AT LAST *a Two-factor authentication*
without the hassle factor

Evaluating Strong Authentication Systems

Introduction & Goals

The purpose of this document is to provide the information required for you to evaluate the WiKID Authentication System on its financial, technical and operational merits.

There are a number of drivers for adopting strong authentication: risks are increasing, the number of connected systems is increasing, the number of users requiring remote access is increasing, and regulations such as Graham-Leach-Bliley, PCI, HIPAA, and various state disclosure laws are affecting more companies. Authentication is a key pillar for secure systems and enterprises are more and more aware that passwords are not a strong enough form of authentication. They have looked at hardware-based tokens and other solutions, but have not significantly adopted strong authentication.

There are *three reasons why hardware tokens have failed* to significantly penetrate the authentication market: **convenience, extensibility, and cost**. Users tend to dislike having the carry the additional hardware. If it's lost or left at home, it is very inconvenient. Further, tokens are a micro-point solution: they are only good for one authentication system at one enterprise. They cannot be used across multiple enterprises, nor are they very extensible inside the enterprise.

The WiKID Authentication System addresses these concerns with a new approach built for today's authentication needs. The WiKID Authentication System is easy to implement and maintain and it is highly extensible.

Two-factor Authentication Architectures

There are three factors of authentication: something you know, something you have and something you are. Weak authentication only requires one factor, strong authentication more than one. Passwords are the most dominant form of weak authentication. ATM cards, the most ubiquitous two-factor system, require both knowledge of the PIN and possession of the card. WiKID Systems has developed a unique, patent-pending architecture that dramatically simplifies two-factor authentication and creates an extensible platform for future needs.

Historically, there are three types of strong authentication architectures for reader-less single-purpose hardware devices ("tokens"):

- Challenge-response – A challenge is issued and the response must match the expected response.
- Time-synchronous – The passcode changes periodically and must match the expected passcode. The system must deal with clock drift, usually by allowing multiple codes to be valid at any given time, reducing security. Time-synchronous soft-tokens are vulnerable to the generation of future valid-codes by moving the device clock forward.
- Event-synchronous – A counter on the device generates the next valid passcode, which must match the expected passcode on the server. The server must be able to "hunt" for a future valid passcode in case the counter on the device is moved ahead of the counter on the server.

While these solutions were acceptable in their day, they are expensive, costly to set-up and maintain, and not extensible for today's authentication needs.

How WiKID Works:

The WiKID Authentication System is based on a "***request-response"*** architecture. When the user wants to login to a service, they enter a PIN into the device; it is encrypted and sent as part of a request to the WiKID Authentication Server (WAS). The WAS checks the encryption, validates the PIN and if the account is active and enabled, responds with an encrypted passcode. The device decrypts the passcode and the user enters it into, say, their VPN service, which in turn validates it with the WAS. If the code matches, the user is granted access. When the client doesn't have Internet access such as when a cell phone is out of coverage, the client falls back into a challenge-response mode.

Development & Security Philosophy

Our goals in developing the WiKID Authentication System are as follows:

- The system should be as secure as existing two-factor authentication solutions
- The system should be easier to use, maintain and administer than existing two-factor authentication systems
- The system should be less expensive in TCO than existing authentication systems – either passwords or tokens, relying on self-service as much as securely possible
- The system should be highly extensible, addressing multiple authentication needs as they arrive

It is our belief that a company can increase security while simultaneously reducing costs. We have designed and developed our system with these goals in mind.

The WiKID Architecture

The WiKID Authentication System consists of the WiKID Authentication Server (WAS), the WiKID Token Client and various protocol modules that connect to network clients such as a RADIUS server, firewall, LDAP directory, Domain servers, etc.

The WiKID Authentication Server

The WiKID Authentication Server is a hardened Linux software appliance available as an ISO or a VMWare image. The WAS handles requests from network clients (while a network client may be a server, it is a client to the WAS), authentication (passcode) requests, logging, reporting, user management, certificate management, WiKID Domain creation, protocol module management and administrative preferences. The WAS is running a firewall and is not accessible via Ping.

WiKID Domains

Each instance of the WAS runs under a particular security domain. The security domain is intended to segregate users with respect to access and services. For example, Intranet access may be provided with one domain, partner Extranet access with another and public Internet (Website) access with a third. Separate security policies can be provided for each domain and access can be granted on a device/individual user basis. The security policy includes PIN length, max bad PIN attempts, max bad passcode attempts, passcode lifetime, and max number of consecutive offline challenge-response logins. Upon creation, each domain generates a key pair for encryption within the passcode request/passcode reception process.

Each domain is represented by a 12-digit code, which represents the zero-padded IP address of the server (or zeropaddedipaddress.wikidsystems.net). The domains have both a device name and a server name, so that the Client can have a domain that is "VPN" on the client but refer to "Executive VPN" on the server.

The WiKID Token Client

The WiKID Token Client runs on a PC or Internet-enabled wireless device (Blackberry, J2ME phone, Palm device or PocketPC). The client generates the public/private key pair and maintains the domain connection information. It does not store the PIN (PINs are stored on the WAS). The client can add new domains, delete domains and select domains for passcode requests. Unlike other two-factor systems, the WiKID Token Client for each domain is the same, whether the domain is on the same WAS or for a completely different company. This capability makes WiKID perfect for cross-enterprise strong authentication, which is increasingly important for today's extended supply chains.

Protocol Modules

The WAS communicates to various network clients (VPN concentrators, firewalls, servers, etc) via protocol modules. Currently, WiKID supports LDAP, TACACS+ and RADIUS and we provide COM, EJB, PHP, Ruby and Python interfaces for custom application integration. The EJB interface is known as the "wAUTH protocol" and it is used to provide payload and transport encryption. wAUTH is automatically enabled on all WiKID Authentication Servers as it is used for all local host protocol modules.

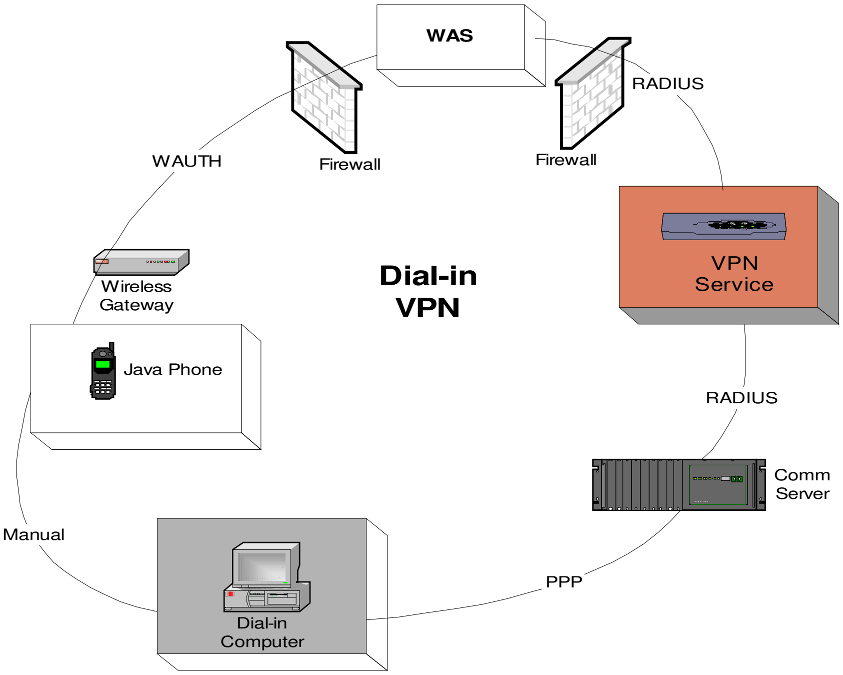
Network Client

Network clients are simply the devices on the network responsible for granting access to the users: RADIUS servers, VPN concentrators, firewalls, routers, switches, applications, etc. They accept the username and passcode from the user, send the passcode to the WAS, and, if the WAS validates the code, they grant access. For a network client to be active on a security domain, it must be registered on the WAS.

Below is a sample diagram of a fully configured WIKID System:

Example Scenario

In this scenario, Bill Jones, an employee of a telecom company, is the administrator of several systems within the company's headquarters building. Bill requires VPN access to his systems late at night and on the weekends from home. Bill has a Java-enabled phone and a personal computer at his house.



The components of the system are as follows:

- The WiKID Authentication Server is, as always, the WAS.
- The Protocol Module is RADIUS. It is important to note that RADIUS is used on a *trusted* network in this case.
- The domain is the telecom's corporate admin VPN.
- The Wireless Devices is Bill's Java Phone.
- The Network Client is the corporate VPN service (provided by a Cisco device).
- The Terminal Service is Bill's home computer equipped with a modem.
- The additional devices provide access (the wireless gateway, the communication server) and security services (the firewalls).

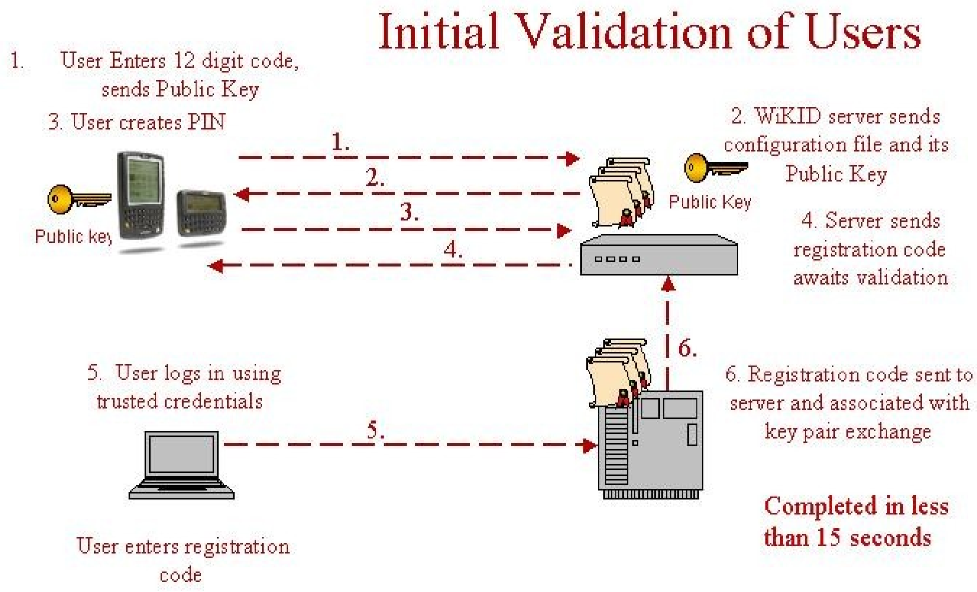
Operational Considerations

One of the reasons why two-factor authentication has failed to deeply penetrate the authentication market is due to the operational hassles of hardware tokens. WiKID Systems has eliminated the largest aggravations in deploying, managing and using strong authentication. By enabling employee self-service to the fullest extent possible, deploying a WiKID Authentication System is easy, cost-effective and secure.

Deployment & Initial Validation

The WiKID Authentication System is the only solution where initial validation can be completed automatically – 100% self-service by the end-user. The WiKID Token Client as shipped contains no security information, only the ability to create the public/private key pair, so this client can be installed anywhere by anyone. (Most devices now support over-the-air download, making software deployment incredibly simple.) It is not until after the key pairs are exchanged between the WiKID Token Client and the WiKID Authentication Server *and* after that exchange has been validated through a second, trusted channel that a security relationship has been created.

Below is a diagram of how the initial validation can work in full self-service mode:



Initial Validation of Users
1. User Enters 12 digit code, sends Public Key
3. User creates PIN
2. WiKID server sends configuration file and its Public Key
Public key
Public Key
1.
2.
3.
4.
4. Server sends registration code awaits validation
6.
5. User logs in using trusted credentials
5.
6. Registration code sent to server and associated with key pair exchange
User enters registration code
Completed in less than 15 seconds

- The user enters the 12-digit WiKID Domain identifier. The WiKID Token Client creates the public/private key pair and sends its public key to the server requesting a configuration file for the domain.
- The WiKID Authentication Server responds with the configuration file and its public key, encrypted by the Client's public key.
- The User is prompted for a PIN and told the minimum PIN length for that domain. The PIN is encrypted and sent to the Server.
- The Server decrypts the PIN and stores it. It returns a one-time registration code. The account has now been created, but is not validated. In order to validate the account, the WiKID Authentication Server must receive the same registration code from a trusted Network Client over a second channel.
- The end-user logs into the trusted Network Client, perhaps using an existing hardware token or, more likely, they log on their LAN (a trusted network) using their existing LAN credentials. WiKID will supply ASP scripts that run on IIS using Active Directory credentials. The user logs into the IIS server and enters the registration code.
- The registration code is sent to the WiKID Authentication Server and the account is activated.

Alternatively, the administrator can manually enable the user. Please note that the end-user can re-initialize the same way should they forget their PIN.

With hard-tokens, the device must be associated with the user on the server and physically delivered to the user, a process that costs $15 or more.

Management

The web-based interface of the WiKID Authentication server is intuitive and easy to use. Security professionals will appreciate working on security, rather than the logistics of token management.

The WiKID Token Client is easily replaceable. There is no box of hardware tokens with limited lifetimes locked up in the administrator's office. If a device is lost, it is much easier to replace software.

Convenience & Ease of use

WiKID Systems is committed to ease-of-use for the end-users and the administrators. Employees dislike having to carry additional hardware, such as hardware tokens. Most employees tend to like self-service applications and to dislike calling helpdesks. WiKID automated initial validation system is simple for employees to use. Our ability to reset LAN passwords is a perfect self-service application.

Portability – works anywhere

A key requirement for mobile workers is to be able to log in from any location, even from a kiosk or other non-company owned PC. WiKID provides the online request-response mode and falls back to the challenge-response mode when out of wireless coverage. It requires no reader and works everywhere.

Extensibility

Counter-based and time-synchronous tokens have a one-to-one relationship with their authentication server. The WiKID Token Client, can have relationships with multiple WiKID Authentication Servers. This extensibility can save money internally and increase security externally.

Increasingly, companies are opening up their networks and applications to suppliers, vendors, consultants and other 3rd parties. While these tight ties have increased productivity and smoothed supply chains, they have also increased security risks. Do your vendors use strong authentication to log into your network? Do your employees use strong authentication when they log into your vendor's networks? The ease of deployment and low price-point of WiKID makes it practical to deploy strong authentication through the whole supply chain. Moreover, end-users can use the same client for both companies. Your vendors will see the same financial benefits as you do, increase the value of the entire chain.

Scalability

Each request and response on the system is a mere 251 bytes. This small transaction size allows the server to handle a huge number of users. Our stress tests have demonstrated that a 1.4ghz, 256 meg ram, IDE-based server can process *50 authentications per second*. Multiple servers are more for fault-tolerance than scalability.

Mutual Authentication

Mutual authentication is really **site or host authentication** to the user combined with user authentication to the site. WiKID uses a hash of the server certificate stored on the authentication server to perform site authentication. When the user requests an OTP, the hash is also sent to the token client. Before presenting the user with the OTP, the token client fetches the certificate from the website, hashes it and compares it to the retrieved hash. If the hashes match, the URL is presented as validated and the default browser is launched to that URL. This method leverages the security and investment in SSL certificates and provides a consistent session and mutual authentication method to the user and it will eliminate 99% of phishing attacks.

Transaction Authentication

Transactional authentication is equivalent to *digitally signing a transaction with a one-time passcode*. For example, when a user wishes to make a suspicious transaction, such as a one-time, large payment to a new payee, they should enter a second one-time passcode to validate the transaction. ***It is critical that the transactional authentication be cryptographically distinct from the session authentication mechanism*** or the attacker will try to get the user to re-authenticate for the session. A simple notice saying that the "Connection was lost, please re-authenticate" will get most users to enter a new one-time password.

This requirement highlights a key difference between shared-secret systems and the WiKID Strong Authentication System. WiKID can support multiple authentications domains with no reduction in security. One WiKID domain can be for sessions and another for transactions or a user could have more than one key pair on separate devices. For example, they might have a session token on their PC and a transaction token on their cell phone.

Future capabilities

Already WiKID Systems is the first authentication system to combine two-factor authentication with a password reset capability out of the box. The WiKID architecture provides for robust future flexibility through new clients, new protocol modules, new network clients and new technologies.

Security/Technology

There are many flavors of two-factor authentication, some more secure than others. We believe that relative security is a very central factor in choosing an authentication solution. While cost savings, extensibility and manageability are important, without security, you don't want to create a false sense of security.

Relative Security Analysis Chart

Item	Notes
Two-factor	The combination of PIN and device is very strong
Passcodes random	There is no way to predict the passcodes or to brute-force attack the server
Passcode length random	Randomizing the passcode length protects against a race attack/man-in-the-middle attack on a fixed length response system (future release).
Only one passcode valid at any moment	Passcode lifetime can be set per domain by the administrator, which can't be done with a time-synchronous system.
Eliminates shoulder surfing, keyboard sniffers, Trojans	Passcode is only used once.
PINs and passcodes never sent over network together	In some systems, the PIN is appended with the passcode, which increases the risk of PIN compromise. With WiKID, the PIN and passcode are never transmitted together and are always asymmetrically encrypted.
Published algorithm	WiKID uses only published algorithms, increasing the security of the system through peer-review process.
Risk from loss	Users more likely to keep wireless device separate from laptop, decreasing risk of combined loss. Tokens are often kept with laptop. A lost or stolen token is a nuisance. A lost cell phone is a financial risk for the user, aligning incentives.
No password file for attackers to target	Password files are the gold mine for attackers. WiKID removes that target.
PIN stored on server	There is no way to brute force attack the PIN as it is stored safely on the WiKID Authentication Server. Certificates protected by passwords are subject to cloning and brute-force attacks on the password.
Network clients require a WiKID Authentication Server Certificate	Prevents a Denial-of-service attack from un-approved Network Clients.
Domain Security Options	Maximum bad PIN attempts Maximum bad passcode attempts Maximum consecutive challenge-response logins PIN length configurable Passcode lifetime
Cross-enterprise security	There is no reduction in security when multiple domains are created making cross-enterprise strong authentication viable for the first time. This capability fits well with Single Sign-On efforts such as Liberty Alliance.
Logging	Complete logging and reporting. Integration via Syslog is available.
Transaction Authentication	Built in ability to handle transaction authentication with cryptographically distinct keys.
Mutual Authentication	WiKID validates the SSL certificate for the user and launches the browser to the correct site.

Financial Considerations

Potential Cost Savings

The Request-Response architecture provides a flexible, extensible platform that significantly increases the cost savings potential from implementing the WiKID Authentication System.   Wherever possible, WiKID has enabled self-service and automation and we have attacked the costs associated with authentication.

The cost of hardware tokens

The vast majority of companies still rely on usernames and passwords for remote authentication. Some companies have deployed one-time passwords systems, especially time-synchronous, hardware based tokens, but the majority of deployments are very small and the total penetration is minimal. Each token costs money; it is expensive (and an administrative hassle) to distribute tokens; the tokens have a limited life; the authentication servers are expensive both upfront and in ongoing costs.

Below is a breakdown of total authentication costs for two companies WiKID's prcing is available at our website: http://www.wikidsystems.com/features/financial

3 Year Key Fobs

Number of users	**10**	**250**
Number of Years	**3**	**3**
Cost per 3 Year Token	$54.00	$54.00
Total Token Cost	$540.00	$13,500.00
Server Cost	$1,140.80	$20,470.00
Server Maintenance @ 20% per year	$684.48	$12,282.00
Server Hardware	$1,100.00	$1,100.00
Token Distribution & Validation cost per Gartner	$20.00	$20.00
Total Upfront Distribution	$200.00	$5,000.00
Total Costs for key fobs	**$3,665.28**	**$52,352.00**

WiKID Systems

Server Hardware	$1,100.00	$1,100.00
Software License Fees (with pre-pay discount)	$504.00	$12,600.00
Total Costs for WiKID	**$1,604.00**	**$13,700.00**
Savings	***$2,061.28***	***$38,652.00***

This analysis does not include the potential cost of lost hardware tokens, which have to be re-purchased. WiKID is based on seat licenses, so new tokens can be generated without penalty.

By combining strong authentication, self-service and automation into a secure authentication platform, WiKID Systems can save an enterprise money while increasing security.

Conclusion

More than ever, companies are seeking to get more for their money. Security traditionally has been viewed as a cost center that mitigates risk primarily. WiKID Systems takes the approach that there are always ways to improve systems to eliminate costs that can also improve security. The WiKID Authentication System embodies this belief by attacking authentication costs and risks simultaneously and providing automated self-service capabilities that eliminate the overhead of typical strong authentication systems. The WiKID Authentication System improves upon traditional authentication systems operationally, in security capabilities and in financial terms.

WiKID is the first company to create a fully automated, self-service initialization solution; the first to deliver a two-factor authentication system that is capable of easily, and the first to provide strong mutual authentication in a single package. We intend to continue this level of innovation.

A final note

We have also tried to make it very simple to test and purchase WiKID.

You can download a 30-day evaluation copy of the WAS in ISO or VMWare image format here:
http://www.wikidsystems.com/registration/DownloadTheServer/?reviewerguide

You can see our pricing here:
http://www.wikidsystems.com/product-info-downloads/financial/?reviewerguide

You can purchase online directly from WiKID here:
http://www.wikidsystems.com/simplecartitem/?reviewerguide

You can download the WiKID token client here:
http://www.wikidsystems.com/product-info-downloads/token-clients/?reviewerguide

You can test the WiKID token client on this page:
http://www.wikidsystems.com/signup/example.jsp/

You download documentation from our extranet here:
http://www.wikidsystems.com/signup/index.jsp