MAPS – Mobile Application Protection Suite

Suite para proteção de aplicativos – desde o desenvolvimento, segurança das chaves, ofuscação do Código, contra adulteração, engenharia reversa, criptografia até o uso no cliente final

Dezembro de 2021

Sobre as Empresas

A Zimperium é uma empresa global baseada nos Estados Unidos com sedes em Dallas, Texas e São Francisco. É líder mundial em segurança móvel e possui milhares de clientes e funcionários em todo o mundo. Entre seus clientes, empresas de todos os tamanhos e segmentos, inclusive bancos, grandes agências governamentais e operadoras de telefonia, entre outros. A Zimperium levantou milhões de dólares em investimentos privados de empresas como Samsung, Telstra e SoftBank.

A Gold Lock foi criada em 2003 em Israel e chegou ao Brasil em 2005. Pioneira na tecnologia de criptografia para telefones celulares e Licenciada pelo Ministério de Defesa de Israel, é líder mundial nesse segmento e referência na proteção de dispositivos móveis. Presente em mais de 20 países e representando no Brasil diversas empresas internacionais – incluindo companhias de Estados Unidos, Israel e Suíça – tem entre seus clientes empresas privadas, públicas, órgãos governamentais e militares.

A Gold Lock é distribuidor e representante oficial da Zimperium no Brasil.

Sumário Executivo

Dispositivos móveis agora constituem a maioria do tráfego da web globalmente, e os consumidores estão adotando aplicativos móveis em sua vida diária. As organizações que desenvolvem aplicativos móveis estão profundamente cientes da necessidade de proteger esses apps. Os aplicativos móveis estão cada vez mais sofisticados e processam informações pessoais e corporativas mais confidenciais do que nunca. As organizações também estão cientes dos enormes riscos financeiros e de reputação, diretos por meio de fraude ou indiretos por meio de multas etc., que podem resultar de violações móveis.

As equipes de desenvolvimento e segurança precisam de visibilidade em todo o ciclo de vida de desenvolvimento de software (SDLC), mas seus esforços foram frustrados por um conjunto altamente fragmentado de soluções e nenhuma visibilidade das ameaças nos dispositivos dos usuários finais. Os aplicativos móveis podem gerar um valor comercial significativo, mas apenas se as barreiras à segurança do aplicativo puderem ser superadas, por exemplo.

As soluções tradicionais de segurança de aplicativos se concentram em uma abordagem de dentro para fora para proteger e fornecer verificação de tempo de execução de dispositivos comprometedores. Essas abordagens geralmente são baseadas em assinaturas e não acompanham o cenário de ameaças em rápida evolução nos dispositivos móveis. Zimperium oferece uma abordagem externa para enfrentar esses desafios com técnicas comportamentais / de aprendizado de máquina como parte do DevSecOps e segurança integrada, monitorando e fechando o ciclo de feedback.

A Zimperium oferece o único Mobile Application Protection Suite (MAPS) completo. Zimperium MAPS identifica riscos de segurança / privacidade / conformidade durante o desenvolvimento de aplicativos e protege / monitora aplicativos de ataques durante o uso. O MAPS é a única solução de segurança de aplicativo móvel que fornece identificação de risco e proteção em todo o SDLC.
O MAPS consiste em quatro componentes, cada um dos quais abordando necessidades empresariais específicas e estágios SDLC. Com o MAPS, o todo realmente é maior do que a soma das partes. Todas as soluções utilizam o mesmo back-end e console administrativo, zConsole, para fornecer visibilidade abrangente e contínua e gerenciamento de riscos e ameaças. Por ter um conjunto integrado em todo o SDLC, MAPS fornece informações valiosas que não apenas detectam os riscos atuais, mas também ajudam os desenvolvedores a identificar problemas que podem ser resolvidos em versões futuras.

● zScan: Ajuda sua organização de desenvolvimento de aplicativos móveis a descobrir e corrigir problemas de conformidade, privacidade e segurança dentro do processo de desenvolvimento antes de você lançar publicamente seus aplicativos (Análise Automatizada para Aplicativos em Desenvolvimento)

● zKeyBox: A melhor solução de criptografia de WhiteBox (WBC) que protege suas chaves para que não possam ser descobertas, extraídas ou manipuladas

● zShield: protege o aplicativo, a propriedade intelectual e os dados de ataques potenciais, como engenharia reversa e violação de código, além de adulteração e ofuscação do código

● zDefend: Fornece visibilidade de ameaças e proteção em tempo de execução baseada em ML no dispositivo (RASP) contra ataques no dispositivo, rede, phishing e malware

Prêmios e Reconhecimentos da Zimperium

Cybersecurity Excellence Awards 2020: Gold Winner in Mobile Threat Defense – zPlatform

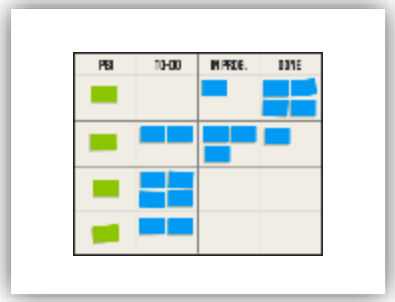
Cybersecurity Excellence Awards 2020: Silver Winner in Best Cybersecurity Company

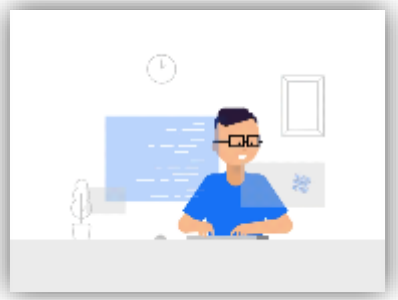
Most Innovative Enterprise Mobile Threat Defense - 2020
Next Gen Application Security – 2020
Best Product Mobile Endpoint Security - 2020
Best Product Artificial Intelligence and Machine Learning - 2020

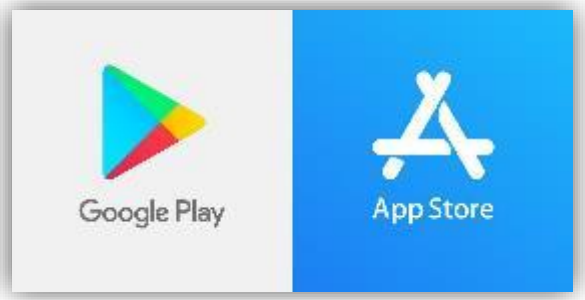
Best Application Security in 2019

Cybersecurity Excellence Awards 2019: Best Mobile Application Security - zIAP

Ciclo de vida simplificado do desenvolvimento de software (SDLC)







Desenvolvimento

Execução

Desenvolvimento em conformidade

Quais problemas devo corrigir antes de liberar para a produção?

zSCAN™

Desenvolvimento seguro

Proteção do meu aplicativo contra possíveis ataques e engenharia reversa

zSHIELD™

Execução segura

O que está acontecendo com meu aplicativo na utilização que eu deveria estar ciente?

zDEFEND™

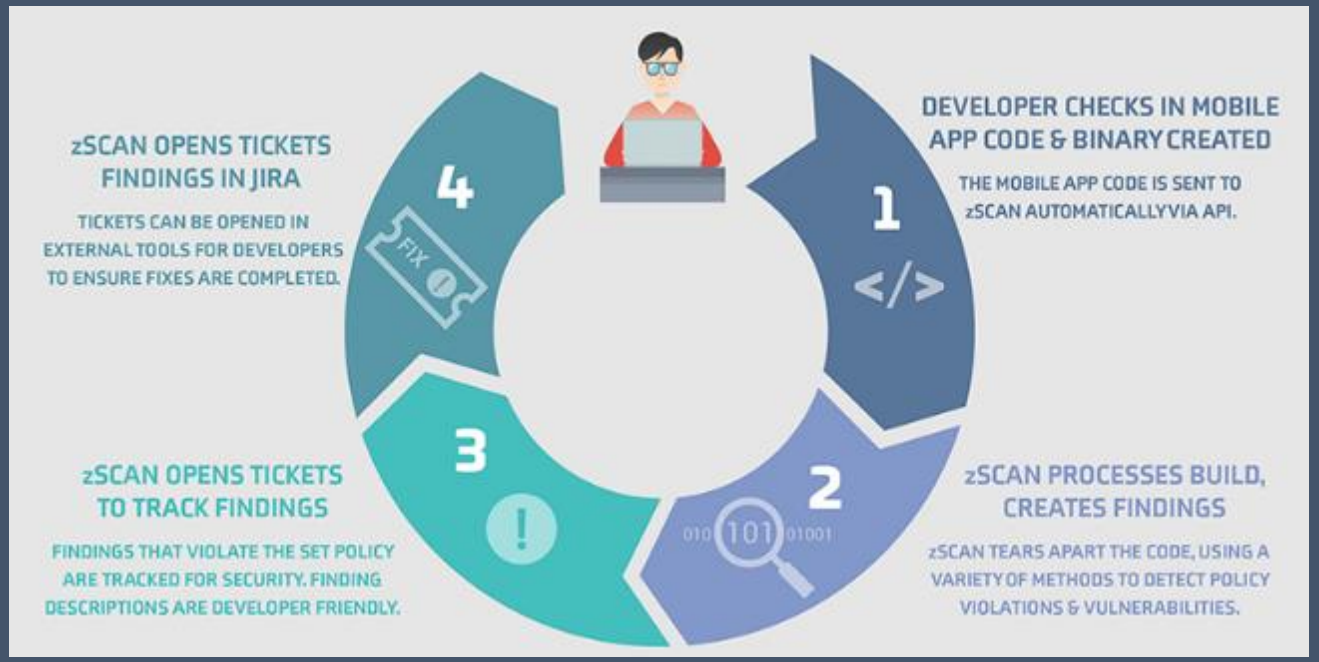
zSCAN™

O zSCAN ajuda os desenvolvedores de aplicativos móveis a evitar riscos de reputação e financeiros, identificando automaticamente os riscos de privacidade, segurança e conformidade no processo de desenvolvimento antes que os aplicativos sejam lançados ao público. Embora as ferramentas tradicionais de análise de código ajudem a avaliar a qualidade geral do código de um desenvolvedor, os recursos de análise binária do zSCAN identificam os riscos que um invasor pode descobrir para explorar o aplicativo concluído.

O zSCAN documenta riscos em aplicativos móveis, incluindo uso específico de hardware, chamadas inseguras de API e manipulação de dados confidenciais. Os aplicativos podem ser adicionados diretamente do pipeline de compilação ou carregados manualmente, conforme desejado. No console administrativo do zSCAN, o zConsole, conformidade e segurança podem definir e personalizar políticas para garantir que apenas as descobertas relevantes sejam trabalhadas.

O zScan se integra diretamente ao seu processo de desenvolvimento sem exigir que seus desenvolvedores alterem processos, implementem qualquer novo código ou tenham que fazer login em um console de sistema separado. Depois que as descobertas são descobertas, o zScan abre tíquetes em sistemas de tíquetes (como JIRA, Cloudbees Jenkins e TeamCity) para fornecer aos desenvolvedores informações detalhadas e pacotes de trabalho necessários para lidar com o risco. Depois de corrigidas, as informações são sincronizadas de volta com o zScan para que as equipes de segurança e conformidade possam verificá-las.

Além disso, o recurso "Build Compare" do zScan mostra rapidamente se os riscos estão aumentando ou diminuindo em cada versão subsequente. As comparações de versões permitem que as organizações meçam o progresso da conformidade e forneçam aplicativos móveis mais resilientes.



ZIMPERIUM

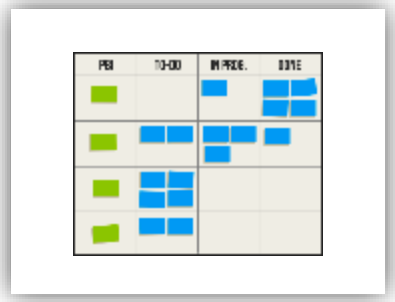
O zScan da Zimperium ajuda as organizações a superar desafios e produzir aplicativos móveis de forma consistente com menos riscos de privacidade, segurança e conformidade:

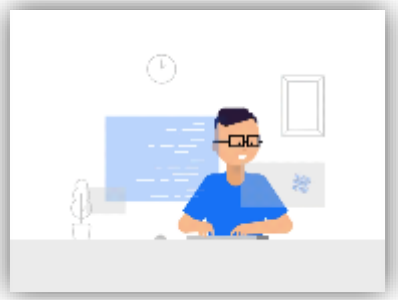
Dando a você visibilidade imediata dos riscos do aplicativo que você não veria em outros scanners em termos de privacidade e segurança;

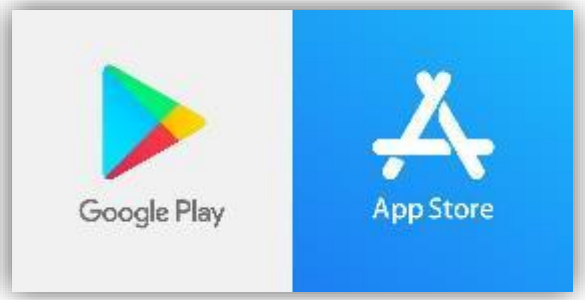
Identificação de problemas de conformidade para NIAP, GDPR e OWASP Mobile Top 10; e

Reduzindo os tempos de ciclo, analisando dentro do pipeline de construção, inspecionando os dados e documentando detalhes em sua ferramenta scrum existente.

Processo de entrega de aplicativo | Feedback de Risco?







Desenvolvimento **Execução**

Equipe de segurança

Teste automatizado de aplicativos

ZIMPERIUM

zKeyBox - Segurança avançada para suas chaves e segredos

Os aplicativos móveis usam chaves para criptografar as comunicações enviadas e recebidas pois contêm dados confidenciais. Mesmo os métodos de criptografia mais fortes falham quando as chaves criptográficas estão comprometidas. Hackers podem facilmente encontrar e roubar chaves expostas em código ou memória. Zimperium zKeyBox aproveita a criptografia de White box para proteger as chaves e segredos em seu aplicativo móvel. Ele transforma e obscurece algoritmos criptográficos para que as chaves nunca apareçam claras e a lógica de execução seja indetectável. Suas chaves não podem ser extraídas - mesmo que o próprio dispositivo tenha sido violado.

Benefícios:

Proteção de chave baseada em software mais forte. Oculta e obscurece as chaves e a lógica do algoritmo para que as chaves não possam ser extraídas e as tentativas de violação sejam encerradas. Sem dependência de quaisquer mecanismos baseados em hardware fornecidos pelas plataformas (Ex. Keystores, Secure Enclave, Trusted Execution Environment (TEE) no Android).

Proteja as chaves quando armazenadas, em trânsito e em uso
Mantenha as chaves sempre seguras, mesmo em dispositivos comprometidos, desbloqueados ou com acesso root. As chaves nunca são expostas na memória; algoritmos operam diretamente em chaves codificadas.

Acelere o tempo de chegada ao mercado
Substitua suas bibliotecas criptográficas padrão com proteção de chave segura de caixa branca plug and play.

Qualquer algoritmo, qualquer plataforma
A segurança agnóstica funciona em todas as plataformas e dispositivos. Proteja qualquer algoritmo criptográfico, como AES, 3DES, RSA, ECC, HMAC e outros. O suporte a algoritmos personalizados também estão disponíveis.

Cumprir os regulamentos
Atenda e exceda os requisitos de segurança de aplicativos e privacidade de dados, minimizando os prazos de aprovação e teste. Suporta especificações PCI-DSS, incluindo separação de cartão de pagamento e dados de PIN.

Apoiado por especialistas
A profunda experiência da Zimperium para guiar cada etapa de sua implantação. zKeyBox protege as chaves em milhões de aplicativos instalados e passa por testes de segurança independentes regulares.

Implementação fácil que acelera o tempo de colocação no mercado
Integração perfeita: zKeyBox é um substituto plug and play simples de integrar para bibliotecas criptográficas padrão.
Suporte integrado para regulamentações de segurança: passa por testes regulares de penetração e oferece suporte ao gerenciamento de chaves DUKPT, blocos de chaves TR-31 e separação de cartão de pagamento e dados de PIN conforme especificado pelo PCI-DSS.
Nenhum hardware de segurança dedicado: não são necessários dispositivos TPM, TEE, SE, SIM ou HSM.

O zShield fortalece e protege o aplicativo com funcionalidade avançada de ofuscação e anti-adulteração para limitar ataques como engenharia reversa, pirataria, remoção de anúncios, extração de ativos, extração de chaves de API e reembalagem com malware.

O zShield fortalece e protege seus aplicativos de três maneiras principais:

1) Ofuscação para impedir engenharia reversa
2) Visibilidade de adulteração de aplicativos na operação
3) Desenvolvimento contínuo e integrações de segurança

ZIMPERIUM

Para evitar tentativas de engenharia reversa utilizando análise estática, o zShield implanta várias técnicas de proteção de código para ofuscar a visibilidade do código. Duas técnicas de muitas são ofuscação de nome e ofuscação de fluxo de controle:

Ofuscação de nomes Android - zShield ofusca os nomes de classes, campos, métodos, bibliotecas nativas, recursos, ativos e atributos XML de recursos.

Ofuscação de nomes iOS - zShield ofusca identificadores em código Swift e Objective-C para ocultar informações semânticas de engenheiros reversos. As construções de reflexão mais comuns são suportadas fora da caixa.

Ofuscação do fluxo de controle para Android e iOS - o zShield ofusca o fluxo de controle do código dentro dos métodos e da lógica da função original para impedir a análise de código automatizada e manual.

Ao contrário de outras soluções que dependem de pen teste manual para demonstrar eficácia e não têm relatórios ativos, o zShield fornece relatórios imediatos e contínuos sobre tentativas de hacking.

O zShield relata eventos de violação de aplicativo no painel de administração e relatório do Zimperium, zConsole, e oferece análise forense abrangente. O zShield protege seus aplicativos contra análises dinâmicas e ataques ao vivo usando vários mecanismos de autoproteção em tempo de execução, como fixação de SSL, detecção de gancho e verificações de certificado.

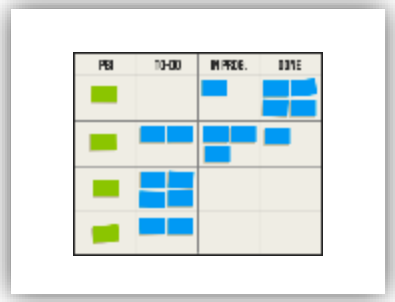
O zShield se integra de forma transparente ao seu processo de construção e não requer alterações no código-fonte. Ele fornece plug-ins para todas as ferramentas comuns de construção e ambientes de desenvolvimento, como Gradle, Android Studio, Ant, Eclipse, Maven e construções personalizadas.

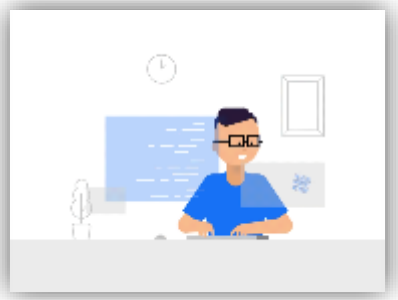
Depois que seu aplicativo for otimizado e ofuscado com zShield, ele relatará tentativas de hacking e adulteração diretamente em seu console Zimperium e pode ser facilmente integrado com seu sistema de gerenciamento de informações e eventos de segurança (SIEM) para análise e ação adicionais.

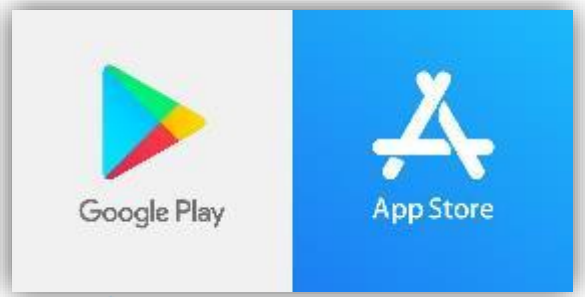
O zShield oferece proteção de código estático implementando técnicas de ofuscação no binário do aplicativo antes de publicá-lo no armazenamento público. Aqui estão os principais recursos cobertos pela solução zSheild. A tabela abaixo resume os principais recursos de proteção do código-fonte do Zimperium:

Android	iOS
❖ Name obfuscation ❖ String/Class/Library encryption ❖ Reflection ❖ Code obfuscation ❖ Code virtualization ❖ Removing logging code and stack traces ❖ Tamper detection -CertificateChecker -FileChecker ❖ Asset encryption ❖ Resource file encryption ❖ Resource string encryption ❖ Metadata encryption ❖ Packer	❖ Name obfuscation ❖ Arithmetic Obfuscation ❖ Control Flow Obfuscation ❖ Data Obfuscation ❖ Code Integrity ❖ Application Integrity

Processo de entrega de aplicativo | Proteção estática?



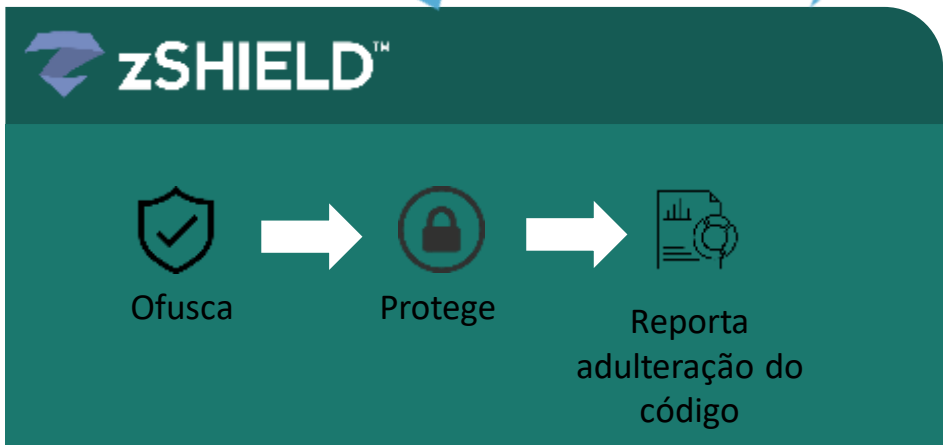




Desenvolvimento

Execução

Equipe de segurança



Equipe de prevenção de fraudes

Antes -> Depois

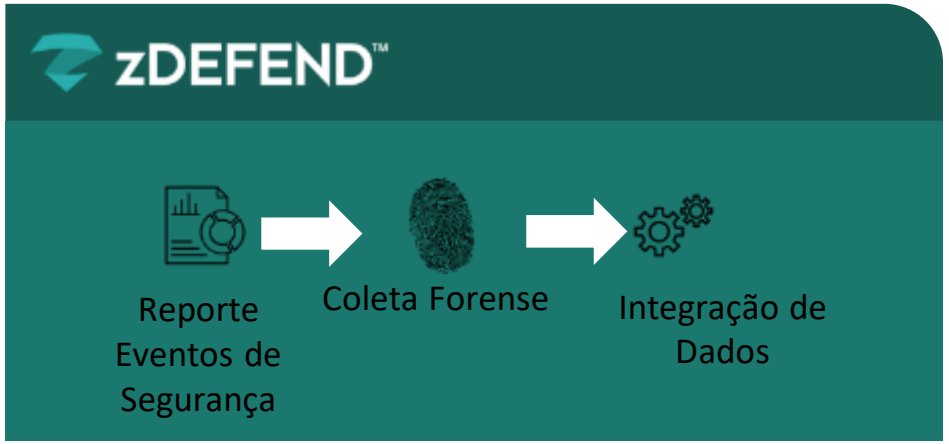
com.google.android.gms.common.Feature[] getAvailableFeatures() -> I
com.google.android.gms.common.api.Api$ClientKey -> o.cı$iF:
1:1:void <init>() -> <init>
com.google.android.gms.common.api.Api$SimpleClient -> o.cı$IF:
java.lang.String getStartServiceAction() -> I
java.lang.String getServiceDescriptor() -> ı
android.os.IInterface createServiceInterface(android.os.IBinder) -> ı
void setState(int,android.os.IInterface) -> ı
com.google.android.gms.common.api.Api$zaa -> o.cı$I:
com.google.android.gms.common.api.Api$zab -> o.cı$ı:
com.google.android.gms.common.api.ApiException -> o.lı:
com.google.android.gms.common.api.Status mStatus -> ı
1:6:void <init>(com.google.android.gms.common.api.Status) -> <init>

Antes -> Depois

com.google.android.gms.common.Feature[] getAvailableFeatures() -> I
com.google.android.gms.common.api.Api$ClientKey -> o.cı$iF:
1:1:void <init>() -> <init>
com.google.android.gms.common.api.Api$SimpleClient -> o.cı$IF:
java.lang.String getStartServiceAction() -> I
java.lang.String getServiceDescriptor() -> ı
android.os.IInterface createServiceInterface(android.os.IBinder) -> ı
void setState(int,android.os.IInterface) -> ı
com.google.android.gms.common.api.Api$zaa -> o.cı$I:
com.google.android.gms.common.api.Api$zab -> o.cı$ı:
com.google.android.gms.common.api.ApiException -> o.lı:
com.google.android.gms.common.api.Status mStatus -> ı
1:6:void <init>(com.google.android.gms.common.api.Status) -> <init>

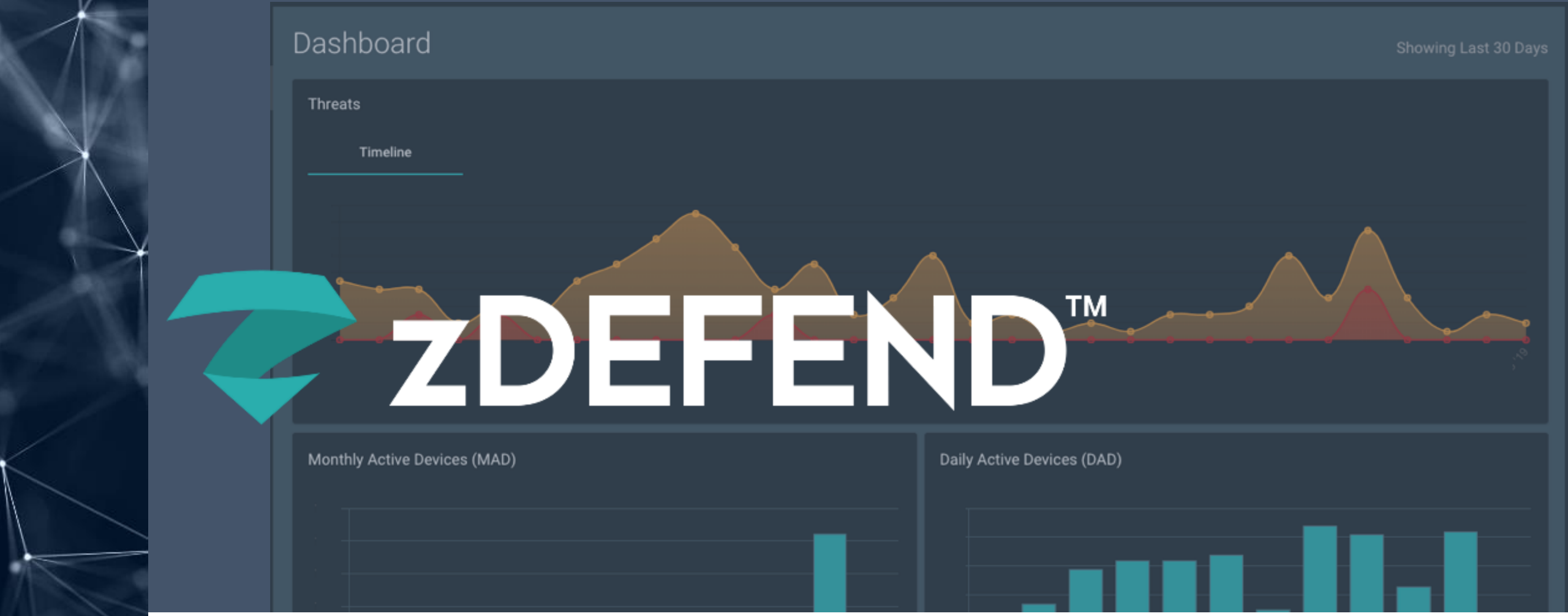
Desenvolvimento

Execução



Equipe de segurança

Equipe de prevenção de fraudes



O zDefend incorpora rapidamente o mecanismo avançado z9 em aplicativos móveis desenvolvidos para uso interno ou externo do cliente. O z9 oferece os seguintes benefícios:

- Segurança móvel no dispositivo baseada em aprendizado de máquina para ataques em dispositivos, redes, phishing e aplicativos maliciosos
- Incorpora rápida e invisivelmente um SDK dentro dos aplicativos, resultando em 100% de adoção de segurança
- Possui sobrecarga insignificante e requisitos mínimos de permissão

zDefend atua on-device e em tempo real, utilizando aprendizado de máquina para determiner se o dispositivo está comprometido, se a rede Wi-fi está com problema ou se há apps maliciosos instalados. Também protege contra malware, zero-day, MITM, Phishing, exploit, spyware, riskware, trojan, BankBot, Adware, Hacktool e muito mais com o mecanismo de detecção patenteado z9.

O SDK zDefend permite que os desenvolvedores incorporem de forma rápida o mecanismo de detecção baseado em aprendizado de máquina líder da Zimperium, z9, diretamente em qualquer aplicativo móvel. Com o SDK zDefend incorporado, os aplicativos móveis podem determinar imediatamente se o dispositivo de um usuário está comprometido, se algum ataque de rede está ocorrendo e até mesmo se aplicativos maliciosos estão instalados. zDefend é totalmente configurável por desenvolvedores de aplicativos, que podem selecionar qualquer ação corretiva a ser aplicada quando uma determinada ameaça for detectada. Quando um dispositivo está sob ataque, o zDefend informa o aplicativo e inicia as ações de mitigação de risco predeterminadas. Os eventos de ameaça em tempo real, à medida que são detectados e informados ao aplicativo, serão relatados ao console de gerenciamento.

O mecanismo de detecção baseado em aprendizado de máquina da Zimperium, z9, detecta ataques de dispositivos, redes, phishing e aplicativos móveis no dispositivo. O Zimperium z9 foi desenvolvido especificamente para dispositivos móveis, não portado da segurança de endpoint tradicional, para proteger contra ameaças exclusivas, conhecidas e desconhecidas destinadas a dispositivos iOS, Android e Chromebook. A detecção do Zimperium z9 é executada no dispositivo de maneira eficiente, sem introduzir latência ou violar a privacidade do usuário. O zDefend da Zimperium incorpora o mecanismo avançado do Zimperium z9 em aplicativos móveis desenvolvidos para uso interno ou externo. Zimperium z9 oferece os seguintes benefícios: Não depende de assinatura e a lógica de detecção de ameaças pode ser atualizada dinamicamente sem a necessidade de atualização do aplicativo. Possui sobrecarga desprezível e requisitos mínimos de permissão. Nos últimos cinco anos, o Zimperium z9 detectou 100% dos exploits móveis de dia zero à solta no dispositivo e sem a necessidade de atualizações. Exemplos de detecções de dia zero são exploits Stagefright, Pegasus, BlueBorne, SockPuppet, Checkm8 e MediaTek-su; nenhum deles exigia que o Zimperium z9 fosse atualizado antes que pudesse detectar esses exploits.

Para que os invasores violem arquivos e dados críticos em sua própria sandbox, o dispositivo deve estar comprometido de acordo com o design de segurança do Android / iOS. O SDK da Zimperium detecta comprometimentos no nível do sistema operacional e, por exemplo, alertará o aplicativo para executar uma ação de resposta:

- Dispositivo Rooted/Jailbroken
- Adulteração de sistema
- Sistema de arquivos alterado
- SE Linux desativado
- Elevação de privilégios
- Atestado Google SafetyNet

O SDK do Zimperium fornece detecção de tempo de execução quando o aplicativo é debugged. Esta é uma verificação de combinação da configuração do dispositivo, incluindo o seguinte:

- Modo de depuração USB ativado
- Modo de desenvolvedor habilitado
- Aplicativos Android Debug Bridge não verificados
- Ataque de dispositivo / violação de sistema operacional

Benefícios

• Proteção abrangente no dispositivo: segurança móvel baseada no aprendizado de máquina no dispositivo para ataques a dispositivos, redes, phishing e aplicativos maliciosos

• Visibilidade da ameaça: Fornece telemetria da ameaça do dispositivo em tempo real na base de instalação para Equipes de segurança, risco e conformidade

• Resposta dinâmica à ameaça: a resposta do dispositivo pode ser atualizada dinamicamente sem precisar de uma nova versão do aplicativo

• Integrações de ecossistemas de segurança: o painel se integra com SIEM / SOAR e outros sistemas de resposta a incidentes

• Fácil de implementar: incorpora de forma rápida e invisível como um SDK dentro de aplicativos resultantes em 100% de adoção de segurança

• Sobrecarga insignificante: Tem sobrecarga insignificante e permissão mínima de requisitos

• Modelos de implantação flexíveis: a solução pode ser implantada como SaaS e OnPremises

• Suporte de aplicativo nativo e híbrido: oferece suporte a aplicativos desenvolvidos usando o nativo e Estruturas híbridas

Com o zDefend SDK incorporado, os aplicativos móveis podem determinar imediatamente quando o dispositivo de um usuário está comprometido, quando quaisquer ataques de rede estão ocorrendo e até mesmo se aplicativos maliciosos estão instalados. Os desenvolvedores de aplicativos podem configurar ações corretivas apropriadas quando uma determinada ameaça é detectada. Os cenários de configuração incluem:

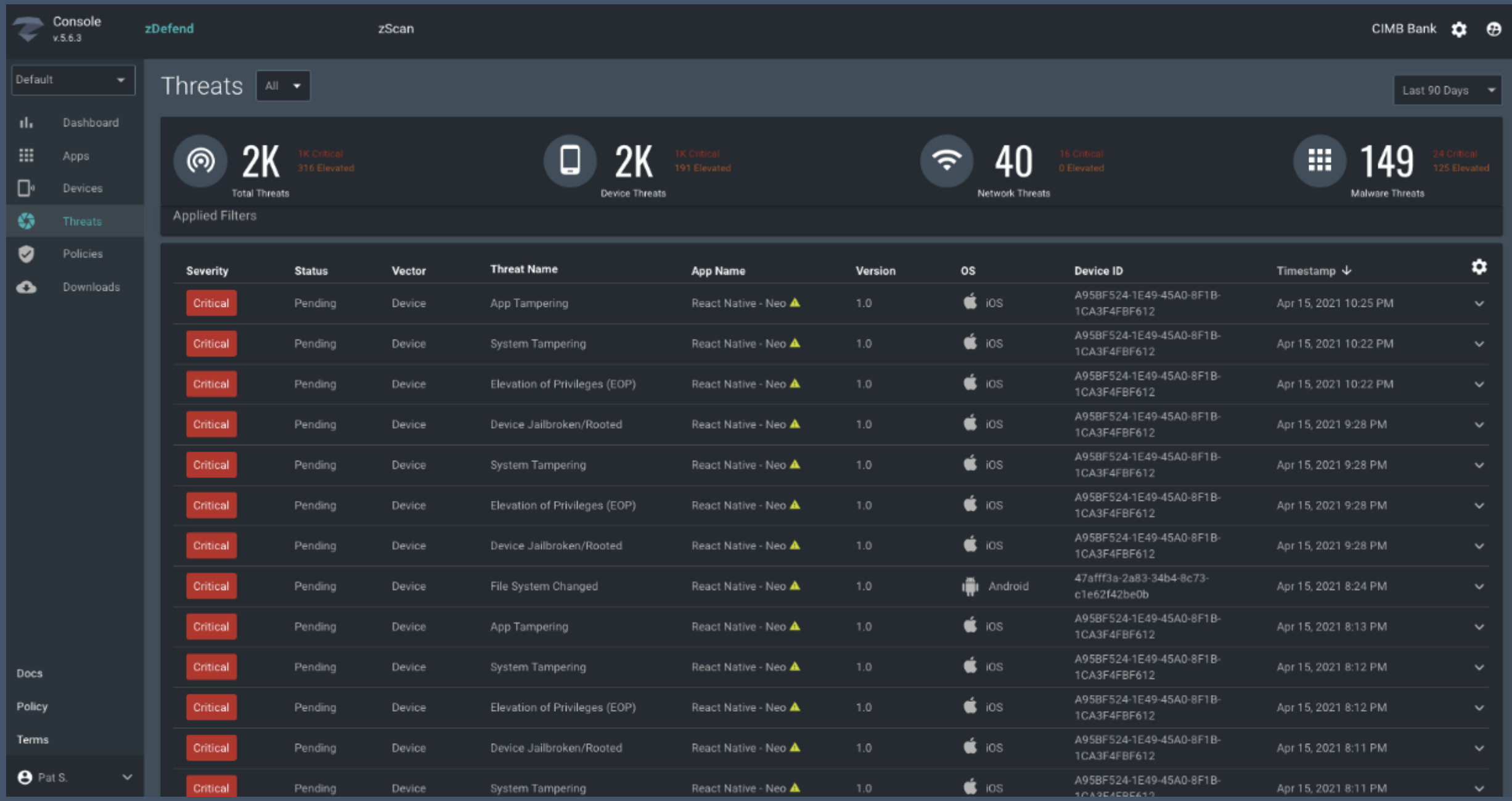
Quando um ataque man-in-the-middle (MITM) está ocorrendo, o aplicativo host estabelece automaticamente uma VPN para criar um túnel seguro;

Quando um dispositivo tem malware como o BankBot instalado, o aplicativo aciona etapas imediatas para congelar o acesso até que o usuário exclua o aplicativo infectado e redefina sua senha online; ou

Quando um dispositivo é desbloqueado, o aplicativo pode acionar políticas para aumentar a pontuação de fraude do usuário para compensar o risco adicional ou acionar outras políticas condicionais.

Example of actions your app can take with zDEFEND embedded

Threat / Attack Detected	Potential Action
A man-in-the-middle (MITM) attack occurs	The app can automatically establish a VPN to create a secure tunnel
A device has malware like BankBot installed	The app can trigger immediate steps to freeze access until the user deletes the BankBot-carrying app and resets their password online
A device has been Jailbroken by the user	The app can end the session or flag certain transactions to additional verification
A device has been compromised by an external actor	The app can display a dialog box asking the user to complete their transaction from a different device
A money transfer attempt was made on an unsafe WiFi network	Ask user to connect to safer network, reducing transfer limits or request additional verification
A 2FA token is sent to a compromised device	Request additional authorization on a different device



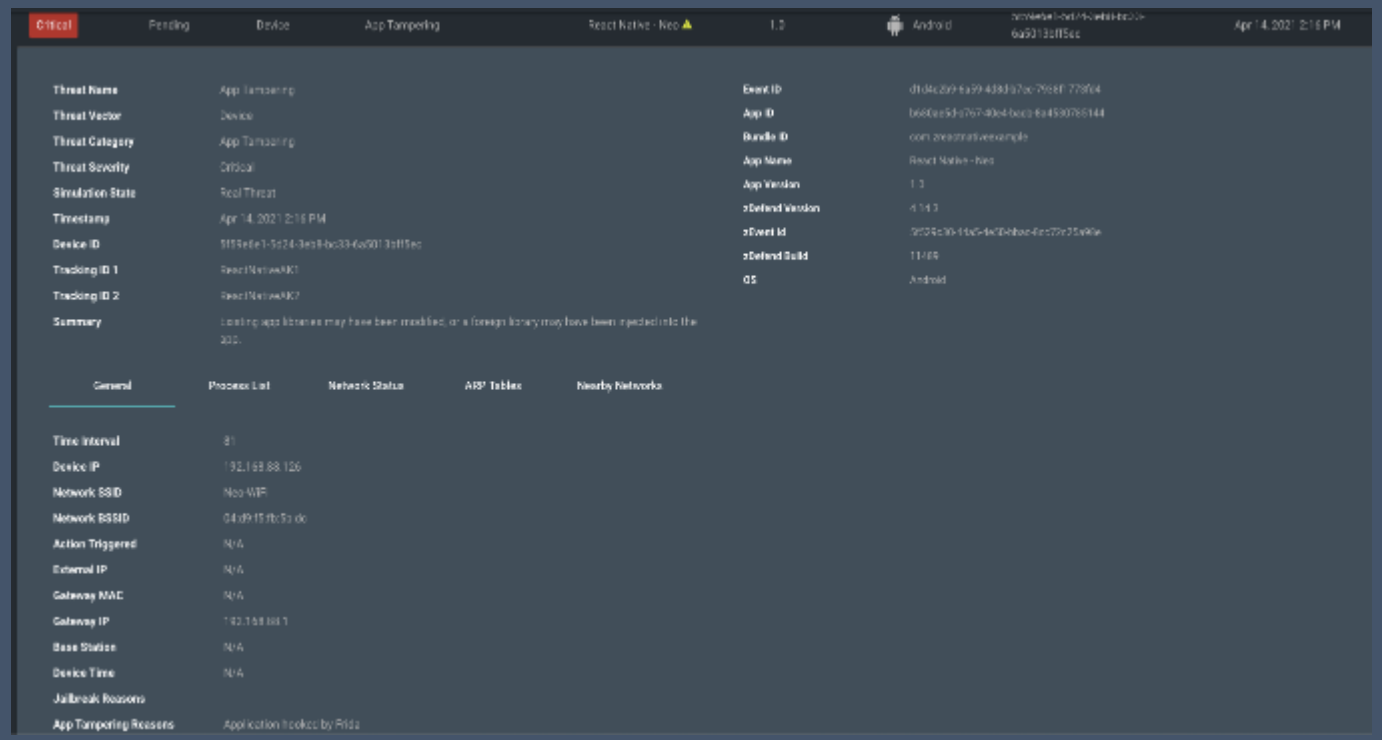
Threats

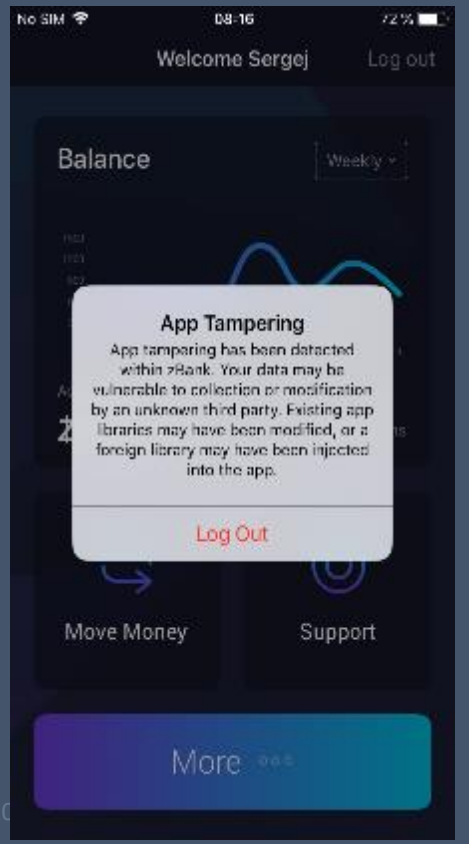
All

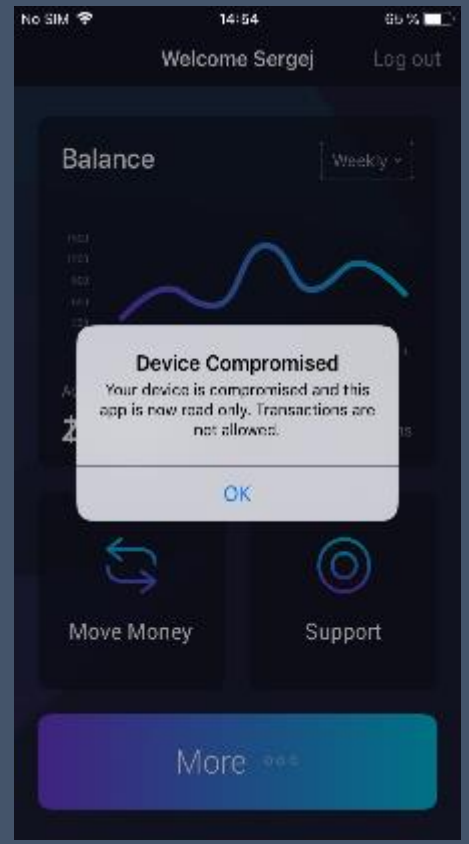
Last 90 Days

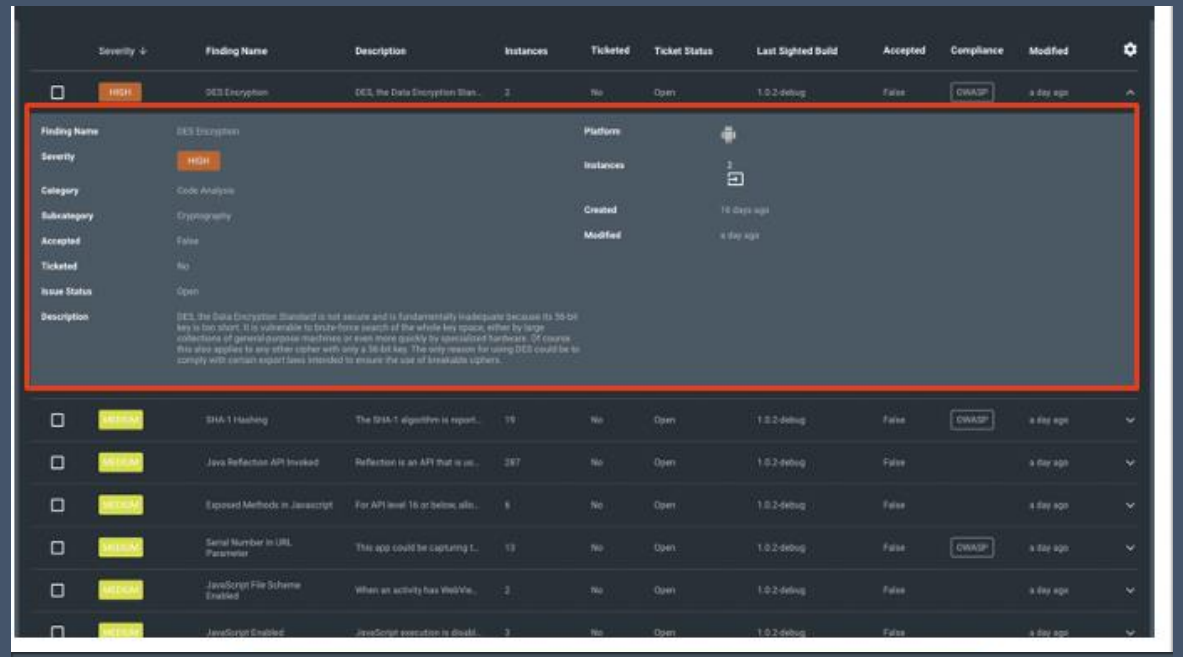
1K	1K Critical, 253 Elevated	Total Threats
1K	1K Critical, 159 Elevated	Device Threats
34	11 Critical, 0 Elevated	Network Threats
116	22 Critical, 94 Elevated	Malware Threats

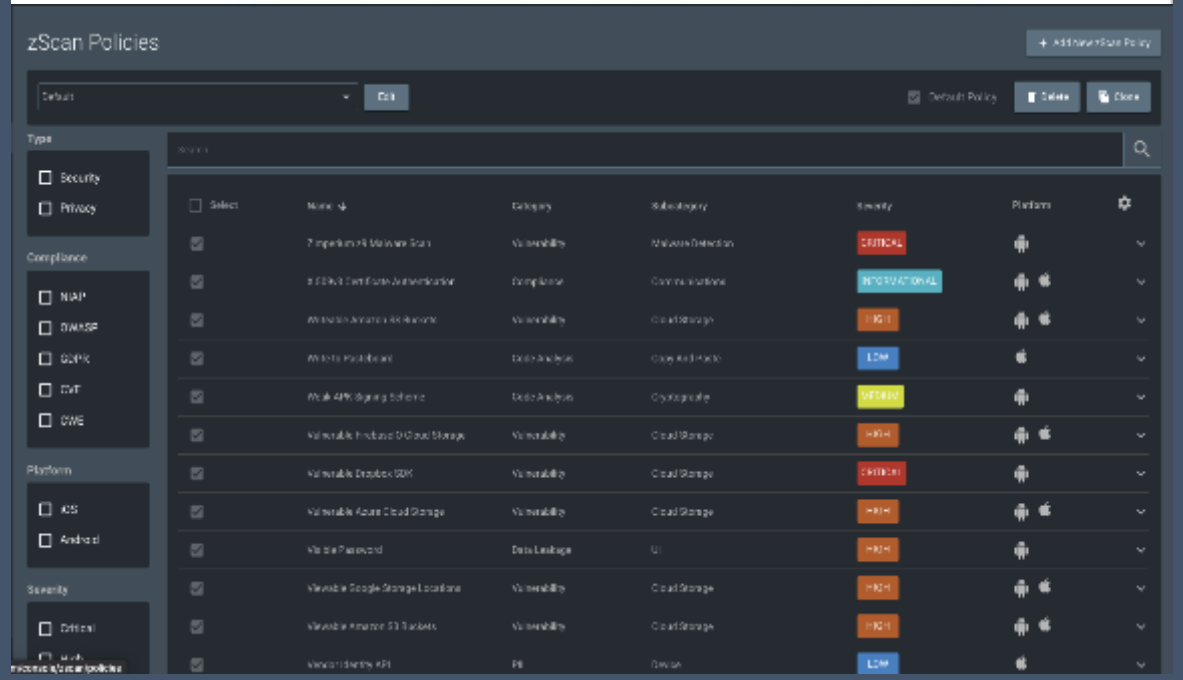
Severity	Status	Vector	Threat Name	App Name	Version	OS	Device ID	Timestamp ↓
Critical	Pending	Malware	Suspicious Android App	React Native Test	1.0	Android	0b65c97d-8018-3739-8c64-660a07e2307e	Apr 14, 2021 2:41 PM
Critical	Pending	Device	App Tampering	React Native - Neo	1.0	Android	5f59e6e1-5d24-3eb8-bc33-6a5013bff5ec	Apr 14, 2021 2:16 PM
Elevated	Pending	Device	Elevation of Privileges (EOP)	React Native - Neo	1.0	Android	5f59e6e1-5d24-3eb8-bc33-6a5013bff5ec	Apr 14, 2021 2:15 PM
Elevated	Pending	Device	Elevation of Privileges (EOP)	React Native - Neo	1.0	Android	5f59e6e1-5d24-3eb8-bc33-6a5013bff5ec	Apr 14, 2021 2:15 PM
Elevated	Pending	Device	Elevation of Privileges (EOP)	React Native - Neo	1.0	Android	5f59e6e1-5d24-3eb8-bc33-6a5013bff5ec	Apr 14, 2021 2:15 PM
Critical	Pending	Device	App Tampering	React Native - Neo	1.0	Android	5f59e6e1-5d24-3eb8-bc33-6a5013bff5ec	Apr 14, 2021 2:13 PM
Elevated	Pending	Device	Elevation of Privileges (EOP)	React Native - Neo	1.0	Android	5f59e6e1-5d24-3eb8-bc33-6a5013bff5ec	Apr 14, 2021 2:13 PM
Elevated	Pending	Device	Elevation of Privileges (EOP)	React Native - Neo	1.0	Android	5f59e6e1-5d24-3eb8-bc33-6a5013bff5ec	Apr 14, 2021 2:13 PM
Elevated	Pending	Device	Elevation of Privileges (EOP)	React Native - Neo	1.0	Android	5f59e6e1-5d24-3eb8-bc33-6a5013bff5ec	Apr 14, 2021 2:12 PM
Elevated	Pending	Device	Elevation of Privileges (EOP)	React Native - Neo	1.0	Android	5f59e6e1-5d24-3eb8-bc33-6a5013bff5ec	Apr 14, 2021 2:12 PM
Elevated	Pending	Device	Elevation of Privileges (EOP)	React Native - Neo	1.0	Android	5f59e6e1-5d24-3eb8-bc33-6a5013bff5ec	Apr 14, 2021 2:12 PM
Elevated	Pending	Device	Elevation of Privileges (EOP)	React Native - Neo	1.0	Android	5f59e6e1-5d24-3eb8-bc33-6a5013bff5ec	Apr 14, 2021 2:11 PM
Elevated	Pending	Device	Elevation of Privileges (EOP)	React Native - Neo	1.0	Android	5f59e6e1-5d24-3eb8-bc33-6a5013bff5ec	Apr 14, 2021 2:11 PM
Elevated	Pending	Device	Elevation of Privileges (EOP)	React Native - Neo	1.0	Android	5f59e6e1-5d24-3eb8-bc33-6a5013bff5ec	Apr 14, 2021 2:11 PM
Critical	Pending	Device	SELinux Disabled	React Native - Neo	1.0	Android	5f59e6e1-5d24-3eb8-bc33-6a5013bff5ec	Apr 14, 2021 2:11 PM

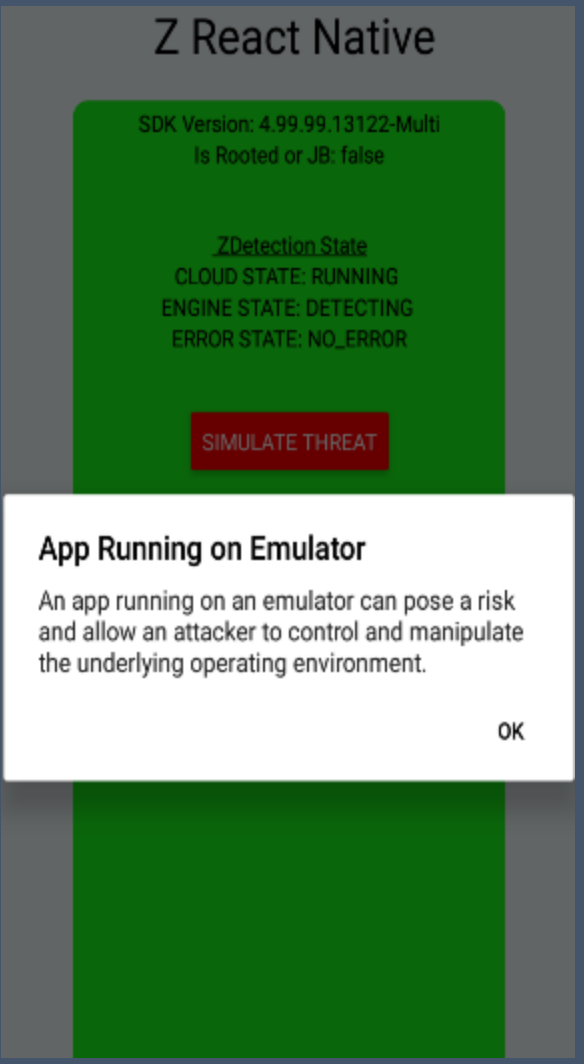










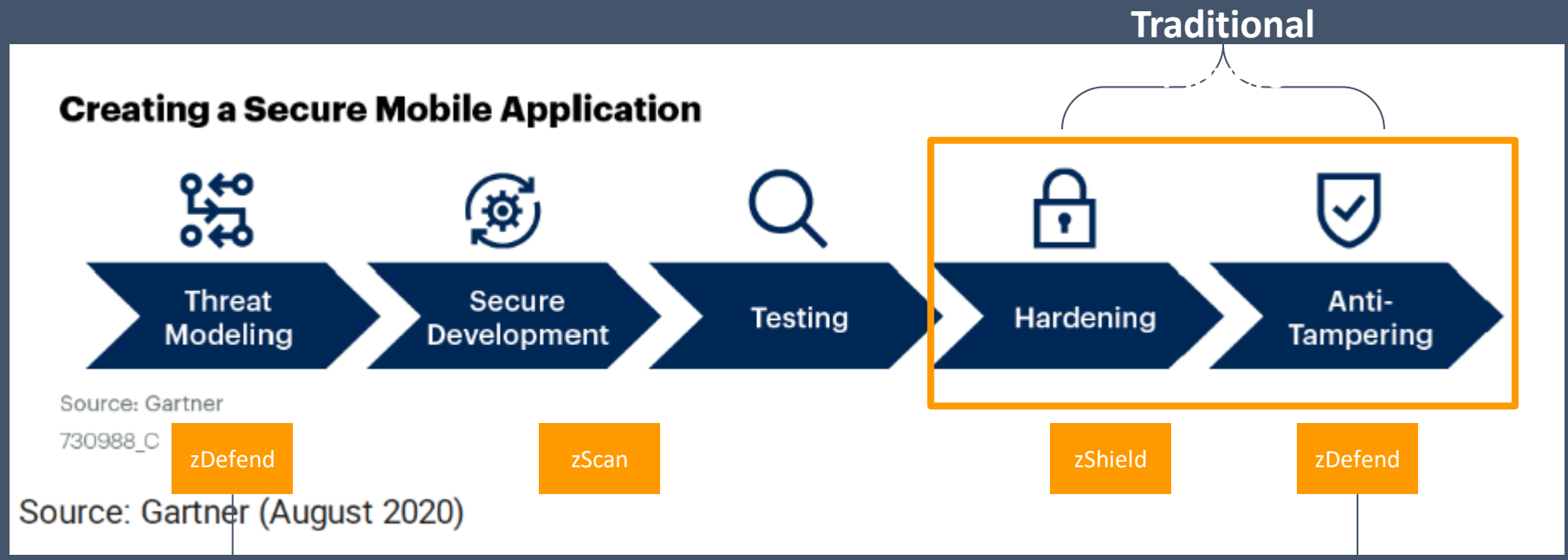


Comparações de recursos

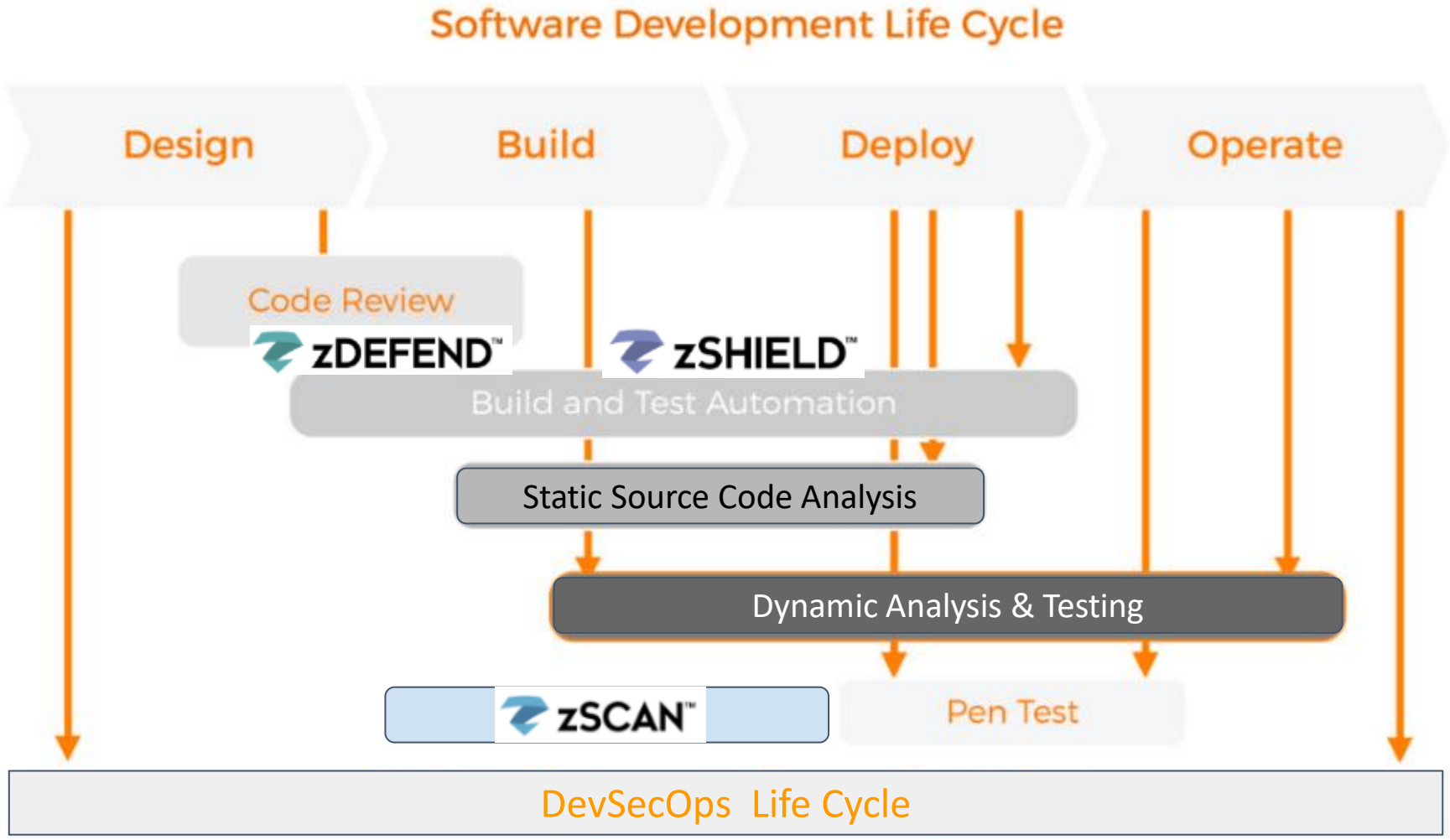
	zShield Android	zShield (iOS)	zDefend
Ofuscação de código e recurso	●	●	
Anti-Debug	●	●	●
Anti-Adulteração	●	●	◉
Jailbreak / Root Detection	●	●	◉
Criptografia Whitebox (SDK)	●		
Checksums / Verificação de Assinatura	●	●	
Swizzling & Hook Detection	●	●	●
SSL Pinning	●		
Detecção de Bot ou Clickjacking			●
Detecção Zero-Day / OS Tampering			●
MITM – Detecção de ataque do homem do meio			●
Risco de configuração do dispositivo			●
Detecção de Malware			●
Detecção de Vulnerabilidade			●
Forense de eventos de ameaças	●	●	●
Painel de relatórios	●	●	●

◉ Verificação comportamental avançada

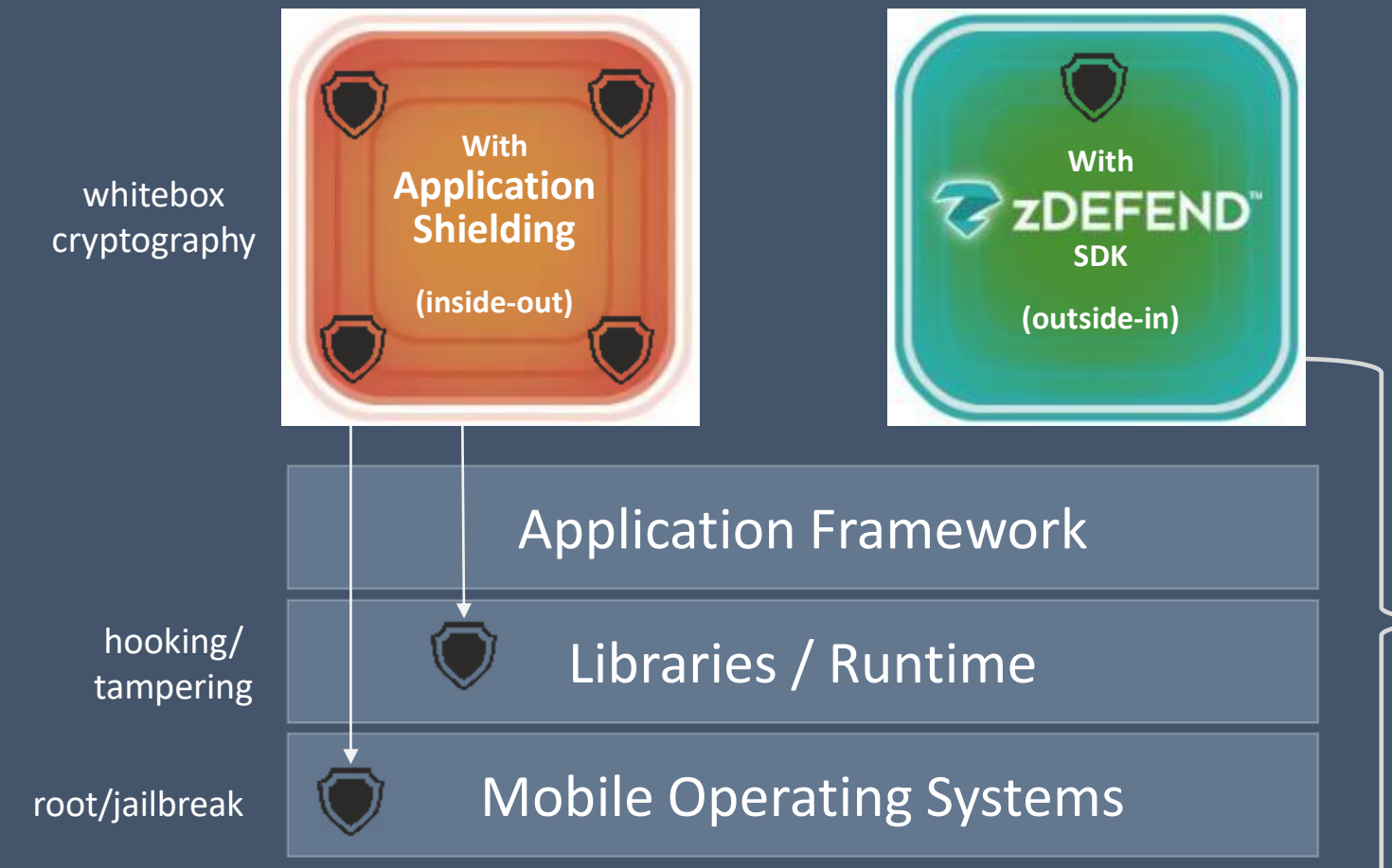
Segurança de Apps Segundo a Gartner



“Os líderes de segurança devem buscar envolvimento no início do processo de desenvolvimento de aplicativos”.



Traditional App Shielding vs zDEFEND



Signature-based Detect Known Threat Require App Update

Machine Learning-based Detect Known/Unknown Threat Dynamic Update

Route to Compromise

Abnormal Process Activity
Device Tampering
App Tampering / Hooking
Malicious Processes
File system changed
System Tampering

Internal Network Access
MITM
MITM - ARP
MITM - Fake SSL certificate
MITM - ICMP Redirect
MITM - SSL Strip
Rogue Access Point
SSL Downgrades
TCP Scans
Unsecured WiFi

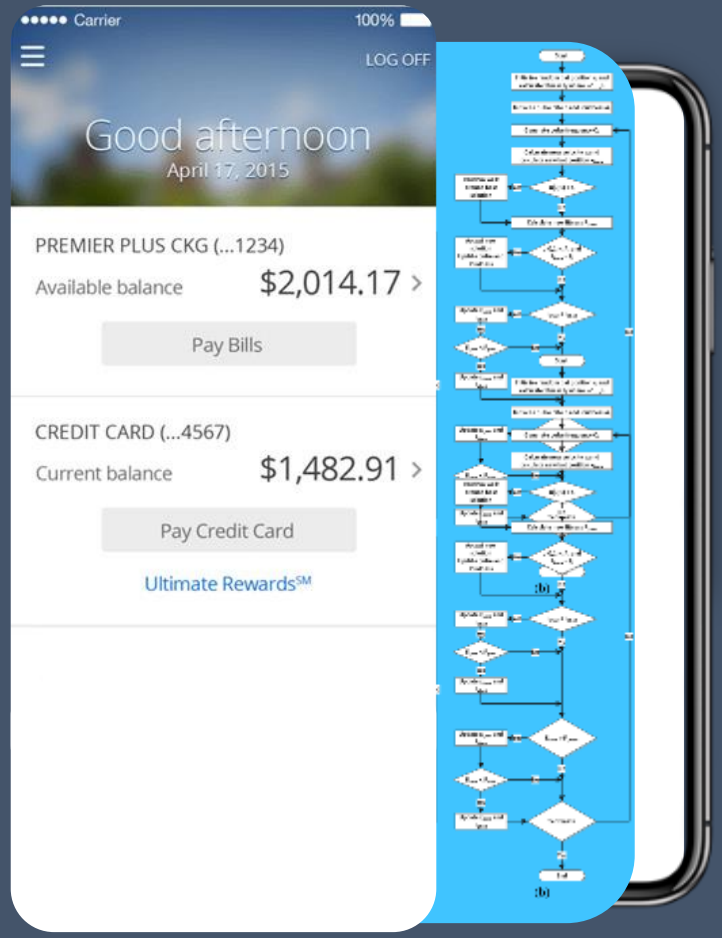
No Encryption
No Device Pin
Malicious Processes
SELinux Disabled
Sideloaded App(s)
Stagefright Vulnerability
Third Party App Store Enabled
USB Debugging Mode Enabled
Device jailbreaking / rooting
Debugger Detection
Emulator Detection

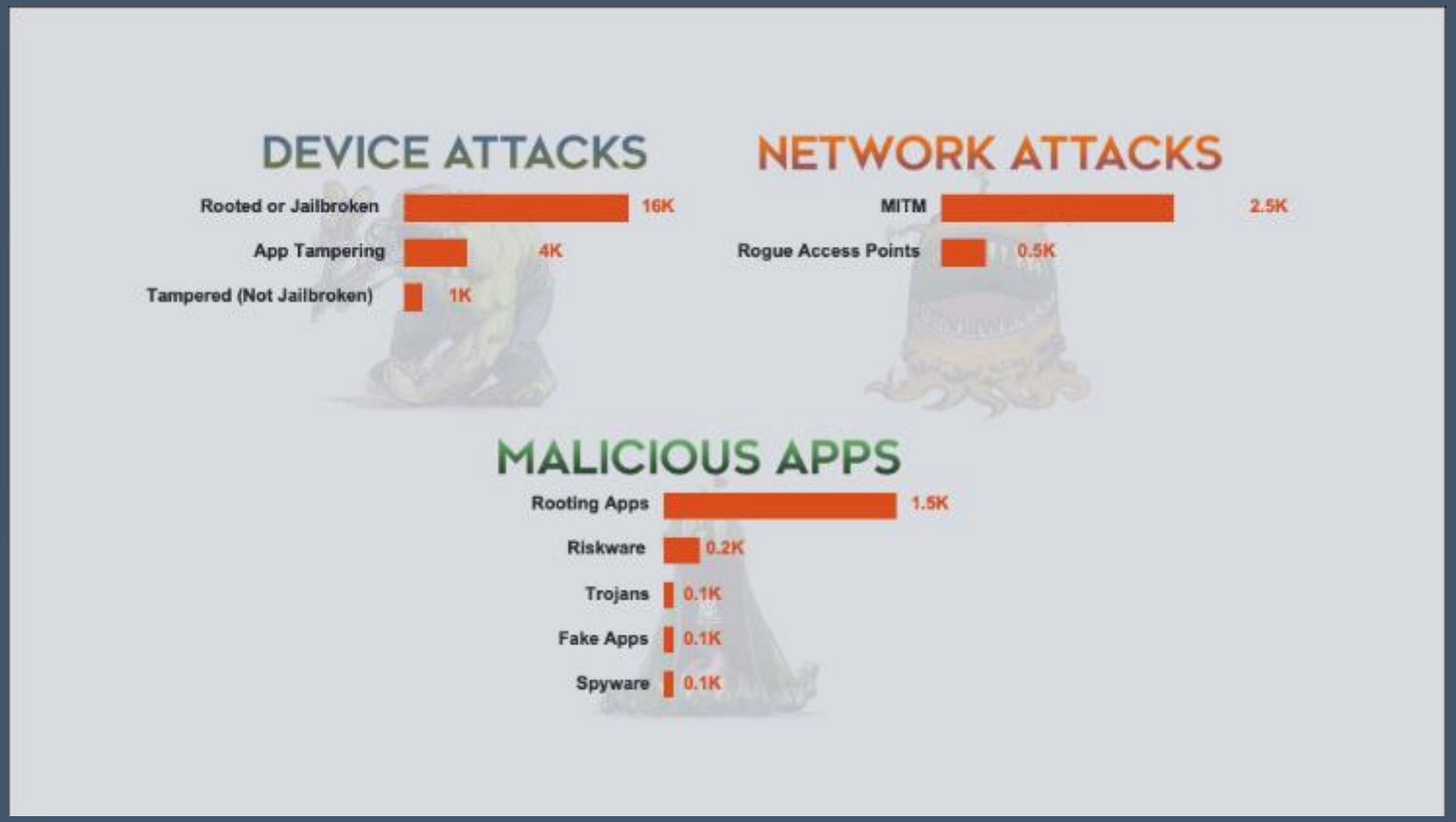
Malware

Phishing

Cases: Maior banco dos EUA - > 60M de usuários

Implantou Zimperium para reduzir mais de US$ 1 bilhão em fraudes mobile. Nos primeiros 30 dias, viram mais de 900.000 ameaças com muitas conexões com Wi-Fi ruim ou aplicativos instalados de lojas de aplicativos de terceiros (não seguras).





Ítens	Especificações Técnicas	Compatibilidade
a	Para bloquear o teclado virtual de terceiros de gravar pressionamentos de tecla digitados ao usar o aplicativo móvel	zDefend
b	Para evitar a captura de tela usando os recursos de captura de tela e instantâneo do Android na tela que contém informações confidenciais.	zDefend
c	Melhore e atualize constantemente a assinatura de bloqueio.	zDefend
d	Proteção / anti-violação de aplicativos contra hackers.	zDefend/ zShield
e	O aplicativo detecta e responde à presença de um dispositivo com acesso root ou desbloqueado, alertando o usuário ou encerrando o aplicativo.	zDefend
f	O aplicativo evita a depuração (debugging) e / ou detecta e responde a um depurador (debugger) sendo anexado. Todos os protocolos de depuração disponíveis devem ser cobertos.	zDefend/ zShield
g	O aplicativo detecta e responde a adulteração de arquivos executáveis e dados críticos em sua própria sandbox.	zDefend
h	O aplicativo detecta e responde à presença de ferramentas e estruturas de engenharia reversa amplamente utilizadas no dispositivo.	zShield
i	O aplicativo detecta e responde quando está sendo executado em um emulador.	zDefend
j	O aplicativo detecta e responde a adulteração do código e dos dados em seu próprio espaço de memória.	zDefend/ zShield
k	Os mecanismos de detecção acionam respostas de diferentes tipos, incluindo respostas retardadas e furtivas.	Não Aplicável
l	A ofuscação é aplicada a defesas programáticas, que por sua vez impedem a desofuscação por meio de análise dinâmica.	zShield

m	**O aplicativo implementa uma funcionalidade de 'vinculação de dispositivo' usando uma impressão digital do dispositivo derivada de várias propriedades exclusivas do dispositivo.**	**zDefend**
n	Todos os arquivos executáveis e bibliotecas pertencentes ao aplicativo são criptografados no nível do arquivo e / ou códigos importantes e segmentos de dados dentro dos executáveis são criptografados ou compactados. A análise estática trivial não revela códigos ou dados importantes.	zShield
o	Se o objetivo da ofuscação é proteger cálculos sensíveis, um esquema de ofuscação é usado que seja apropriado para a tarefa específica e robusto contra métodos de desofuscação manuais e automatizados, considerando a pesquisa publicada atualmente. A eficácia do esquema de ofuscação deve ser verificada por meio de testes manuais. Observe que os recursos de isolamento baseados em hardware são preferíveis à ofuscação, sempre que possível.	zShield
p	O cache do teclado é desativado em entradas de texto que processam dados confidenciais.	zDefend
q	O aplicativo remove dados confidenciais das visualizações quando movido para o segundo plano.	Não Aplicável
r	O aplicativo implementa vários mecanismos em cada categoria de defesa (f a k). Observe que a resiliência se escala com a quantidade, diversidade e originalidade dos mecanismos usados.	zDefend/ zShield
s	A ofuscação é aplicada a defesas programáticas, que por sua vez impedem a desofuscação por meio de análise dinâmica.	zShield
t	O aplicativo impõe uma política mínima de segurança de acesso ao dispositivo, como exigir que o usuário defina uma senha de dispositivo.	zDefend
u	Implementação do recurso de detecção de Keylogger.	zDefend

Supported Platforms

Operating Systems:

- iOS – 10 and above
- Android – 5.1.1 and above
- Linux, MacOS and Windows

Language:

- Android: Kotlin, Java
- iOS: Objective-C, Swift
- Hybrid: Cordova, Ionic, React Native, Flutter, Xamarin

Deployments:

- On-Premise
- Cloud SaaS

Alguns clientes: