RAPIDIDENTITY MFA AND RAPIDIDENTITY eSSO

SECURITY MODEL AND AUTHENTICATION METHODS EXPLAINED

IDENTITY AUTOMATION

TABLE OF CONTENTS

RAPIDIDENTITY MFA & SSO SECURITY

SIMPLICITY IN SECURITY

Security, a necessary component of any IT infrastructure, is only as effective as its rate of adoption. Typically, the more users must deviate from their most convenient way of doing things, the less likely they are to abide by security policies.

For years, authentication applications have been weighted down with cumbersome solutions – complicated to install, expensive to maintain, and irritating to end users in varying degrees. These issues have prevented the widespread adoption of two-factor authentication.

RapidIdentity was born out of the belief that security works best when it operates seamlessly within the processes users already expect to experience. Within those confines, RapidIdentity enables greater adoption of authentication throughout organizations large and small. Additional authentication methods are quick to install, simple to understand, and easy to use with RapidIdentity. As such, the simplicity of the application tends to leave some pundits wondering how it is possible for an application that is so straightforward to be so secure at the same time.

To that end, this paper addresses some of the most frequently asked questions concerning RapidIdentity's security architecture, specifically, RapidIdentity Server and RapidIdentity Client. The rationales behind security decisions are also discussed, as well as many of the checks and balances which occur "under-the-hood".

BACKGROUND & SCOPE

True to the easy-to-understand vision of RapidIdentity MFA and RapidIdentity eSSO, the answers to the following frequently asked questions are presented first at a high level and then elaborate with further levels of technical detail as appropriate. To best appreciate the architecture, it is helpful to have a good understanding of how Microsoft® Windows® operates today (without RapidIdentity) and a general awareness of the various types of cards, readers, and authentication methods supported by RapidIdentity. Additional technical explanations may require a much more technical understanding of the inner workings of not only the operating system, but also of standards such as PKCS, CAPI/CSP, and PKI (X.509).

STRONG AUTHENTICATION – AN OVERVIEW

RapidIdentity supports nine forms of authentication out-of-the-box: 1) username/password, 2) contact/contactless smart cards or tokens with PKI, 3) fingerprint biometrics, 4) RFID cards and tokens, 5) One-Time Password (OTP), 6) Magnetic strip/2D barcode, and 7) Risk-Based Authentication 8) Push-Authentication, and 9) Challenge/Response authentication. It is important to point out that Windows® natively supports only two forms of logon – via username/password and through a digital certificate located on a smart card or token. Therefore, each of the other authentication methods outlined below isolates the underlying Windows® authentication process from the user's logon experience, but ultimately is based on one of these two forms.

CONTACT/CONTACTLESS SMART CARD WITH PKI – A smart card contains a cryptographic module to facilitate the generation and security of PKI keys that are used to authenticate to operating systems and applications. RapidIdentity manages the lifecycle of smart cards, the creation of certificates on the card, and the selection of PINs by users. Smart cards are commonly used in the US Government and are considered one of the stronger forms of authentication. The common workflow for users utilizing smart cards requires the user to insert a card or cryptographic USB token into a reader, then enter the associated PIN, and once validated, a key exchange occurs between the operating system or an application to validate the certificate and associated keys. Once validated the user is permitted access to the operating system or application. RapidIdentity also supports contactless PKI authentication utilizing dual-interface smart cards.

FINGERPRINT BIOMETRICS – Fingerprint biometrics are commonly accessed by organizations seeking to comply authentication requirements. Fingerprint biometrics are used around the world, and are considered one of the more secure forms of authentication. Fingerprint biometrics matches a known fingerprint template with that presented by a user at the time of an authentication request. RapidIdentity manages the lifecycle of fingerprint biometrics by capturing three or more of a user's fingerprint templates during enrollment and a user selected PIN. The templates are encrypted and stored on RapidIdentity Server. The common workflow for using fingerprint biometrics requires the user to present or swipe one of their enrolled fingerprints on a sensor; once recognized the user must enter the associated PIN. This information is

then compared to the information stored on the RapidIdentity Server. Once validated, the user is permitted access to the operating system or application. Settings can be configured within RapidIdentity to require: 1) server-best authentication all the time, 2) server-based authentication when connected to the network, or 3) offline identity management, that permits users to authenticate when not connected to the network.

RFID – RFID contains a module that emits a unique identifier when presented to an RFID reader. RFID comes in many different form factors and generally supports two different frequencies, 125 kHz and 13.56 MHz. Most organizations currently use the 125 kHz technology with newer implementations using 13.56 MHz. 13.56 MHz technology is generally considered more secure due to mutual authentication and more complex cryptography. RFID based authentication is considered less secure than contact smart cards and biometrics, but more secure than other forms of authentication. RapidIdentity manages the lifecycle of both 125 kHz and 13.56 MHz cards and the selection of PINs by users. RDIF is in broad use throughout the world for building access and is considered a preferred form of authentication by users due to the ease of use and the multi-purpose capabilities. The common workflow for RFID authentication requires the user to present their device (card, fob, tag, phone, etc.) to a connected or embedded RFID reader (USB, embedded, PCMCIA, PC Express), RapidIdentity then identifies the user's information (the user does not have to enter a username) and requests the user to enter their PIN associated with the card; the user then enters their PIN and RapidIdentity validates the two components. Once validated the user is permitted access to the operating system or application.

ONE-TIME PASSWORD – A one-time password (OTP) is a password that is valid for only one login session or transaction. OTPs avoid many shortcomings that are associated with traditional (static) passwords. The most important shortcoming that is addressed by OTPs is, that in contrast to static passwords, they are not vulnerable to replay attacks. This means that a potential intruder who manages to record an OTP that was already used to log into a service or to conduct a transaction will not be able to abuse it, since it will be no longer valid. On the downside, OTPs are difficult for users to memorize; therefore, they require additional technology to work such as a token or

application. OTPs are in common use throughout the world for remote access. OTP is considered one of the stronger forms of authentication. Many organizations consider the use of OTP when supporting remote users or if an existing implementation is present. RapidIdentity manages the lifecycle of OTP token seeds that are assigned to users; the token seeds are then associated with the user and a specific device, such as a token or mobile phone application. The common workflow for OTP is, for a user to enter a six-digit code in conjunction with their username and an associated PIN. The codes are routinely entered in websites or VPN applications. The codes are generated on tokens or from within an application. Once validated, the user is permitted access to the operating system or application.

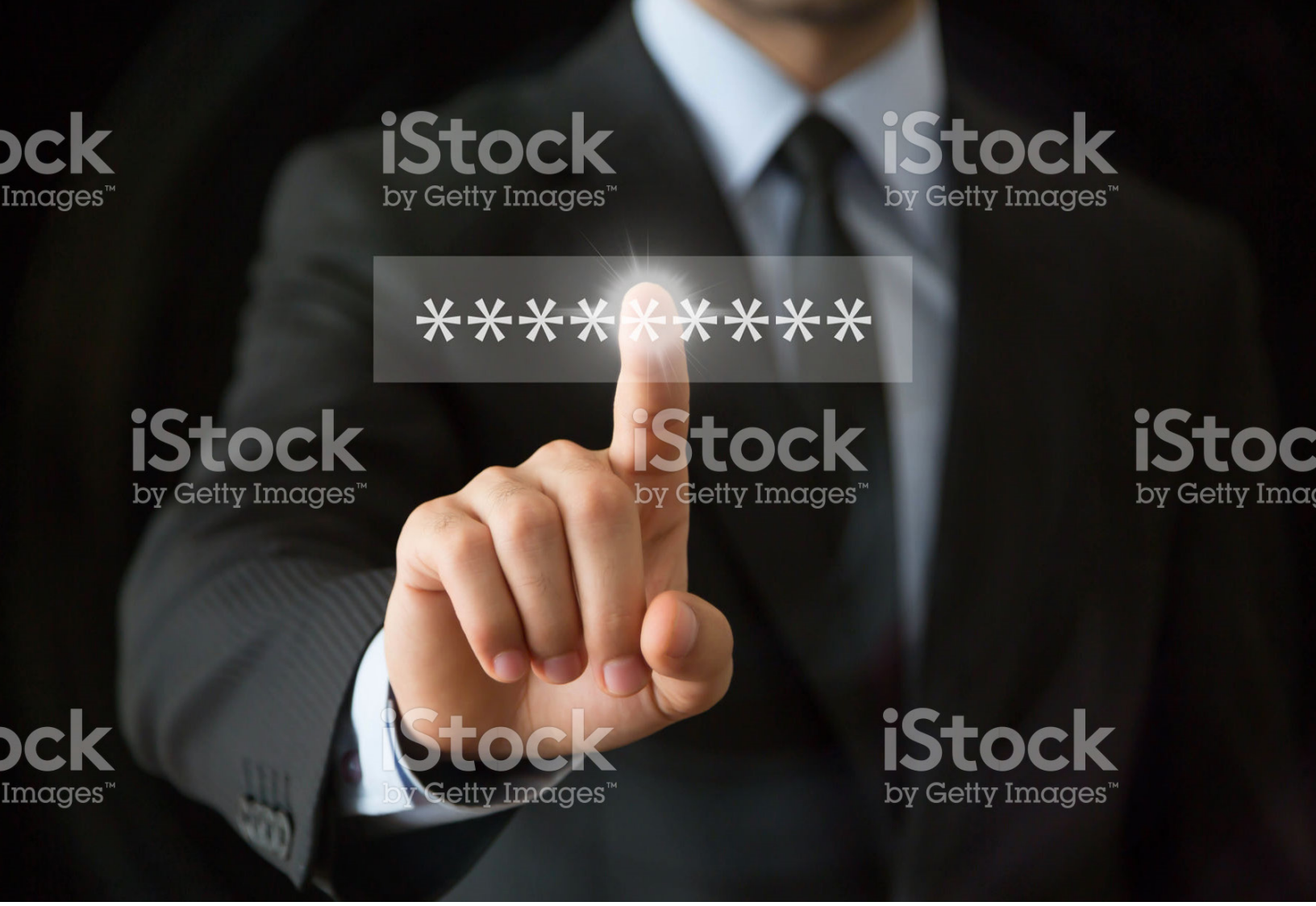
PUSH-AUTHENTICATION – Push-Authentication, called PingMe in RapidIdentity utilizes push notifications to a pre-registered mobile phone or other device. This is an out-of-band authentication that cannot be intercepted at the point of password entry. RapidIdentity manages which devices are authorized for use for authenticating a PingMe authentication request. In addition, if a user receives a request that they did not initiate, a "Deny" response not only prevents authentication, but can also be configured to notify an admin or other party. Traditionally, at login, this process is combined with the use of a password, but can be configured to not require a password at all. Behind the scenes, the authentication is protected by actually utilizing a one-time password in the response message back, thereby protecting against replay attacks and other types of attacks, similar to the security benefits associated with OTP. Because of RapidIdentity's implementation of PingMe, it is considered a stronger form of authentication, particularly as it pertains to adding out-of-band authentication to an otherwise typical authentication request.

MAGNETIC STRIPE AND 2D BARCODE – A magnetic stripe or 2D barcode contains data stored on a magnetic stripe or represented in a 2D image, both are read by magnetic stripe or optical readers when presented by the user. Many organizations consider the use of magnetic stripe or 2D barcode when their vehicles or point of sale systems contain a reader for swipe or scan of a card. The technology is generally considered one of the least secure methods of authentication due to the ease in copying or cloning

magnetic stripe and 2D barcode technology. RapidIdentity manages the lifecycle of both magnetic stripe and 2D barcode technology, and the selection of PINs by users. The common workflow for magnetic stripe and 2D barcodes requires the user to present their card to a connected magnetic stripe or 2D barcode reader, RapidIdentity then identifies the users information (the user does not have to enter a username) and requests the user to enter their PIN associated with the card; the user then enters their PIN and RapidIdentity validates the two components. Once validated, the user is permitted access to the operating system or application.

RISK-BASED AUTHENTICATION – Risk-Based Authentication (RBA) includes a software token element comprised of many factors, such as network information, user information, positive device identification (i.e. device forensics, user pattern analysis and user binding), user profiling, and high-risk challenge/response questions. The technology is generally considered one of the least secure methods of authentication. Organizations routinely deploy RBA as a fallback method to another form of authentication or as a primary in deployments where budgetary constraints are the driving force. RapidIdentity manages the generation of the user-based software token and associated user-based profiling. During enrollment, users select questions from a list of 27 questions that they provide answers to. The answers are then encrypted and stored in RapidIdentity Server. The common workflow for RBA includes the user logging on to the operating system with username and password, RapidIdentity then assesses the level of risk associated with the logon event and locks the system in the event the risk threshold is triggered. The user then must either logon with an approved form of authentication or correctly answer the responses to three challenge questions. Once validated, the user is permitted access to the operating system or application.

CHALLENGE/RESPONSE – Challenge/Response, also known as Knowledge-Based Authentication utilizes previously answered challenge questions to authenticate a user. This process is not considered two-factor authentication and if configured incorrectly can be weaker than a username and password; however, challenge/response is very useful when configured correctly and used in lieu of a password or to reset "something" the user should know, like a PIN or password. This technology may also



be used with RapidIdentity as an Emergency Access option when other stronger forms of authentication are available. Care must be given to when the technology is used as it does create a weak link in your security architecture. During enrollment, users select questions from a list of 27 questions that they provide answers to. The answers are then encrypted and stored in RapidIdentity Server. The common workflow for RBA includes the user logging on to the operating system with username and password, RapidIdentity then assesses the level of risk associated with the logon event and locks the system in the event the risk threshold is triggered. The user then must either logon with an approved form of authentication or correctly answer the responses to three challenge questions. Once validated, the user is permitted access to the operating system or application.

Note: *Challenge/Response should not be confused with RBA or used in lieu of RBA. RBA utilizes other elements to strengthen the authentication process. Additionally, Challenge/Response alone is not compliance for CJIS Advanced Authentication.*

COMPARING LOG-ON SECURITY

Authentication with smart cards, fingerprint biometrics, RFID cards, OTP, Push-Authentication, and magnetic stripe/2D barcode all support two-factor authentication, requiring both something the user has, such as a card or token or something the user is, such as a fingerprint template, and a PIN (something the user knows). However, each has varying levels of security.

Smart card authentication is the strongest form of authentication supported in RapidIdentity, but requires the additional overhead of a PKI to implement. Cryptographic processes secure the PIN and associated keys on the smart card itself. Certificate-based logon requires domain-joined machines and domain user accounts, as well as running certificate services in the directory infrastructure. The US DoD adopted this technology more than 10 years ago and the technology is in broad use throughout the world. Organizations commonly explore the use of smart cards with PKI, but run into usability and cost constrains during evaluation and deployments. In many cases only a subset of a larger organizations end up utilizing smart cards with PKI, while other users utilize another form of authentication. This flexibility drives organizations to elect to deploy RapidIdentity over the competition. Coupling a smart card with physical access technology in an all-in-one authentication token has proven more useful and convenient for users and organizations alike.

RFID logon with 125 kHz RFID technology is more secure than a username and password alone; however, it contains known security vulnerabilities in that the card performs no cryptographic processes. The 125 kHz proximity card transmits a static card number unencrypted to the reader/application, and the user enters a PIN. Validation of the card number and PIN are sufficient for logon. However, the unencrypted nature of the transmission means that a sniffing device operating on the same frequency and in close proximity to the card could intercept the card number, enabling duplication of that card and subsequent reply attacks. A potential hacker would still need to gain access to the user's PIN and a domain joined system to accomplish authentication.

In contrast, 13.56 MHz devices (iCLASS, Mifare, NFC, Felica, etc.) following the ISO 14443 specification contains three separate encrypted data values, performs cryptographic processes, plus utilizes a PIN to validate against in the logon process. Additionally, 13.56 MHz cards are writeable, enabling mutual authentication during

logon. After validating the card is registered itself, RapidIdentity then validates the PIN against the secure, encrypted cache. If successful, it then decrypts the logon credentials in the secure cache and passes them to the Local Security Authority (LSA) in Windows®, in the same manner as Microsoft® does during natively supported logon processes. 13.56 MHz provides several usability advantages over contact-based smart cards by enabling secure logon of non-domain joined machines and local user accounts, and not requiring the deployment of Public Key Infrastructure and use of certificate services.

The security of fingerprint biometrics is driven largely in part to the type of reader technology that is used to capture and authenticate the user's fingerprint template. Cost and availability are commonly the driving forces behind which readers are used in organizations today. RapidIdentity provides the broadest support for fingerprint sensors on the market today, including support for fingerprint sensors from: Authentec, BIO-key, Broadcom, CrossMatch, Dell, Digital Persona, HP, Fujitsu, Lenovo, Lumidigm, Next Biometrics, Panasonic, UPEK, and others. Costs range from $0 for existing embedded readers in tablets, laptops, and other devises, to several hundred dollars for specialty readers that perform live sub-surface dermal scans, such as those provided by Lumidigm.

RapidIdentity supports biometric readers following three integration paths: 1) vendor provided APIs, 2) BIO-key, and 2) Windows Biometric Framework (WBF). Vendor specific integration with RapidIdentity requires Identity Automation to enter into a non-disclosure agreement with the providing vendor to gain access to special software development kit (SDK) features provided by the specific vendor. The vendor SDK provides APIs that dictate how user's fingerprint data is collected and matched during authentication processes. Each vendor implements proprietary processes concerning how fingerprint data is secured by the associated readers. Since this information is proprietary to the vendor we will address how each technology works with RapidIdentity, not how each vendor implements their security. Additionally, more information on how WBF works can be found here: http://msdn.microsoft.com/en-us/library/windows/hardware/gg463089.aspx.

Fingerprint biometrics refers to the automated method of verifying a match between two human fingerprints. This document touches on two major classes of algorithms (minutia and pattern) and four sensor designs (optical, ultrasonic, passive capacitance, and active capacitance). The analysis of fingerprints for matching purposes generally requires the comparison of several features of the print pattern. These include patterns, which are aggregate characteristics of ridges, and minutia points, which are unique features found within the patterns. It is also necessary to know the structure and properties of human skin to successfully employ some of the imaging technologies.

Patterns

The three basic patterns of fingerprint ridges are the arch, loop, and whorl:

- ARCH: The ridges enter from one side of the finger, rise in the center forming an arc, and then exit the other side of the finger.
- LOOP: The ridges enter from one side of a finger, form a curve, and then exit on that same side.
- WHORL: Ridges form circularly around a central point on the finger.

Minutia

The major Minutia features of fingerprint ridges are: ridge ending, bifurcation, and short ridge (or dot). The ridge ending is the point at which a ridge terminates. Bifurcations are points at which a single ridge splits into two ridges. Short ridges (or dots) are ridges which are significantly shorter than the average ridge length on the fingerprint. Minutiae and patterns are very important in the analysis of fingerprints since no two fingers have been shown to be identical.

Fingerprint Sensors

A fingerprint sensor is an electronic device used to capture a digital image of the fingerprint pattern. The captured image is called a live scan. This live scan is digitally processed to create a biometric template (a collection of extracted features) which is stored and used for matching. This is an overview of some of the supported fingerprint sensor technologies.

Optical (Digital Persona and Lumidigm – USB connected)

Optical fingerprint imaging involves capturing a digital image of the print using visible light. This type of sensor is essentially a specialized digital camera. The top layer of the sensor, where the finger is placed, is known as the touch surface. Beneath this layer is a light-emitting phosphor layer which illuminates the surface of the finger. The light reflected from the finger passes through the phosphor layer to an array of solid state pixels (a charge-coupled device) which captures a visual image of the fingerprint. A scratched or dirty touch surface can cause a bad image of the fingerprint. A disadvantage of this type of sensor is the fact that the imaging capabilities are affected by the quality of skin on the finger. For instance, a dirty or marked finger is difficult to image properly. Also, it is possible for an individual to erode the outer layer of skin on the fingertips to the point where the fingerprint is no longer visible. It can also be easily fooled by an image of a fingerprint if not coupled with a "live finger" detector. However, unlike capacitive sensors, this sensor technology is not susceptible to electrostatic discharge damage. Both Digital Persona and Lumidigm utilize live finger detection technology.

Ultrasonic (Lumidigm – USB Only – USB connected)

Ultrasonic sensors make use of the principles of medical ultrasonography in order to create visual images of the fingerprint. Unlike optical imaging, ultrasonic sensors use very high frequency sound waves to penetrate the epidermal layer of skin. The sound waves are generated using piezoelectric transducers and reflected energy is also measured using piezoelectric materials. Since the dermal skin layer exhibits the same characteristic pattern of the fingerprint, the reflected wave measurements can be used to form an image of the fingerprint. This eliminates the need for clean, undamaged epidermal skin and a clean sensing surface.

Capacitance (Authentec, Digital, Personal, embedded swipe sensors – USB connected and embedded)

Capacitance sensors use principles associated with capacitance in order to form fingerprint images. In this method of imaging, the sensor array pixels each act as one plate of a parallel-plate capacitor, the dermal layer (which is electrically conductive)

acts as the other plate, and the non-conductive epidermal layer acts as a dielectric. A passive capacitance sensor use the principle outlined above to form an image of the fingerprint patterns on the dermal layer of skin. Each sensor pixel is used to measure the capacitance at that point of the array. The capacitance varies between the ridges and valleys of the fingerprint due to the fact that the volume between the dermal layer and sensing element in valleys contains an air gap. The dielectric constant of the epidermis and the area of the sensing element are known values. The measured capacitance values are then used to distinguish between fingerprint ridges and valleys.

Algorithms (Authentec, BIO-key, Broadcom, Dell, Digital Persona, HP, Fujitsu, Lumidigm, and others)

Matching algorithms are used to compare previously stored templates of fingerprints against candidate fingerprints for authentication purposes. In order to do this either the original image must be directly compared with the candidate image or certain features must be compared.

Pattern-based (or image-based) algorithms:

Pattern based algorithms compare the basic fingerprint patterns (arch, whorl, and loop) between a previously stored template and a candidate fingerprint. This requires that the images be aligned in the same orientation. To do this, the algorithm finds a central point in the fingerprint image and centers on that. In a pattern-based algorithm, the template contains the type, size, and orientation of patterns within the aligned fingerprint image. The candidate fingerprint image is graphically compared with the template to determine the degree to which they match.

RapidIdentity supports FIPS 201 certified sensors from Authentec, Futronics, UPEK, and Digital Persona.

Fingerprint biometrics contain separate encrypted data values, the user's encrypted fingerprint templates plus the PIN to validate against in the logon process. Prior to logon, the user authenticates to the RapidIdentity Client by placing or swiping their finger on a connected reader. The vendor's API collects the fingerprint data and sends it to RapidIdentity Client for comparison, the data is compared and upon validation the

user's name appears in the logon screen. RapidIdentity Client then requires the user to enter their PIN. RapidIdentity then validates the PIN against the secure, encrypted cache. If successful, it then decrypts the logon credentials in the secure cache and passes them to the Local Security Authority (LSA) in Windows®, in the same manner as Microsoft® does during natively supported logon processes. Fingerprint biometrics provides several usability advantages over contact-based smart cards by enabling secure logon of non-domain joined machines and local user accounts, and not requiring the deployment of Public Key Infrastructure and use of certificate services.

RapidIdentity utilizes TOTP and HOTP for One-Time Password. HOTP is an HMAC-based One Time Password algorithm. It is a cornerstone of Initiative For Open Authentication (OATH).

HOTP was published as an informational IETF RFC 4226 in December 2005, documenting the algorithm along with a Java implementation. Since then, the algorithm has been adopted by many companies worldwide. The HOTP algorithm is a freely available open standard.

RapidIdentity leverages Microsoft RADIUS (Network Policy Server) as an authentication server. Both hard tokens and soft-tokens are available for use with RapidIdentity. RapidIdentity Client does not support the use of OTP for Windows logon. OTP can only be used to authenticate client or browser-based applications.

Magnetic swipe or 2D barcode logon technology is more secure than a username and password alone; however, it contains known security vulnerabilities. Both magnetic swipe and 2D barcode can be easy cloned or copied and transmit static data unencrypted to the reader. Validation of the data and PIN are sufficient for logon. However, the unencrypted nature of the transmission means that copying or cloning the data, enabling duplication of that card. A potential hacker would still need to gain access to the user's PIN and a domain joined system to accomplish authentication.

Once the data is read, RapidIdentity Client then validates the PIN against the secure, encrypted cache. If successful, it then decrypts the logon credentials in the secure cache and passes them to the Local Security Authority (LSA) in Windows®, in the same manner as Microsoft® does during natively supported logon processes.

Additional process security can be leveraged using RapidIdentity Server. Credentials can be immediately revoked, and cards terminated. Users are able to perform secure, self-service of their authentication data (such as unblocking a card or updating their security questions and answers). This allows for faster deployment of strong authentication credentials, in a way that performs secure binding of a credential to a user when the user enrolls.

The final means for logon is via Emergency Access, which makes use of alternative authentication methods to validate the user. When configured (and it is enabled by default), RapidIdentity utilizes a set of security questions that are public in nature and allows the user to select a certain number to answer. The authentication method is straightforward, in that the user must correctly answer a pre-determined number of those questions correctly in order to authenticate. Once authenticated, the process follows the same secure process for logging users on with their secured Windows® logon credentials as the other methods would.

Policies help round out the security infrastructure, by allowing two main types of policy to be implemented. The first is general authentication-method-specific policies. These are policies that govern each method and are specific to those methods. For example, a PIN policy to govern how complex a PIN must be or a Q&A policy to ensure answers are of a certain length, unique from one another or dynamic. These policies are configured in RapidIdentity Server and enforced during browser-based processes and synchronized to associated RapidIdentity Client installations.

The second type of policy is general security policies. RapidIdentity can enforce that Emergency Access only be allowed to be performed a certain number of times before answers must be changed. It can also enforce a lock-out period for invalid Emergency Access attempts. Most of all, RapidIdentity can set which "tiles" appear at logon, thereby removing the option for Emergency Access or Username/password, etc. altogether. This would, in effect, force users to utilize strong, two-factor forms of authentication when logging on to protected systems.

A differentiating feature of RapidIdentity is the ability to secure a user's Windows® logon password. When enabled, RapidIdentity will automatically change the user's logon password to a random, 32 or 64-character password (depending on the operating system) that utilizes the entire Unicode alphabet, including numbers and non-alphanumeric characters. In effect, this scrambles the user's password to the approximate equivalent of a 1024-bit key, known only to RapidIdentity. Given that the user now does not know their password, this process enforces the adoption of strong authentication throughout the organization. Even machines without RapidIdentity installed are then protected from traditional password vulnerabilities as the password has been strengthened to such a degree.

To make sure this remains usable, RapidIdentity supports Windows 7®, Windows 8®, Windows 10®, , Windows Server 2008®, Windows Server 2012®, Windows Server 2016® server environments, including RDP support for each, making it possible to easily utilize strong, two-factor authentication throughout the entire environment. Therefore, when enabled, RapidIdentity handles all password change and reset events on behalf of the user.

FREQUENTLY ASKED QUESTIONS

The following FAQ format addresses common concerns and further technical descriptions.

APPLICATION KEYS

- What are the different types of keys?
- What is a diversified key?
- How are application keys secured?

CACHING & SYNCHRONIZATION

- Why and how does the application cache local user data?
- How is local data secured?
- How is the data synchronized?
- How is synchronization between the client and server applications secured?

AUTHENTICATION DATA

- How is Q&A data secured?
- Does RapidIdentity change the PUK?
- How does RapidIdentity protect stored PUKs?
- Does RapidIdentity store user PINs?
- Does RapidIdentity cache user PINs?
- How is the RapidIdentity data repository protected?
- Does RapidIdentity follow the GlobalPlatform specification for java smart cards?
- Can someone who steals my card get my Windows password?

SECURITY & ENCRYPTION

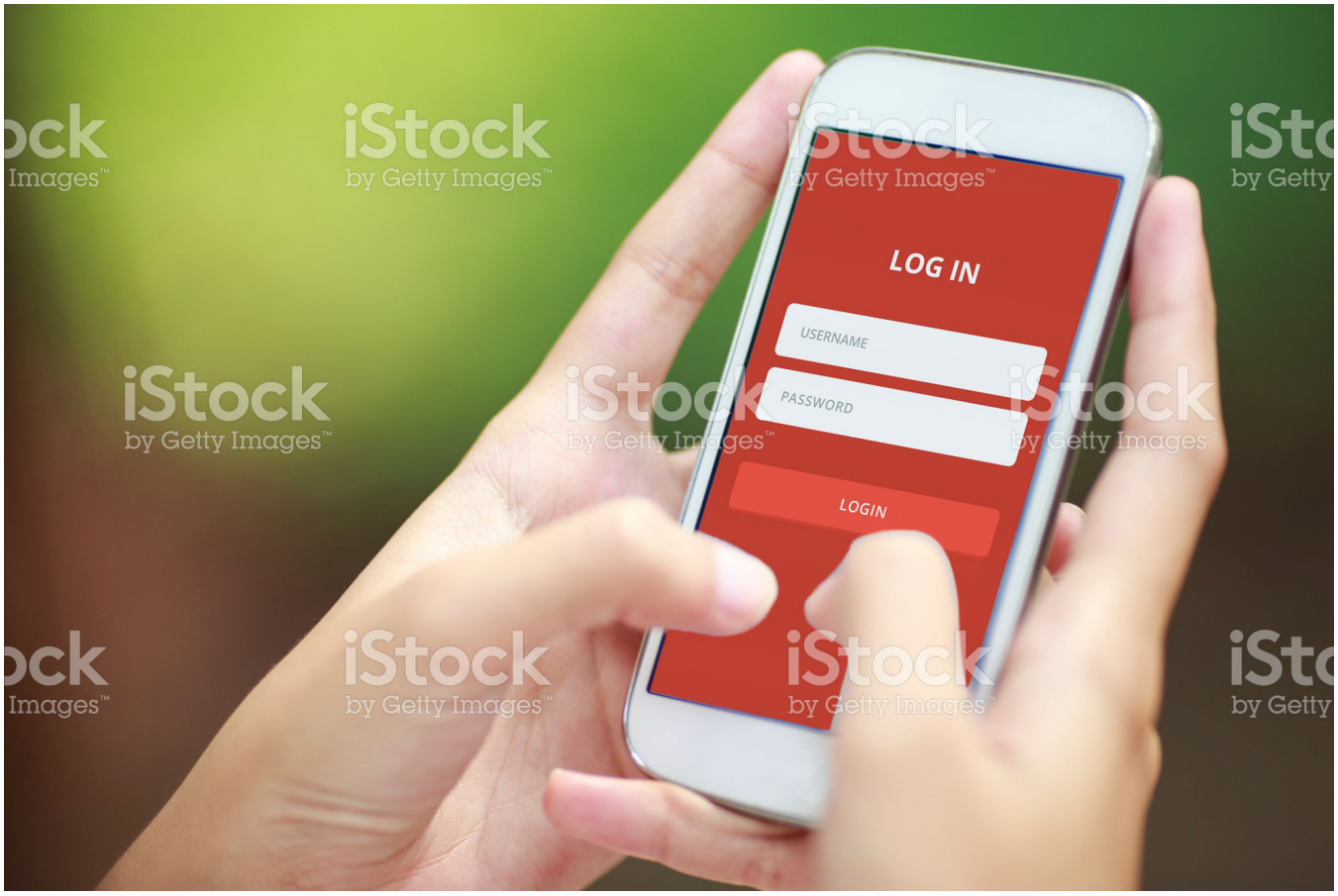
- What cryptographic algorithms are used?
- Can the application support other or custom cryptographic algorithms?
- Can I enable SSL on the server?
- How is the data removed from memory?
- How is audit data secured on the server?

SUPPORT

- What is the ongoing process for updating / protecting this application? How do you plan for and actively protect against potential hacks or future security threats?
- How are bugs reported?

THIRD-PARTY VALIDATION

- What level of quality assurance is RapidIdentity subjected to?
- What level of third-party assessment does RapidIdentity undergo?



APPLICATION KEYS

WHAT ARE THE DIFFERENT TYPES OF KEYS?

In general, "application keys" refer to all keys used in the RapidIdentity product. The master key is the secret key that, while not directly used for any encryption/decryption, is only used as an input into the diversification algorithms which create all other keys. It is assembled 'just-in-time' and wiped from memory after use.

A client key, or Client key, refers to the general encryption key used to encrypt/decrypt the local XML database on a client machine. Client keys are created using a diversification algorithm specific to the machine they are used on. Therefore, every machine has a different client key. If the key for one machine is compromised, all the other machines in the organization remain secure. These keys are assembled 'just-in-time' and wiped from memory after use.

A server key is used to encrypt sensitive data in the data repository on the server. It is derived using a diversification algorithm to ensure that each company with RapidIdentity deployed has a unique key. Specifically, all shared or communicating repositories will have the same server key.

A user key is used to encrypt each specific user's data. They are created through the same diversification process as other keys, but with information unique to each user, result in different user keys. Therefore, security exposure is limited from one user to another.

WHAT IS A DIVERSIFIED KEY?

True to the easy-to-understand vision of RapidIdentity MFA and RapidIdentity eSSO, the answers to the following frequently asked questions are presented first at a high level and then elaborate with further levels of technical detail as appropriate. To best appreciate the architecture, it is helpful to have a good understanding of how Microsoft® Windows® operates today (without RapidIdentity) and a general awareness of the various types of cards, readers, and authentication methods supported by RapidIdentity. Additional technical explanations may require a much more technical understanding of the inner workings of not only the operating system, but also of standards such as PKCS, CAPI/CSP, and PKI (X.509).

HOW ARE APPLICATION KEYS SECURED?

RapidIdentity Client uses key diversification and 'just-in-time' encryption to secure application keys. Keys are never stored on the system, but rather created 'just-in-time' using a one-way diversification algorithm with multiple inputs to ensure that each installation has a unique key that is not susceptible to persistent attacks. This diversification algorithm results in a key that is unique from any other in the system. Therefore, if a key is compromised in one location, keys in other locations remain secure.

'Just-in-time' encryption indicates that the key does not remain physically resident in memory anywhere. When required, the diversification algorithm generates the key in volatile memory, the key is used to perform either encryption or decryption, and then it is immediately disposed of through a secure wipe of the corresponding section of memory. In addition, the RapidIdentity application has been reviewed by a third-party security assessment organization to ensure protection against known key attacks.

CACHING & SYNCHRONIZATION

WHY AND HOW DOES THE APPLICATION CACHE LOCAL USER DATA?

RapidIdentity caches local data to enable offline access and to improve the speed of logon. 100% network connectivity may be impractical in many situations, so caching permits users to gain access while disconnected from the network. Caching also allows comparisons such as those required for RFID logon to be processed locally to best accelerate speed of logon. Caching may be turned off in RapidIdentity to accommodate specific regulatory requirements, such as, CJIS' requirement to not cache biometric fingerprint data. The result will limit a user's ability to logon when offline and slow down logon times.

The caching process occurs through synchronization with RapidIdentity Server and only at key events when required by the system. Such events may include PIN changes or a new enrollment. To optimize the speed of the application, RapidIdentity Client does not have to communicate with the server at every logon. Furthermore, user data is only cached for those that have previously authenticated to a particular machine while on the network or have a local machine account, in keeping with the Microsoft® standard logon processes. This keeps the file size on the local machine small; however, it also means that to log on to the machine for the first time only, it must be connected to the network for domain joined machines. This process mirrors Windows behavior today.

HOW IS LOCAL DATA SECURED?

Local data is secured against the two-primary means of attack – decrypting data and copying encrypted data. First, all authentication data is encrypted using a diversified key. The entire XML database residing on the machine is encrypted by a second unique application key. Therefore, sensitive data is essentially encrypted twice by two unrelated keys.

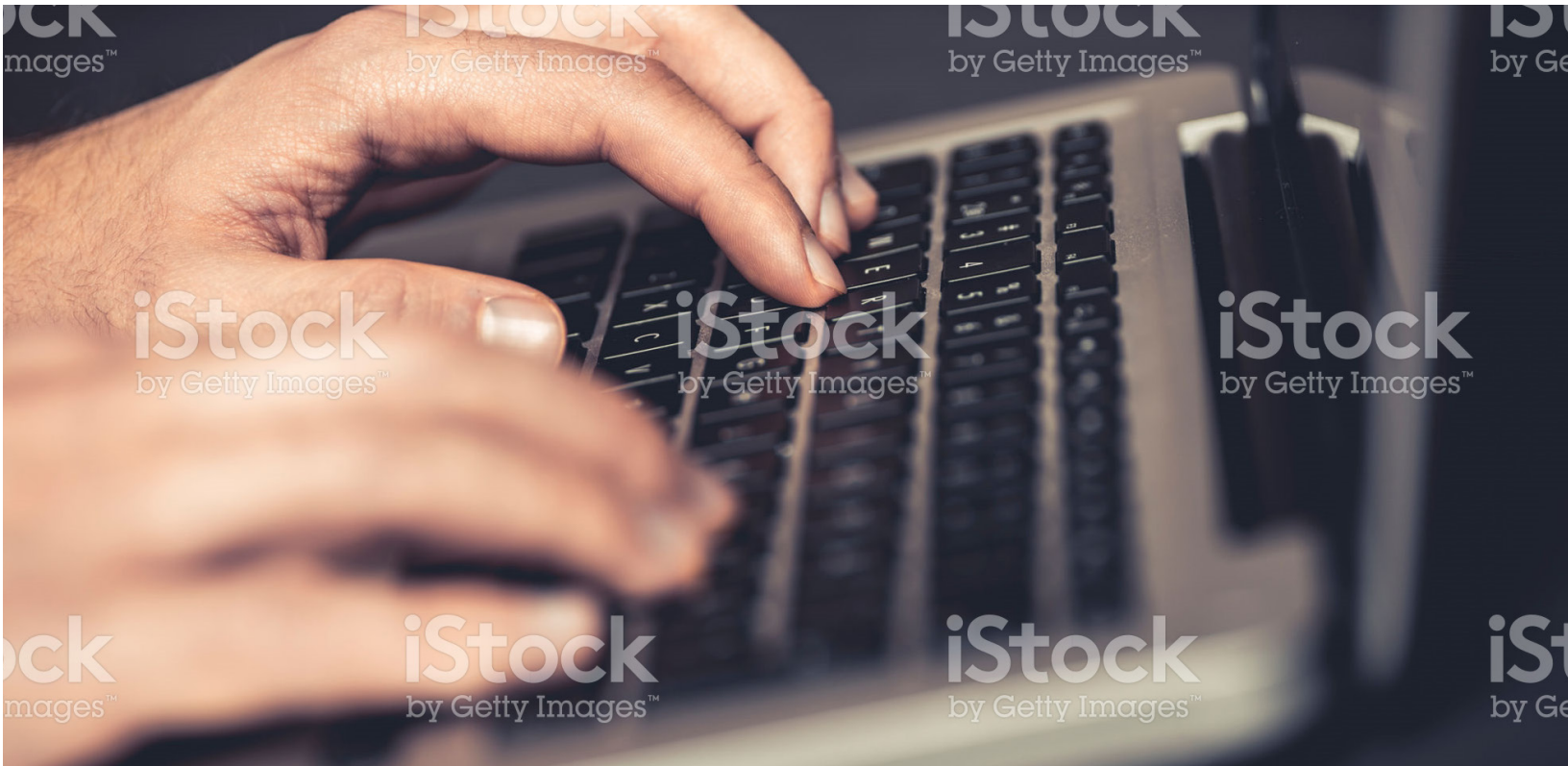
Second, local data is secured from copying encrypted data. Keys to encrypted authentication data are also 'salted'. In the 'salting' process, extra data is fed as input into the algorithm to ensure that although the same key is used to encrypt different items, each encryption result will look unique from the others when it is stored in the database. For example, if the same PIN is assigned to two different cards, each encrypted string will look different in the stored form. Therefore, a potential hacker cannot copy the encrypted string from the first card and gain access with the second card.

HOW IS THE DATA SYNCHRONIZED?

The synchronization process occurs through the use of web services between RapidIdentity Server and RapidIdentity Client. Synchronization only occurs at key events when required by the system. Such events may include PIN changes, updated answers or a new enrollment. To optimize the speed of the application, RapidIdentity Client does not have to communicate with the server at every logon. Data is only synchronized for users who have previously authenticated to a particular machine while on the network or have a local machine account, in keeping with the Microsoft standard logon processes.

HOW IS SYNCHRONIZATION BETWEEN THE CLIENT AND SERVER APPLICATIONS SECURED?

Synchronization is secured through WSE 3.0 and authenticated web service calls. All sensitive data that passes between the client and server is encrypted with the client key and stored securely in the database with the server key. RapidIdentity may be configured for SSL if an additional layer of security is desired. In the event SSL is used, the web services are done utilizing SSL encryption across the wire. This effectively results in two layers of encryptions performed on the data.



AUTHENTICATION DATA

HOW IS Q&A DATA SECURED?

User answers are treated as sensitive data throughout RapidIdentity, therefore they are encrypted in all places, including during synchronization. Policies can be set through RapidIdentity Server to enforce answers to be a certain length and unique from one another. To increase the usability of the question and answer logon method, when validating user answers, all white space is stripped and all capitalization disregarded. Therefore, user answers are not dependent on spaces and are not case sensitive.

DOES RAPIDIDENTITY CHANGE A SMART CARD PIN UNBLOCKING KEY (PUK)?

By default, RapidIdentity does not change the PUK. Within RapidIdentity Server settings, the system can be configured to change the PUK from the default provided by the manufacturer, to one randomly generated by RapidIdentity.

HOW DOES RAPIDIDENTITY PROTECT STORED PUKS?

RapidIdentity protects stored PUKs by using industry best practices for security. There are three aspects of PUK security which RapidIdentity works to enforce. The first has to deal with PUK diversification (or randomization). Basically, it is never a good idea to allow one PUK to be the "master PUK" for all smart cards or other tokens deployed in the organization. By enabling "PUK Rollover", each token's PUK is changed to a new PUK on enrollment. This new PUK is unique* to that card and cannot be determined based on known information (such as another PUK or the default PUK in the system).

[***NOTE**: *the PUK is not necessarily guaranteed to be unique across all cards deployed. While the probability of correctly guessing the PUK is nearly 1 in 2x10^15, the likelihood of the same PUK being utilized by two different people is nearly 1 in 2x10^8, both statistically highly improbable*.]

The second aspect of PUK security is actually encrypting the PUK itself within the data repository. By using best practices of encryption (including adding entropy to the encryption algorithm to guarantee that keys used are not revealed with known data), the security of the PUK itself for that user's token is protected (See "How is local data secured?"). Furthermore, since the PUK is treated as sensitive

information within the data repository, all the layers of security at work in securing the repository act as additional layers of security for the PUK (see "How is the RapidIdentity data repository protected?").

Third, the PUK is protected by the smart card itself. The smart card enforces only a few allowed attempts at the PUK before the PUK becomes blocked and can never be verified. By allowing only 5 attempts (or something similar) for the PUK to be presented, it becomes near impossible to uncover the PUK through brute force. This also highlights why RapidIdentity is such an integral part of any security-token based deployment: PUK management. A smart card that has had the PUK blocked can never have its PIN unblocked again. Therefore, if that card's PIN becomes blocked, it is no longer useful as a logon device. RapidIdentity takes great care to ensure that not only are all PUKs protected, but that each smart card-PUK combination is carefully managed, to extend the life of all smart cards and other tokens within the organization.

DOES RAPIDIDENTITY STORE USER PINS?

Yes and no. When used with a contact smart card, RapidIdentity does not store the user's PIN. When used with RFID, fingerprint biometrics, OTP, or magnetic swipe/2D barcode, RapidIdentity must store the user's PIN.

DOES RAPIDIDENTITY CACHE USER PINS?

Yes and no. RapidIdentity does not store smart card PINs, but can store PINs used for other authentication methods, such as RFID. Smart card middleware is capable of caching PINs. In the event you desire to store or cache the PIN for SSO-like functionality; however, we recommend avoiding doing so.

HOW IS THE RAPIDIDENTITY DATA REPOSITORY PROTECTED?

The RapidIdentity data repository is protected using various layers of protection. The first layer of protection is afforded by the repository itself. For example, if Microsoft SQL Server is utilized as the data repository of choice, then all of the features of extrapolating the data from the application and the security measures in affect for protecting SQL Server apply as well to the RapidIdentity data

within. The repository administrator (e.g. DB Admin for the SQL Server) can utilize a full range of features to ensure appropriate restrictions on account access for the accounts utilized in interacting with the repository. In addition to this, the configuration information for the repository in RapidIdentity is not accessible across the web. IIS has strong safeguards in place to protect the access of this information, and RapidIdentity further works to ensure appropriate access is enforced when accessing the web application. Finally, sensitive data within the repository itself is always encrypted, utilizing the application key and many other diversification/security algorithms employed in protecting the data within. This three-tiered approach to data and repository security is in keeping with industry best practices for security applications.

DOES RAPIDIDENTITY FOLLOW THE GLOBALPLATFORM SPECIFICATION FOR JAVA SMART CARDS?

Yes.

CAN SOMEONE WHO STEALS MY CARD GET MY WINDOWS® PASSWORD?

No. RapidIdentity does not write Windows® password credentials to the card. Card logon occurs as described above. However, the user can still logon to multiple machines through synchronization with RapidIdentity Server.

SECURITY & ENCRYPTION

WHAT CRYPTOGRAPHIC ALGORITHMS ARE USED?

Out-of-the-box, RapidIdentity supports and utilizes AES, DES, 3DES, RSA, SHA-256, and MD5 for cryptography within the applications.

CAN THE APPLICATION SUPPORT OTHER OR CUSTOM CRYPTOGRAPHIC ALGORITHMS?

The architecture of the applications allows for ease of extension to new protection mechanisms (such as AES or Elliptical Curve algorithms) via an internal API. These extensions require custom dll's and may be provided by Identity Automation Professional Services.

CAN I ENABLE SSL ON THE SERVER?

RapidIdentity may be configured for SSL if an additional layer of security is desired. In the event SSL is used, all interactions with RapidIdentity Server, including web service calls for synchronization, and all administrative and user functions are encrypted with SSL. Instructions for enabling SSL are included in product documentation.

HOW IS THE DATA REMOVED FROM MEMORY?

Sensitive data is encrypted always, except for the exact time it is needed for the 'just-in-time' algorithms. Immediate removal is done using a secure wipe of those addresses in memory. This ensures that the data is not left residing in decrypted form after it is no longer needed. The technique specifically protects against in-memory attacks as well as those against the RAM chip. It is like the techniques used by the U.S. Department of Defense for wiping memory.

HOW IS AUDIT DATA SECURED ON THE SERVER?

RapidIdentity signs and timestamps audit data for tamper-proof verifications. This audit data is stored in the RapidIdentity Server database. Best practices should always be followed for securing databases, such as limiting the ability for external users/accounts to delete information.
This is particularly important for the audit logs, as well as all the database information for RapidIdentity Server.

SUPPORT

WHAT IS THE ONGOING PROCESS FOR UPDATING / PROTECTING THIS APPLICATION? HOW DO YOU PLAN FOR AND ACTIVELY PROTECT AGAINST POTENTIAL HACKS OR FUTURE SECURITY THREATS?

Identity Automation continues to do ongoing threat analysis to ensure that all its products are secure. As soon as threats or other issues are identified, they are addressed and updates made immediately available. By participating in Identity Automation's Maintenance & Support program, customers are automatically notified about any new updates that are made available.

HOW ARE BUGS REPORTED?

Identity Automation provides a support portal through which all bugs may be reported. Bugs may also be reported through the telephone support line.

THIRD-PARTY ASSESSMENTS

WHAT LEVEL OF QUALITY ASSURANCE IS RAPIDIDENTITY SUBJECTED TO?

All RapidIdentity products are subjected to an aggressive and comprehensive Software Quality Assurance process. Following a successful internal assessment, the products are released to a third-party Software Quality Assurance process where the products undergo manual and automated tested using best practices for Software Quality Assurance. Finally, the products are released to third-party BETA customers where the products spend a number of months' being subjected to more than thirty varied customer environments from around the global. Should additional development be required following the third-party beta the process starts anew.

WHAT LEVEL OF THIRD-PARTY ASSESSMENT DOES RAPIDIDENTITY UNDERGO?

At specific points in the development and Software Quality Assurance processes, and prior to general availability, RapidIdentity applications are submitted to Veracode (www.veracode.com) where the applications undergo comprehensive Static and Dynamic automated scans to identify any code level security vulnerabilities.

Contact Sales: sales@identityautomation.com
Contact Support: support@identityautomation.com
Other information: info@identityautomation.com

Toll Free: 877-221-8401
Voice: 281-220-0021
Fax: 281-817-5579

Corporate Headquarters:
8833 N. Sam Houston Pkwy. W.
Houston, TX 77064

IDENTITY
AUTOMATION