SECURECHANNEL

AI BASED
MOBILE SECURITY & API SECURITY SOLUTION
WITH
RISK ENGINES

Looking Beyond Future...

Cybernet Security Solutions LLP

www.cybernetsecurityinc.com

Security Objective And Solutions

	Security Objective	Solutions	Protection Coverage
Business/User Security (Primary)	❑ No fraudulent transaction ❑ No customer data theft ❑ No authentication data theft/reuse **Business Impact: VERY HIGH OR UNLIMTED**	SECURECHANNEL	**ONLY** solution that meets these objectives, whether transactions are executed from organization app or third party apps. Only solution that protects against misuse of stolen data from original app.
App Security (Secondary)	❑ **Protecting App** ❑ **Protecting Device** ❑ **Protecting Code** **Business Impact: NEGLIGIBLE OR NOTIONAL**	SECURECHANNEL RSA Protectt.ai ivanti Symantec. ZIMPERIUM. MOBILE THREAT DEFENSE UNIKEN We make connecting safe pradeo appsealing CRYPTOMATHIC IBM KASPERSKY lab	They protect only organization app and **NOT** transactions executed from original app or third party apps.

How Fraudulent Transactions Can be Carried Out Easily?

- A phishing app can be distributed to user stealing User-Id/Password & OTP from user's phone. Send this data to hacker using API call. This phishing can be a modified copy of the original app or independently developed.

- This stolen data can be used by any rogue app or any non-app program, calling required APIs to login and further carrying out fraudulent transactions as user and registered user device.

- There is no technology in the world that can prevent a person from developing a phishing app, uploading it on Playstore/Appstore, distributing it to users, user installing it on his/her phone and user using it. Impossible for an organization to detect it.

- **The weakest link is API (A public component) and NOT App. As long as one can call API with right payload, server will execute the transaction.**

- **Protecting organization app NO way prevents hacker from calling these APIs. Mere app protection provides a false security.**

Protection From API exploitation/Rogue APP

A strong Anti-tampering check on the server side and randomized non-reusable payload data are the only security features that can prevent exploitation of APIs.

All Other Solutions	SecureChannel
It calculates a static app checksum value of app which is sent to organization app sever as an additional data during API call. This data is checked on the server side for detecting tampering.	It calculates dynamic app checksum value of app using 30 odd internal parameters. These parameters can not be influenced externally. This value is passed to server as part of virtual image. Virtual image changes every second. It is device, app, server and network specific. Its life is just a few seconds and one-time use. This dynamic data is passed to server for verification as part of API payload.
Simple Hacking 1. Using a software like Fiddler, trace request data of https request. Get anti-tampering value and reuse it by passing as part of payload for carrying out fraudulent transaction. 2. Modify original apk, substitute signature and checksum from the original app in modified app. App will always generate a valid checksum. 3. All protections can be simply bypassed by commenting invoking code. 4. Even code written in C++ can be easily changed by modifying underlying java classes.	**Impossible Hacking** 1. It's dynamic VPN and time based encryption makes virtually impossible to steal and reuse data that is changing every second, has short life and is one time-use. Also if stolen(though practically impossible), data can not be used from any other device or any other network. Thus, even third party apps can not cause any damage. 2. A modified original app immediately closes down even before transaction is executed. 3. It creates on-time use time sensitive dynamic payload. API only accepts this dynamic payload. Thus, these APIs can not be exploited. 4. Dynamic VPN binds network, ensuring no data leakage. 5. Dynamic C++ code with dynamically generated, frequently changing private data ensures it can not be changed by modifying underlying objects.
Hacking Effort: Maximum One Hour – One Time	**Hacking Effort:** Years. This has to be repeated on each mobile and for each transaction.
Impact: Once broken, all users of rogue app can become victim. It DOES NOT prevent payload from reuse.	**Impact:** It PREVENTS creation or reuse of valid payload of API call. Hence it becomes impossible to exploit APIs.

Protection From Frauds From Original App

Security Features	All Other Solutions	SecureChannel
Auto-adaptive authentication	✗	✓
Multilayer security	✗	✓
Self-learning security	✗	✓
Transaction tracing	✗	✓
Transaction level controls	✗	✓
Device/IP/User level controls	✗	✓
Authentication limits	✗	✓
Automatic access controls (locking/unlocking)	✗	✓
Real-time pre-fact fraud detection	✗	✓
Real-time pre-fact risk management	✗	✓
Transaction limits	✗	✓
Transaction delay	✗	✓
Activity engine	✗	✓
Profiling engine	✗	✓
Device data protection	✗	✓
User data protection	✗	✓
X-linking of security activity with business transaction	✗	✓
Privatize APIs	✗	✓

Solution Highlights At A Glance

SecureChannel

Addressing A Lethal Threat That Nobody Addresses

- Innovative Technology
- Patent Pending
- Cloud Service
- Zero Touch Point To Host Application
- No Privacy Data Sharing
- In-memory Processing
- High Performance Engine
- Successful Implementations

- No Competitor
- Tested & Certified
- Simple Plug-in
- No Change In UI Or Processing
- Covers All Mobile Threats
- Integrated Real Time Risk Engines
- 100% Business Continuity
- Signed Partnerships

Every Business - A Potential Victim With Unlimited Risk

Mobile Security Coverage

SecureChannel

Comprehensive Coverage

A Partial List

- Rogue/Fake App detection
- Remote Desktop Program detection
- Detection of reverse engineering
- Device authentication
- Anti-tampering protection
- Channel security
- Redirection security
- Virtual Private Network

- Malicious app detection
- Blocking spyware piggy-backing
- Encryption Key stealing protection
- Rooting/ Jail-break / Emulator/ Simulator detection
- Root detection bypass framework protection (like Magisk)
- Integrated risk/fraud engines

Simple plug-in. Zero change in processing/UI/security

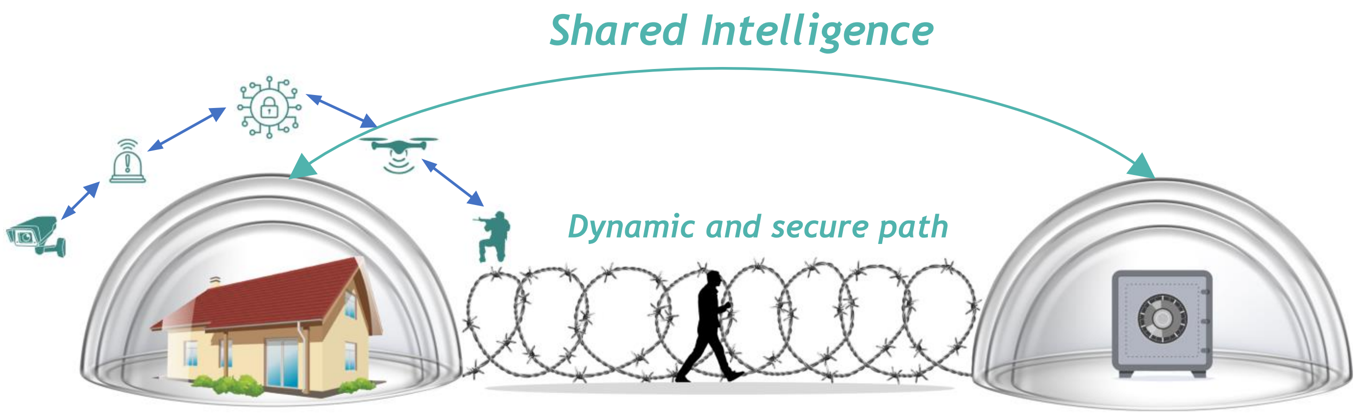
Moving Away From Lock-key Technology to AI Technology

AI Technology - Dynamic, Intelligent, Behavioral, Invisible, Proactive, Deterrent Technology

Easily breakable lock-key based security

A **common** key

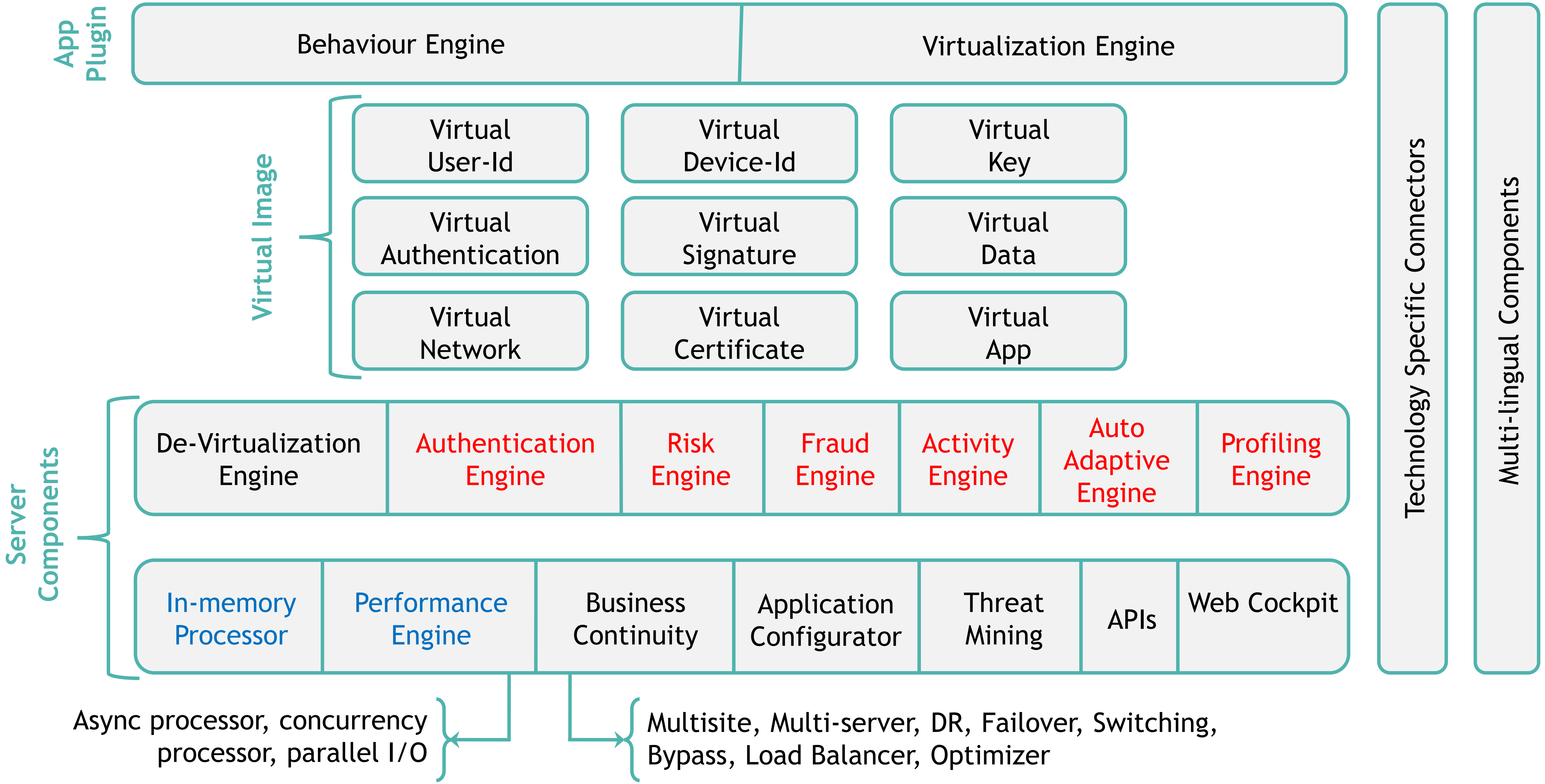
ALL houses can be opened with the same key.



Un-penetrable, multi-layered security

This lock-key-less AI based security, based on shared intelligence and behavior, generates **dynamic** and **private images**, which allows **ONLY** the owner to enter and **ONLY** the owner to open a safe.

Mobile Solution Stack



App Plugin

Behaviour Engine

Virtualization Engine

Virtual Image

Virtual User-Id

Virtual Device-Id

Virtual Key

Virtual Authentication

Virtual Signature

Virtual Data

Virtual Network

Virtual Certificate

Virtual App

Server Components

De-Virtualization Engine

Authentication Engine

Risk Engine

Fraud Engine

Activity Engine

Auto Adaptive Engine

Profiling Engine

In-memory Processor

Performance Engine

Business Continuity

Application Configurator

Threat Mining

APIs

Web Cockpit

Technology Specific Connectors

Multi-lingual Components

Async processor, concurrency processor, parallel I/O

Multisite, Multi-server, DR, Failover, Switching, Bypass, Load Balancer, Optimizer

Virtualization - The Difference-maker

SecureChannel

Base Vulnerability	SecureChannel Protection	Protection Description
Device Id can be easily stolen	**Virtual Device Id**	Dynamically generated device id that can’t be stolen and used. This Id is neither derived from device property nor stored anywhere.
App signature can be stolen	**Virtual Signature**	A signature based on multiple relative chains of 30+ internal dependent properties. It is not an external property of objects; hence it can't be stolen and reused
Encryption keys can be stolen. Any or all data can be modified.	**Virtual Data**	Highly dynamic, short-lived, and one-time use data encrypted using device time-based encryption keys, which makes it impossible to steal and reuse. Neither time nor encryption keys are shared with the server.
Authenticate data can be stolen and substituted	**Virtual Authentication**	Based on the virtualization technology, authentication data is dynamically generated. This data is time based; it changes every second, and has limited life. Thus, it becomes impossible to create, steal and/or reuse dynamic image of authentication data.
SSL Pinning certificate can be stolen	**Virtual Certificate**	Dynamically generated certificate, hence it can't be stolen and used. It prevents data redirection.
Data can be stolen and redirected	**Virtual Dynamic Private Network**	Creates dynamic private network between device and server which ensures data usage from generating device only. Thus, stolen data can’t be used.
User Id can be stolen	**Virtual User Id**	Generates unique virtual user id. Thus maintaining user privacy.
App behaviour can be changed	**Virtual App**	Creates an image of the app based on internal properties which are tracked and authenticated.
Encryption key can be stolen	**Virtual Key**	Generates private and dynamic encryption key which can not be stolen.

Our Proprietary Virtualization Technology – Patent Pending

Based on the intelligence inputs from the server and the inputs from the behaviour engine of SecureChannel plugin introduced in mobile app, the plug-in creates a dynamic virtual image of business data, authentication data, app data, server data, device data, and network data which changes every second.

- These images are created using mobile time-based encryption keys.
- Neither time nor encryption keys are ever shared with the SecureChannel server.
- These images are private and are relative to the internal properties of the device, the network and the application.
- They are one-time use, short-lived and can be submitted from the generating device only.
- The SecureChannel server can calculate the exact time this virtual image was generated on a user’s mobile device.
- This dynamic image is processed by SecureChannel cloud-based de-virtualization engine to get the real image by dynamically generating an encryption key based on the calculated time.
- This real image is further used for threat and risk detection.
- The behavior engine of mobile plugin blocks all threats at app level.

Towards Impossibility

SecureChannel

Intelligence Based, Server Driven Technology

Proactive Blocking at App

Dynamic behaviour engine **closes** app on a threat.

Practically impossible to misuse.

Impossible App/Data Hacking

Possession of user's phone at the time in the past required.

Phone **switches off** on hacking.

Impossible API Exploitation

Virtually impossible to exploit APIs with dynamic and private images.

Making it 100% Secure

Virtually Impossible to penetrate **Multi-layered** and **self-learning** risk engines.

SecureChannel Virtualization Technology is the ONLY technology that can protect you from all threats, including rogue apps.

No common weak point

The Total Protection

Attempt	Protection
Static cracking of SecureChannel component	▪ It may take decades to analyse millions of lines of dynamic de-assembled C++ components. ▪ Complete understanding requires point-of time access to dynamically generated code and dynamically generated data during run time. ▪ Stealing of a signature or encryption key requires dynamic cracking
Dynamic cracking of SecureChannel component	▪ To understand dynamic parts, a memory dump of exactly the time user clicked on Submit button is required. This is practically impossible ▪ Memory dumping will switch off the phone.
Modification of app code or SecureChannel component.	▪ To understand dynamic parts, tracing statements have to be put at thousands of places and run them thousands of times to get any trace. ▪ But, the app immediately closes on hitting the first change itself.
Use of stolen SecureChannel produced data from the generating mobile	• Capturing generated data without app modification is not practical. • This data is one-time use with a very short life. On re-submission of data , server rejects the request.
Bypassing SecureChannel component	• Bypassing does not produce a virtual image,. • The server component will reject the request in the absence of receiving a virtual image.
Use of SecureChannel component by a third application	• Application will crash and it won't produce required virtual image.
Stealing and reuse of application/business data	• Application/business data goes to server as virtual image and it is handed over the host application by SecureChannel server component. Reuse requires submission of data through a tampered app. Tampering immediately closes the app.

SECURECHANNEL

Thank You

Email - sales@cybernetsecurityinc.com
Web - www.cybernetsecurityinc.com