ARXAN

Arxan Best Practices White Paper

7 Key Factors of a Highly Effective Application Protection Solution

Abstract:
This whitepaper discusses the key factors that enable an effective application protection solution -- that mitigates binary code risks and combats the latest security threats.

TABLE OF CONTENTS

Executive Summary

Digital media, gaming, software providers, financial services, healthcare, retail, hi-tech/telecom and other industries lose millions of dollars every year as a result of piracy, intellectual property (IP) theft, software tampering, malware, unauthorized access and fraud. Malicious hackers can access an application's binary code and compromise it's integrity with widely available tools and techniques. An effective application protection solution is paramount to safeguard the integrity and confidentiality of an application. This whitepaper discusses the key factors that enable an effective application protection solution.

The State of Application Security

As the number of applications running in distributed or untrusted environments continue to rise, so do the frequency, sophistication and severity of security breaches. The following are some examples of applications that run in distributed or untrusted environment:

- Mobile applications (Mobile Banking/Payments, Corporate Apps, Healthcare, Digital Media, Gaming, etc.)
- Packaged software (ISV, Gaming, Digital Media, etc.)
- Embedded software / "Internet of Things" (Connected wearable devices, Connected Homes, Connected Cities & Transportation, etc.)
- Software running in untrusted environments (Cloud / Third-party Datacenter, Emerging Markets, etc.)

Recent research reports from leading industry experts have shown that applications are vulnerable to reverse engineering, repackaging, republishing and susceptible to becoming malicious weapons[1].

Attacks on Intellectual Property (IP) and proprietary information are rising.
Based on the Global State of Information Security® Survey 20152, a worldwide study by PwC, CIO, and CSO, IP theft increased 19% in 2014 compared to 2013.

For healthcare providers and payers, the (Internet of Things) IoT is not futuristic, nor are the risks theoretical.
A third wave of Internet expansion, the IoT, has come through the confluence of people, processes, data, and things. According to PwC survey2, 47% of healthcare provider and payer respondents said they have integrated consumer technologies such as wearable health-monitoring devices. HP reviewed 10 of the most commonly used connected devices and found that 70% contain serious vulnerabilities[3].

Verizon found that 54% of all attacks targeted the application layer, and payment card data was the primary target in 95% of incidents within the retail industry[4].

Compromises by organized crime are on the rise.
A successful attack can net millions of payment card records that can be quickly monetized. In addition to credit and debit card data, these criminals increasingly target patient health care

data or other personally identified information that has considerable value in the underworld of information resellers.

No platform is immune to threats.
Threat vectors are constantly evolving and attacks at the application level are prevalent with increasing frequency, sophistication and severity. (See a list of recent application-focused attacks).

Types Of Attacks On Applications

A few easy steps and widely available static analysis tools and debuggers make it easy for the adversary to access, analyze and reverse-engineer unprotected binary code. Following exhibit highlights some of the main threats that an application is exposed to, once it's deployed with unprotected binary code.

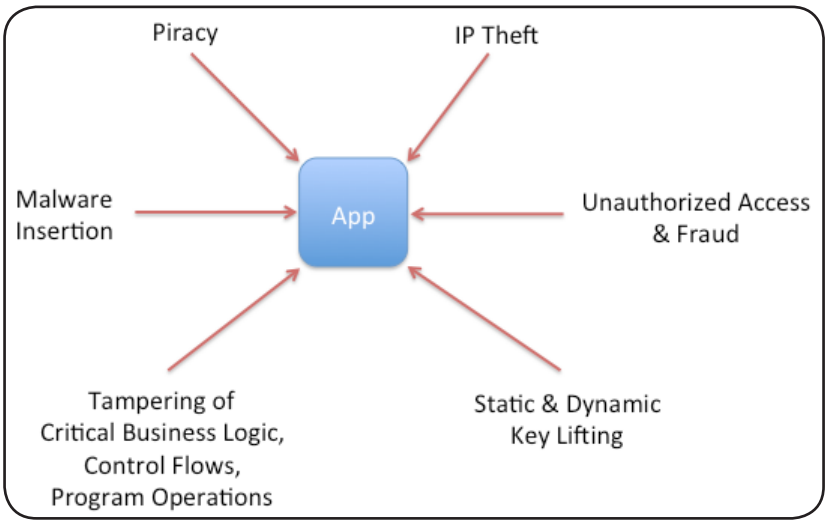

Exhibit 1: Application Threat Vectors

Secure coding and traditional app security practices alone cannot prevent these attacks. An effective application protection solution is paramount to protect binary code from the above threats.

7 Key Factors Of Highly Effective Application Protection Solutions

What should you think about when designing and deploying your approach to protecting your application? There are lots of different approaches you can take to protect your applications. Alternatives should be evaluated based on the following 7 Key Factors:

1. DURABILITY

Ensure your application is being protected against the widest range of static and dynamic attacks.

Leverage layered protection that continually evolves to thwart hackers, so there is no single point of failure.

2. RESILIENCY

Deploy a protection approach that allows your application to "bounce back" and recover gracefully when attacked. This may involve reverting back to the original code, shutting down, and / or having a mechanism to "phone home" to back end system.

3. RUNTIME APPLICATION SELF PROTECTION (RASP)

Leverage protection that is built or linked into an application or application runtime environ ment, and is capable of controlling application execution and detecting and preventing real-time attacks.

4. DEVELOPMENT FRIENDLY

Leverage an approach that's non-disruptive to software development lifecycle (SDLC) – ideally with no impact to source code.

5. IMPACT ON PERFORMANCE

Leverage an approach that offers the ability to tailor the amount of protection per application, resulting in a robust security solution with manageable impact on performance.

6. SCALABILITY

As applications mature and develop, the security solution should have the ability to scale up and increase in complexity without impacting the SDLC.

Ideally your protection solution should support all the platforms that applications run in. This enables you to apply a common approach to all applications – vs. mastering and optimizing many different, platform specific tools.

7. FLEXIBILITY

Leverage an approach that gives precise control over the implementation of the security

Arxan's Application Protection Solutions

Arxan's Application Protection, with it's patented guarding technology, enables you to make your application self-defending, tamper-resistant and self-repairing. This is done by inserting a set of interdependent units of code called Guards® into the application binary. So, there are no source code changes required. These Guards®:

- **Defend** application against compromise via a range of techniques including: advanced obfuscation (code, data and control flow obfuscation), pre-damage, string encryption, symbol stripping, renaming, debug info, call-hiding, resource encryption.
- **Detect** runtime attacks through jailbreak or root detection, resource verification, check-sum, debugger detection, swizzle detection, hook detection, and other means.
- **React** to ward off attacks with self-repair, custom responses, and alerts.

Arxan's Cryptographic Key Protection is a robust implementation of white-box cryptography. It combines mathematical algorithm with data and code obfuscation techniques to transform the keys and related operations so keys cannot be discovered. The keys are never present either in either the static form or in runtime memory.

Cryptographic key protection works in conjunction with our patented guarding technology to provide comprehensive application protection.

Arxan's Unique Approach To Addressing The 7 Key Factors of Application Protection

Arxan's comprehensive application protection solution is the strongest application protection solution and it addresses each of the 7 Key Factors:

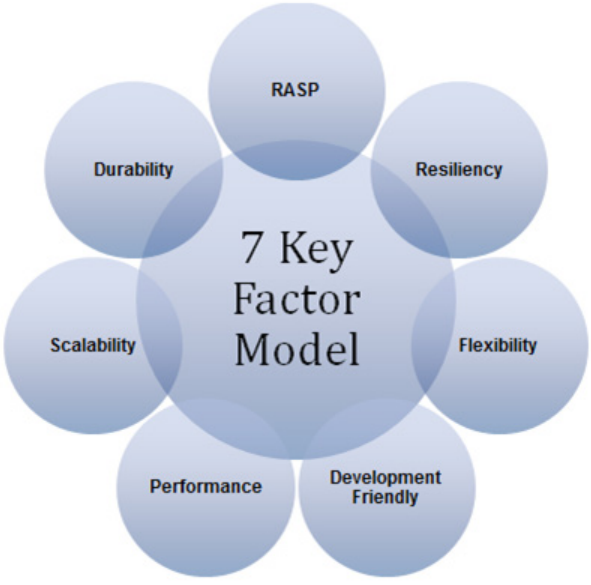

Exhibit 2. 7 Key Factor Model

1. RUNTIME APPLICATION SELF-PROTECTION (RASP)

Runtime application self-protection (RASP) is a security technology that is built or linked into an application or application runtime environment, and is capable of controlling application execution and detecting and preventing real-time attacks.

Without Runtime Application Self-Protection, external malicious apps can intercept the execution of genuine application at runtime and modify it for nefarious activities.

Leading analysts and industry experts are emphasizing the need for Runtime Application Self-Protection (RASP).

> "Make application self-protection a new investment priority, ahead of perimeter and infrastructure protection. It should be a CISO top priority."
> *- Gartner Maverick Research*
>
> "It ('application hardening and runtime protection') is a critical component in the strategy to secure enterprise software, embedded systems, mobile apps and the much-bandied 'Internet of Things.'"
> *- 451 Research*

Arxan's RASP addresses today's sophisticated attacks with security measures that:

- Verify application code and data integrity at runtime
- Accurately identify and block attacks, given visibility into an application's logic and data flow
- Check to ensure that the application is running in a safe environment (e.g., detecting if an app is running on a jailbroken/rooted device or if a debugger is running, that could enable attackers to examine a program while it is running)
- Detect malicious activity from other running apps via Swizzling or Hooking
- Respond to runtime attacks with customizable actions, which may include:
  - Replacing tampered code with the original code during runtime
  - Exiting the application safely when a runtime attack is detected
  - Alerting monitoring systems that an attack has happened

2. DURABILITY

Seasoned hackers are skilled at identifying vulnerabilities and circumventing yes-no decision points, which constitute single points of failure. This enables creation of automated BORE (Break Once Run Everywhere) attack tools.

Arxan provides multiple interconnected layers of defense augmented by complex randomization techniques and strong diversity across binaries to eliminate the possibility of BORE attacks. Techniques include using seed value to create unique instances of protected binaries, and randomization of Guard execution. As a result –

- It becomes extremely difficult and time-consuming task to find and simultaneously disable entire protection.
- It becomes extremely difficult for hackers to create a universal cracking tool.

Arxan's application protection solution protects against the widest range of static, dynamic and

BORE (Break Once Run Everywhere) attacks.

3. RESILIENCY

Arxan's multiple interconnected layers of defense coupled with fully programmable breach management makes the protection solution highly resilient. Protected applications can respond to runtime attacks with customizable actions, which may include:

- Replacing tampered code with the original code during runtime
- Exiting the application safely when a runtime attack is detected
- Alerting monitoring systems that an attack has happened

When protection is defeated, response can be streamlined by renewing the protection scheme quickly and effectively. This can be accomplished without affecting the software development cycle or overall functionality of the software, since the updated protection can be inserted in the binary without modifying the source code.

4. PERFORMANCE

Obfuscating sections of the application that are sensitive to performance degradation, such as computation intensive functions or graphics rendering routines, will have impact on the runtime performance.

Arxan's non-invasive binary-based guard injection engine provides precise control over the implementation of the security. It offers the ability to tailor the amount of protection per application. As a result, you'll have better control on size and performance of the code.

Traditional source code based protection solutions lack tunable performance vs. security tradeoff measures – and often result in performance degradation.

5. SCALABILITY

Arxan's unique binary-based guard injection engine automates the protection insertion into binary or object code without involving source code. Provides easy integration into legacy application, COTS and third-party libraries. Easy and scalable to deploy and update with new releases, and increase the level and complexity of protection techniques, without impacting software development lifecycle.

6. FLEXIBILITY

Arxan's unique binary-based guard injection engine provides precise control over the implementation of the security. It also offers ready-made and custom guards, and automates the insertion of guards. You do not need to design guards from the scratch; the approach is so flexible that it gives you the ability to:

- Choose the type of guard needed
- Specify where to invoke the guard
- Decide what action a guard should take when tampering is detected

Custom guards allow you to create new type of guards for specific applications, based on your unique requirements.

7. DEVELOPMENT FRIENDLY

Arxan's unique binary-based guard injection engine primarily operates at the binary level resulting in a non-invasive approach to application hardening that does not disrupt the software development lifecycle (SDLC). It gives the ability to deploy protection quickly across many applications without risk of compromising the source code. Any changes or modifications to protection scheme require no changes to source code; updated protection can be inserted in the legacy binary without modifying the source code. When protection is defeated, response can be streamlined by renewing the protection scheme quickly and effectively. This can be accomplished without affecting the software development cycle or overall functionality of the software, since the updated protection can be inserted in the binary without modifying the source code.

To watch a short video that describes how Arxan protects applications, click here.

For more information on Arxan and our solutions, contact us at info@arxan.com.

Sources:
1. Fake Apps - Feigning Legitimacy 2014, Trend Micro
1. State of Mobile App Security 2014, Arxan
2. Key findings from The Global State of Information Security® Survey 2015, PwC
3. Internet of Things State of the Union Study, July 2014, HP Fortify
4. 2014 Data Breach Investigations Report, Verizon