
Component	Administration API
Impact	The use of shared credentials removes the ability for ExampleCorp to track user activity once they are logged into the system. If a malicious insider misuses their access on the system, it will be difficult (or impossible) to determine who was responsible for the malicious act.
Description	During interviews, NCC Group was informed that ExampleCorp employees share the same username and password to login to the Secure Camera administration web application. Furthermore, NCC Group was informed that these shared credentials have never been changed.
Recommendation	Set a strong, complex and unique password for each user who has access to the administrative component. If possible, use public key authentication, with keys protected by a passphrase for authentication into the components. Ensure that the host is configured to log both successful and failed login attempts, as well as record activity performed on the hosts after a successful login.

Finding	**Firmware or Bootloader Downgrades Not Prevented**
Risk	**Medium** Impact: Medium, Exploitability: High
Identifier	NCC-ExampleCorp001-004
Category	Patching
Component	Camera
Impact	An attacker can revert to older images in order to exploit vulnerabilities that had since been fixed. Because of the lack of secure boot, it is not possible to implement effective downgrade prevention for this device at the hardware level.
Description	The bootloader does not implement a means to prevent software downgrades. It is common for attackers to downgrade the firmware or software to a version that is known to contain exploitable vulnerabilities. In fact, the release of a security patch can trigger attackers to take notice and develop an exploit for older versions.
Recommendation	Implement an effective downgrade prevention mechanism at each stage of the boot process. This could be in the form of a blacklist of "bad" software versions. The blacklist must be signed to prevent tampering and must itself be protected from downgrades. Many processors contain general-purpose fuses that can be used to implement such protection.

Finding	**Ineffective Firmware Integrity Validation**
Risk	**Medium** Impact: Medium, Exploitability: Medium
Identifier	NCC-ExampleCorp001-005
Category	Data Validation
Component	Camera
Impact	A malicious user can install modified device firmware images on the camera, which could be used to bypass business logic or gain unauthorized access to sensitive information.
Description	During interviews, NCC Group was informed that the firmware download package includes a SHA256 hash of the firmware image, which is used by the camera to ensure the firmware image has not been tampered with by a malicious user. This is not a sufficient security control, as a malicious user who has the ability to modify the firmware image can simply recompute the SHA256 hash of the modified image, and replace the hash included in the firmware package.
Recommendation	The firmware package should be protected using a digital signature as apposed to a simple hash. The firmware package should be signed using a private key maintained by ExampleCorp, and the signature validated on the camera prior to firmware installation.

Finding	**Unencrypted Communication Channel (Camera Web Services)**
Risk	**Medium** Impact: Medium, Exploitability: Medium
Identifier	NCC-ExampleCorp001-007
Category	Data Exposure
Component	Camera
Impact	Attackers on the user's local network can intercept and modify traffic, capturing user credentials, tokens, and data or modifying traffic flowing between the mobile and web applications and the camera.
Description	The mobile and web applications communicate with the camera using unencrypted HTTP. An attacker with access to the user's local network could easily intercept and monitor or modify the application's communication. The unencrypted data includes the following types of data, which could be captured or modified by an attacker: • Credentials, including plaintext WiFi passwords and authentication tokens • Personally identifiable information (PII) • Video footage and audio recordings
Recommendation	Where possible, unencrypted protocols should be replaced with encrypted alternatives. HTTP connections should be replaced with HTTPS using TLS 1.2. In a local-network scenario, it is generally more difficult to deploy and manage TLS certificates to enable TLS communication. NCC Group recommends generating and deploying a device-specific TLS certificate to each camera during the manufacturing or registration process. Assuming the serial number of the camera is "SN12345", then ExampleCorp's backend could request a TLS certificate from a globally-trusted CA for "sn12345.examplecorpwificameras.com". Alternatively, a self-signed certificate could be generated at registration time and pinned in the mobile application. However, this would not protect any communications occurring from the web application. In addition to simply configuring HTTPS, the following changes are also needed to enable secure communications: 1. Ensure that unencrypted requests are redirected to HTTPS. 2. Set the "Secure" attribute on all sensitive cookies. This will prevent the cookies from leaking over unencrypted channels.

Finding	**Horizontal Authorization Bypass in Storage API**
Risk	**Medium** Impact: High, Exploitability: Medium
Identifier	NCC-ExampleCorp001-008
Category	Access Controls
Component	Camera Repository
Location	Auth Controller
Impact	Employees or contractors of advertising partners are able to modify session identifiers to gain access to end user image files in cloud storage.
Description	Authorization controls refer to an application's functionality for verifying user roles and permissions. Authorization controls ensure safe access to the application, the data within, the ability to change the system, and more. Every functional piece of the application should have a set of authorization controls that encompasses the use cases surrounding it and prevents unauthorized users from accessing the protected functionality. The application has insufficient authorization controls for access to the Storage API. According to interviews with the development team, it was explained that end user and advertising partner authentication uses the same session identifier mechanism, with a specially-modified session token used to differentiate end users from advertising partner sessions. As a result, in certain scenarios, advertising partner employees (Threat Agent 4, or TA4) may be able to compute valid end user session identifiers to gain access to the Storage API as an authorized end user (TA1). Although Advertising Partners were considered out of scope for this engagement, NCC Group documented this finding for further verification due to the potential risks to end user data.
Recommendation	All application functionality should have authorization controls in place and validate those controls prior to performing an action, including both data retrieval and data modification. Specifically, ExampleCorp should implement an authorization mechanism that ties user roles to session identifiers in a way that cannot be modified by end users to bypass role-based authorization checks.

Finding	**Application Uses Weak Hash to Store Passwords**
Risk	**Medium** Impact: Medium, Exploitability: Medium
Identifier	NCC-ExampleCorp001-010
Category	Cryptography
Component	User Management
Impact	An attacker who is able to compromise the application's storage, such as through a database flaw or information disclosure, can more easily recover user passwords.
Description	The application stores user passwords using unsalted SHA256. This hash can be performed very quickly, thus hashed passwords are vulnerable to brute-force cracking. An attacker with access to the hashed passwords is likely to be able to recover significant numbers of plaintext passwords using a tool such as hashcat. Passwords are hashed without being salted. A salt is a random per-password value which slows down cracking attempts by requiring attackers to perform a new brute-force attack on each password. For example, with a salt, two users with the password "123456" would have different password hashes. Without a salt, an attacker can perform their brute-force attack on all of the application's passwords at once without any additional work, or make use of a rainbow table.
Recommendation	NCC Group recommends using bcrypt, a widely-supported hash that handles salting automatically. This hashing algorithm is designed to resist brute-forcing attempts. The only aspect of bcrypt that requires manual configuration is the "cost factor", a value which determines how many iterations to perform[1] (thus determining the amount of processing time necessary to hash a single password). NCC Group recommends that most application use a cost factor of 12. On modern processors, this cost factor results in hashing taking approximately half a second for each password.[2] To upgrade to a modern password hash, there are a number of possible strategies: • Discard all existing hashes and force users to set new passwords. This strategy may work well for applications with a small number of users who can be reached directly, such as internal company applications. • Upgrade users at their next login by comparing the existing hash, saving the new hash, and finally discarding the existing hash. This strategy is effective for active users, but can have issues with users who do not log in for long periods of time: their old, weak password hash will remain in the database until they log in. • Immediately upgrading all hashes by using the new hashing algorithm on the existing hashed passwords. When users log in, first hash their password with the legacy algorithm, then hash again with the modern algorithm. This strategy is safe in most cases (even for weak algorithms like MD5), but requires the application to maintain the backwards-compatibility logic indefinitely.

[1] See https://security.stackexchange.com/a/3993 for a more in-depth explanation of work factors for password hashing.
[2] See https://security.stackexchange.com/a/83382 for a benchmark of bcrypt on a recent Intel processor.

Finding	**Insecure Random Number Generation**
Risk	**Medium** Impact: Medium, Exploitability: Medium
Identifier	NCC-ExampleCorp001-013
Category	Cryptography
Component	Device Ownership Manager
Impact	A malicious user who is able to predict device IDs of other cameras can take advantage of finding NCC-ExampleCorp001-003 on page 12 to view the images taken by the camera identified by the device ID.
Description	The Secure Camera system makes use of an insecure random number generator (RNG) for the camera device ID generation. When used in a context that requires the numbers to be truly unpredictable, usage of an insecure random number generator can completely defeat the security of the system. Using an insecure generator, or seeding a secure generator with a predictable value, can allow an attacker to predict the generator's output. The following random number generators are known to be insecure, and should not be used: • Java `Random` • PHP `rand`, `mt_rand` • JavaScript `Math.random` • libc `random()` • Any other random number generator that does not advertise cryptographic randomness
Recommendation	Any functionality that relies on a random number generator to generate unpredictable values should use a cryptographically-secure random number generator (CSPRNG). It is strongly encouraged to use a secure RNG unless the application's functionality depends on the output being predictable. Any usage of an insecure RNG should be replaced by a CSPRNG. NCC Group recommends using: • Unix `getrandom()` (falling back to `/dev/urandom` if not available) • Windows `CryptGenRandom` • Java `SecureRandom` (do not manually seed SecureRandom) • PHP `random_bytes` • JavaScript `window.crypto.getRandomValues` • Other: `randombytes_buf` from `libsodium`

Finding	**Revenue Generating Advertisement Delivery Can Be Bypassed**
Risk	**Low** Impact: Medium, Exploitability: Low
Identifier	NCC-ExampleCorp001-006
Category	Other
Component	Mobile Application
Impact	Users who are able to disable the targeted advertisements will impact the Secure Camera business model, which relies on advertisement delivery for revenue generation.
Description	During design documentation review, NCC Group determined the mobile application periodically fetches advertisements from the advertising partner, which is the primary source of revenue for the Secure Camera platform. It appears possible for a malicious user to disable advertisements by blocking client communication to the advertisement partner, or using a modified version of the client application. This would undermine ExampleCorp's business model and could ultimately mean the service does not achieve profitability. However, the overall risk of the issue is low because it is very unlikely that the issue would be exploited at scale due to the technical expertise required for users to set up network blocking of the advertising hosts.
Recommendation	Client-side controls are difficult to enforce as they are running in an untrusted environment, such as a user's mobile device. Consider modifying the current architecture to deliver the targeted advertisements through the Secure Camera API responses, which then must be called to meet the functional requirement of the mobile application.

Finding	**Weak Password Complexity Requirements**
Risk	**Low** Impact: Medium, Exploitability: Low
Identifier	NCC-ExampleCorp001-009
Category	Authentication
Component	User Management
Impact	An attacker may guess insecure user passwords or brute-force weak user passwords in the event of a password database breach.
Description	The Secure Camera platform does not enforce any restrictions on user or administrator passwords. The only current password requirement is a length requirement of 4 characters. As a result, it is possible for users to set their passwords to simple values such as "1234" or "abcd". If a user does use a weak password, an attacker could guess their password and gain access to their account. Alternatively, in the event of a password database breach, an attacker is more likely to recover a weak password from a brute-force attack.
Recommendation	When creating or changing user passwords, the application should perform checks for password complexity and reuse of previous credentials. Complexity checks may be performed client side in order to provide immediate user feedback, but must be enforced by the backend service, since client-side controls should not be considered an effective security control. NCC Group recommends setting a minimum password length of 12 or more characters. Do not require users to regularly update passwords, as this results in weaker passwords overall (see *NIST's new password rules –what you need to know* from Sophos). Finally, strongly consider providing users with the option to use multi-factor authentication for all applications.

Appendix A: Finding Field Definitions

The following sections describe the risk rating and category assigned to issues NCC Group identified.

Risk Scale

NCC Group uses a composite risk score that takes into account the severity of the risk, application's exposure and user population, technical difficulty of exploitation, and other factors. The risk rating is NCC Group's recommended prioritization for addressing findings. Every organization has a different risk sensitivity, so to some extent these recommendations are more relative than absolute guidelines.

Overall Risk

Overall risk reflects NCC Group's estimation of the risk that a finding poses to the target system or systems. It takes into account the impact of the finding, the difficulty of exploitation, and any other relevant factors.

Critical Implies an immediate, easily accessible threat of total compromise.

High Implies an immediate threat of system compromise, or an easily accessible threat of large-scale breach.

Medium A difficult to exploit threat of large-scale breach, or easy compromise of a small portion of the application.

Low Implies a relatively minor threat to the application.

Informational No immediate threat to the application. May provide suggestions for application improvement, functional issues with the application, or conditions that could later lead to an exploitable finding.

Impact

Impact reflects the effects that successful exploitation has upon the target system or systems. It takes into account potential losses of confidentiality, integrity and availability, as well as potential reputational losses.

High Attackers can read or modify all data in a system, execute arbitrary code on the system, or escalate their privileges to superuser level.

Medium Attackers can read or modify some unauthorized data on a system, deny access to that system, or gain significant internal technical information.

Low Attackers can gain small amounts of unauthorized information or slightly degrade system performance. May have a negative public perception of security.

Exploitability

Exploitability reflects the ease with which attackers may exploit a finding. It takes into account the level of access required, availability of exploitation information, requirements relating to social engineering, race conditions, brute forcing, etc, and other impediments to exploitation.

High Attackers can unilaterally exploit the finding without special permissions or significant roadblocks.

Medium Attackers would need to leverage a third party, gain non-public information, exploit a race condition, already have privileged access, or otherwise overcome moderate hurdles in order to exploit the finding.

Low Exploitation requires implausible social engineering, a difficult race condition, guessing difficult-to-guess data, or is otherwise unlikely.

Category

NCC Group categorizes findings based on the security area to which those findings belong. This can help organizations identify gaps in secure development, deployment, patching, etc.

Access Controls	Related to authorization of users, and assessment of rights.
Auditing and Logging	Related to auditing of actions, or logging of problems.
Authentication	Related to the identification of users.
Configuration	Related to security configurations of servers, devices, or software.
Cryptography	Related to mathematical protections for data.
Data Exposure	Related to unintended exposure of sensitive information.
Data Validation	Related to improper reliance on the structure or values of data.
Denial of Service	Related to causing system failure.
Error Reporting	Related to the reporting of error conditions in a secure fashion.
Patching	Related to keeping software up to date.
Session Management	Related to the identification of authenticated users.
Timing	Related to race conditions, locking, or order of operations.

Appendix B: Interview Notes

NCC Group has included in this report notes recorded from interviews held with ExampleCorp subject matter experts. [NOTE: this is an OPTIONAL component of reports. This information is typically only recorded in NCC Group working papers to support analysis, not provided as a formal deliverable. We've included it here because customers sometimes explicitly request interview notes for evidentiary purposes, and also to illustrate NCC Group's interview process and outcomes.]

Example Interview 1

- **Date**: August 1, 2019
- **Time**: 9:00 AM
- **Location**: ExampleCorp Conference Room 1
- **Participants**: Secure Camera Repository Developer

Q: What is your role?

A: I'm the lead developer for the Camera Repository functionality in Secure Camera Cloud.

Q: How long have you been in this role?

A: 3 years. I did a little work on the mobile application prior to that.

Q. What is the purpose of the Camera Repository?

A. The camera repository stores all camera images for all clients.

Q. How does it store the data?

A. The images are kept in a database. When the camera takes an image it uploads it to the storage API which then sends it to the Image Manager. We then store the raw image in the database.

Q. Are the images encrypted?

A. No, we just store the images in plaintext to the database.

Q. How do you keep track of which user / device stored the image?

A. There is a device ID sent with each upload request. We pass the device ID over to the User Management team and they give us a unique image ID that we keep with the image data in our database.

Q. What happens when the user wants to view their images from the cloud?

A. We get a view image request from the user through the storage API, which contains the device ID. We then pass that to the User Management components, which returns the list of image names associated with the device ID. And we then fetch the images from the database and return them to the device.

Q. When you make changes to the camera repository components, how do you test the changes?

A. We have a staging environment where we clone the production environment each month for our testing purposes.

Q. So, you use real user images for testing purposes?

A. Yes, sometimes, especially when we need a lot of images to fully test this system for performance reasons. And the images usually do not contain anything sensitive in them. Besides, I can view the user images anyway by just logging into the administrator portal.

Q. How many people have access to the administrator portal?

A. I am not sure, we all use the same username and password to login to it.

Q. What exactly does the administrator portal give you access to?

A. We can run queries on all of the databases and view the log files that are stored in the cloud.

Q. And you see the results in cleartext?

A. Yes, but we can't see user passwords. Those are encrypted.

Q. How are they encrypted?

A. I'm not sure, but I think they are SHA256 encrypted or something. You'll have to talk with the User Management team.

Q. When the camera sends you a device ID in the view image request, how do you know if the device ID they sent you is actually owned by the user who is logged in?

A. Yeah, so that's why we made the camera send us the device ID. Because it is a lot harder for someone to change the device ID if it is coming from the camera.

Q. So, you do not attempt to map the device ID to the user's session ID?

A. I am not sure - I don't do that, but maybe the User Management team does.

Example Interview 2

- **Date**: August 1, 2019
- **Time**: 11:00 AM
- **Location**: ExampleCorp Conference Room 4
- **Participants**: Secure Camera User Management Developer

Q: What is your role?

A: I'm a developer for the User Management components.

Q: How long have you been in this role?

A: 6 months.

Q: What is the purpose of the User Management components?

A: We do authentication and authorization relation functions. Such as login and logout.

Q: Tell me a little about how that works on this system.

A: [example response]

Q: [next example question]

A: [etc.]

Appendix C: Client Provided Documentation

The following design documentation was provided to NCC Group by ExampleCorp using NCC Group's encrypted Secure File Exchange (SFE) service.

- Web Administration Use Case Diagrams: `webadm_usecase.docx`
- Wi-Fi Camera Hardware Specifications: `cameraInternals.pdf`
- Example Log File: `20190801_devicelogs.txt`
- Cloud Service API List: `camera_apis.pdf`
- Disaster Recovery Diagrams: `Backup Architecture.pdf`
- API Route Information: `REST_API_Schema.xml`
- User Role Information: `Users_and_Groups.csv`
- Patch Management Specification: `Patching_Process.xls`
- [etc.]

Appendix D: Project Contacts

The team from NCC Group has the following primary members:

- Dade Murphy — Technical Lead
  [firstname.lastname]@nccgroup.com
- Kate Libby — Consultant
  [firstname.lastname]@nccgroup.com
- Lauren Murphy — Account Manager
  [firstname.lastname]@nccgroup.com

The team from ExampleCorp has the following primary member:

- Eugene Belford — Security Architect, ExampleCorp
  [firstname.lastname]@[examplecorp].com