Akamai

WHITE PAPER

Anatomia de um ataque a APIs

Entendendo a BOLA e as explorações de gerenciamento de inventário

Introdução

A maioria das equipes de segurança agora entende que a busca proativa de ameaças é um elemento essencial de um programa de segurança empresarial eficaz, especialmente quando se trata de APIs (interfaces de programação de aplicações). As APIs geralmente fornecem acesso direto a dados, funcionalidades e fluxos de trabalho. E, embora as medidas de segurança de perímetro de linha de base sejam amplamente usadas para proteger aplicações, o abuso de API e outros tipos de ataques estão aumentando. Na verdade, alguns dos incidentes de segurança de mais alto perfil que chegaram às manchetes nos últimos anos foram relacionados a APIs. Para entender melhor esses perfis de ataque, como a BOLA (Broken Object Level Authorization, autorização interrompida em nível de objeto) e explorações impróprias de gerenciamento de inventário, este documento irá:

- Revisar os fundamentos de APIs
- Explorar por que a segurança de APIs é um tópico de importância crescente
- Usar alguns incidentes de segurança de APIs de alto perfil para destacar as principais áreas de segurança de APIs
- Ilustrar os recursos necessários para realizar a busca de ameaças de APIs com eficiência

Noções básicas sobre APIs e ponto de extremidade

Para começar, vamos rever algumas terminologias básicas. As APIs são usadas para muitos fins, desde a funcionalidade B2C (Business-to-consumer) e a colaboração e integração B2C (Business-to-consumer), até funções internas de desenvolvimento e integração. As APIs da Web, que se comunicam pelo mesmo protocolo HTTP usado pelos navegadores da Web, são o modelo de implementação mais comum. A funcionalidade específica que essas APIs oferecem também pode ser chamada de serviços ou produtos de APIs.

Ao pensar em segurança de APIs, também é importante entender o conceito de um ponto de extremidade. Embora esse termo às vezes seja usado para se referir a dispositivos de computação do usuário final, ele tem um significado diferente no contexto de APIs. Você pode pensar em um ponto de extremidade de API como um único recurso acessível que faz parte da API, juntamente com a operação que pode ser realizada nele.

> Este é um exemplo simples. Um ponto de extremidade de API que retorna informações de pedido para uma empresa específica pode ser representado como: GET /orders/{orderID}. Nesse caso, GET é um método HTTP específico, enquanto orders e orderID representam o recurso específico que está sendo solicitado por meio da API.

Por que as APIs são o próximo grande desafio de segurança?

No passado, um invasor estava provavelmente de olho na violação de um data center corporativo para acessar e extrair os dados de uma organização a partir de um servidor específico. Ou podiam tentar inspecionar o tráfego de rede empresarial para capturar dados confidenciais. Nesses cenários, a busca proativa de ameaças pode se concentrar em atividades como testes de penetração para eliminar os possíveis pontos de entrada dos agentes de ameaça.

Em um mundo habilitado para APIs, essa dinâmica é diferente. Muitas APIs são inerentemente acessíveis a qualquer pessoa do mundo externo, com credenciais e chaves às vezes agindo como a única linha de defesa. E os agentes de ameaça estão cada vez mais aptos a comprometer esses elementos. Além disso, alguns dos tipos mais prejudiciais de abuso de APIs podem se originar de partes que receberam acesso a APIs, mas optam por usá-las de formas não sancionadas.

Ataques a APIs no mundo real

Na Akamai, 31% de todo o tráfego que protegemos é o tráfego de APIs. Esse aumento no tráfego de APIs leva a efeitos de downstream, como aumento de ataques e de abuso. A Gartner projeta que, em 2024, o abuso e as violações de dados de APIs dobrarão. Enquanto isso, muitas equipes de segurança estão presas em tentar recuperar o atraso. As APIs simplesmente continuam se multiplicando, enquanto as ferramentas de segurança de aplicações existentes oferecem proteção de APIs muito limitada.

Em 2024, as violações de APIs e dados praticamente **dobrarão.**[1]

As soluções atuais de segurança de aplicações são incompatíveis com APIs

31%
do tráfego da Web corresponde a APIs [2]

1 10 principais informações que os líderes de engenharia de software precisam saber sobre APIs

2 Os pesquisadores de ameaças da Akamai identificaram que 31% de todo o tráfego protegido pela Akamai é o tráfego de APIs

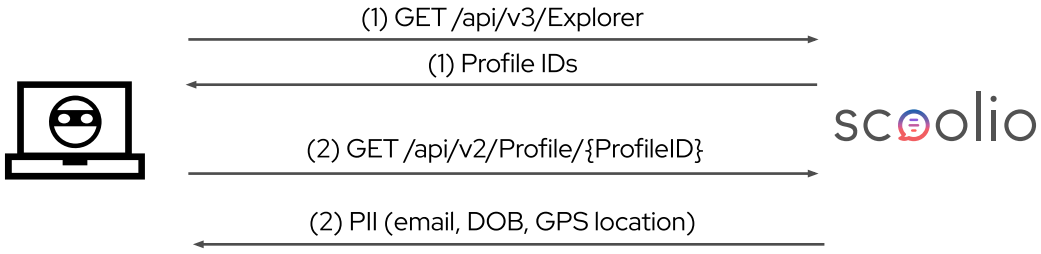
Para trazer essa questão à tona, vamos analisar um estudo de caso que ilustra o impacto real que os ataques a APIs podem ter sobre as empresas e seus clientes.

Estudo de caso

Apropriação de contas | Scoolio

Um exemplo de alto perfil é um incidente de 2021 que afetou a aplicação educacional alemã Scoolio. A aplicação coleta informações abrangentes dos usuários alunos. Por exemplo, ele realiza testes de personalidade, fornece recursos de rede social e chat e gerencia atividades como planejamento de estudos e aulas de reforço. Esses recursos incluem uma grande variedade de PII. A pesquisadora de segurança Lilith Wittmann descobriu uma vulnerabilidade de BOLA nas APIs da aplicação educacional que possibilitou a utilização de duas chamadas de API para acessar PII e outros dados para qualquer outro usuário da aplicação educacional.

Funcionava da seguinte forma:



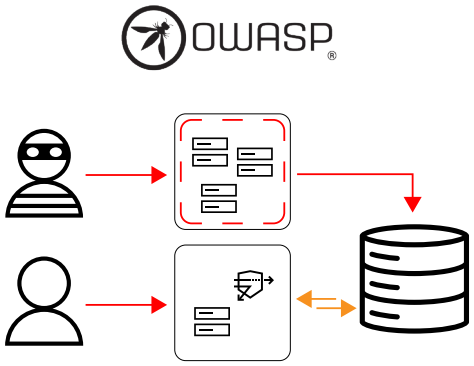
Etapa 1
Envio de uma chamada de API GET /api/v3/Explorer.
Essa chamada retornava UUIDs, que foram chamados de ProfileID nesta implementação.

Etapa 2
Envio de uma chamada de API GET /api/v2/Profile/{ProfileID}.
Essa solicitação retornava PIIs extensivas para o usuário relevante, incluindo e-mail, data de nascimento, localização GPS, entre outros.

O valor do uso de UUID

Embora ambos os cenários se concentrem no uso de UUIDs, o uso de UUIDs é, na verdade, uma prática muito boa. O uso de números gerados aleatoriamente em vez de uma sequência previsível de identificadores de usuário torna mais difícil para um agente de ameaça acessar informações para usuários em massa. O problema surge quando as informações de UUID são expostas de forma muito permissiva e combinadas com as vulnerabilidades de BOLA.

Gerenciamento inadequado de inventário



Outra faceta dessa exploração de vulnerabilidade de API é que ela aproveitou o gerenciamento inadequado de inventário, que é o número 9 na lista Top 10 de API do OWASP. Se você observar atentamente a sequência de ataque, perceberá que a primeira etapa é aplicada à versão 3 da API, enquanto a segunda etapa foi tomada contra a versão 2. Foram feitas melhorias na versão 3 que proporcionaram um acesso mais restrito às PII. No entanto, essas melhorias foram prejudicadas pelo fato de a versão 2, mais vulnerável, ter permanecido acessível a todos. Em última análise, as versões 2 e 3 foram afetadas pela vulnerabilidade de BOLA. Mas a presença desnecessária da versão 2 tornou o impacto da vulnerabilidade mais grave.

Quais medidas as organizações tomam hoje para proteger suas APIs?

Muitas organizações abordam a segurança de APIs concentrando-se nestes três pilares:

1. Autorização centralizada: primeiro, a implementação de um mecanismo de autorização centralizado para todas as portas de acesso à API reduzirá o risco de vulnerabilidade da API, eliminando erros no desenvolvimento que resultam em mecanismos de autorização com falhas.

2. Teste de API: uma segunda prática importante é o teste de API. O teste de todas as vulnerabilidades, especialmente a autorização interrompida, usando análise de código estático e testes dinâmicos, mostrará os problemas no início do processo de desenvolvimento.

3. Proteção de tempo de execução: o terceiro pilar fundamental é um conjunto de proteções de tempo de execução para o ambiente de produção. Mesmo as equipes mais proativas não detectarão todas as vulnerabilidades antes da implantação. Por isso, é essencial inspecionar o acesso dos usuários aos dados de produção e impedir a exploração de categorias conhecidas de vulnerabilidades na medida do possível.

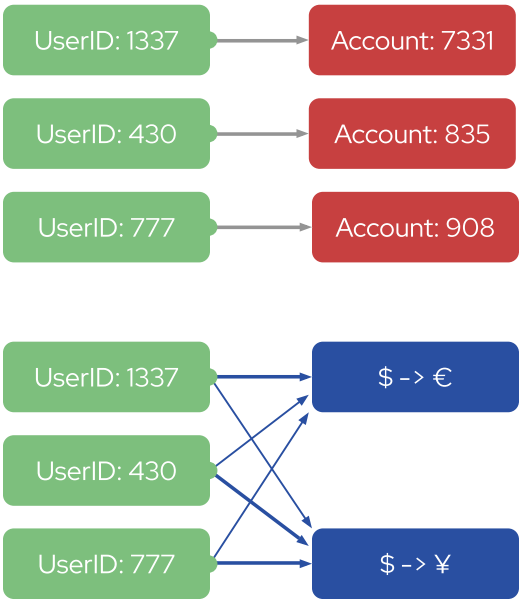
Essas três práticas fornecem uma excelente base para sua estratégia de segurança de APIs. Mas também é importante lembrar que elas não são perfeitas ou abrangentes. Por exemplo, mesmo organizações com autorização centralizada não têm garantia de que os desenvolvedores seguirão sempre as práticas recomendadas. E, por fim, as ferramentas de proteção de aplicações existentes geralmente são boas na detecção de padrões de ataque conhecidos, mas menos capazes de detectar ameaças mais sutis, como a BOLA.

Como você pode se basear nessa base com técnicas mais avançadas de detecção da BOLA?

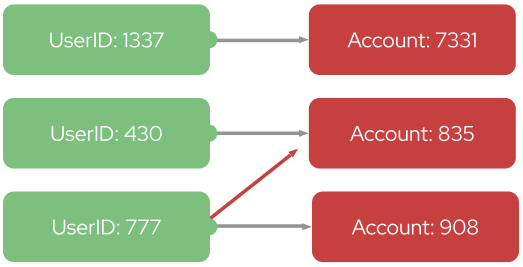
Uma das chaves para detectar e mitigar a BOLA e outras vulnerabilidades de APIs avançadas é modelar as relações entre as entidades envolvidas na atividade de APIs. Isso inclui agentes, como usuários, tentando acessar recursos, além dos próprios recursos. Se você puder mapear essas conexões entre as entidades de agentes e as entidades de processo de negócios que interagem com uma API, isso desbloqueará a capacidade de diferenciar entre atividade legítima e ilegítima ao analisar eventos de API idênticos.

Mapeamento de relacionamento ilustrado

Para entender melhor o mapeamento de relacionamento, considere este exemplo básico. Uma aplicação bancária oferece suporte a duas ações. Uma ação é ler os dados da sua conta, incluindo informações como saldo da conta, transações recentes etc. A segunda ação é visualizar as taxas de câmbio da moeda. A relação entre usuários e recursos nesses exemplos é muito diferente. O acesso às informações da conta deve ser limitado a um único usuário. Por outro lado, a função de taxa de câmbio deve estar geralmente disponível para todos os usuários.



Embora esse seja um exemplo muito básico, a criação de um modelo mais sofisticado de mapeamentos de relacionamento entre entidades torna muito mais prático prevenir ou detectar a BOLA.



Aqui vemos um usuário tentando acessar uma conta que ele não possui. A chamada de API específica pode ser idêntica, mas o contexto adicionado fornecido pelo mapeamento de entidade deixa claro que isso não deve ser permitido.

Detecção avançada de ataque de BOLA na prática

A seguir, vamos aplicar esse conceito a exemplos mais complexos, como as vulnerabilidades do estudo de caso. Veja abaixo trechos das entidades envolvidas no cenário:

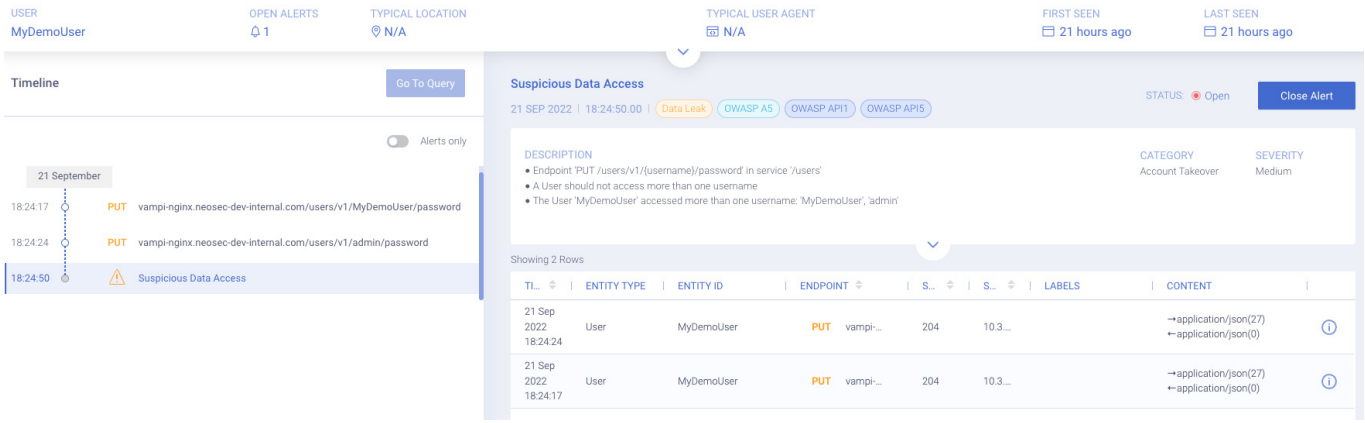
scoolio

GET/api/v3/Profile/ {ProfileID}

Cabeçalhos:

– Autorização: <MyAccessToken>

A entidade agente é destacada em verde e o recurso solicitado (o ID do perfil) é destacado em vermelho. Uma vez que esses relacionamentos são compreendidos, podem ser tomadas medidas para impor a lógica geral, como limitar o acesso de um agente a um único recurso, quando apropriado. Isso está longe do trivial, pois os relacionamentos podem ser mais complexos do que isso e incluem dimensões um-para-muitos. Mas técnicas como machine learning e análise comportamental tornam isso possível. Por exemplo, uma detecção bem-sucedida de uma vulnerabilidade de BOLA para um de nossos clientes seria assim:



Neste exemplo, uma vulnerabilidade de BOLA foi simulada em um ambiente de laboratório. Por meio do mapeamento de entidades e da análise comportamental, nossa plataforma detetou a BOLA e gerou um alerta rico em informações. Um analista de segurança ou caçador de ameaças que visualiza o alerta verá que o MyDemoUser acessou seu próprio perfil de usuário para alterar sua senha, uma ação sancionada. Mas logo após isso na linha do tempo, podemos ver que eles realizaram outra chamada de API para alterar a senha de administrador. Como esse é claramente um ato não sancionado com base na relação entre o agente e o recurso, o alerta foi gerado.

Onde começar com sua iniciativa de segurança de APIs

A segurança de APIs é um trabalho contínuo em andamento para a maioria das organizações. Isso pode tornar difícil saber por onde começar. Embora os três pilares fundamentais acima ofereçam um ponto de partida útil, a eficácia de sua abordagem aumentará bastante se você seguir essas três recomendações com sua implementação:

1. Certifique-se de ter um inventário de APIs sempre atualizado
2. Monitore ambientes de API de não produção e de produção
3. Imponha relacionamentos entre entidades

Você não pode proteger APIs que você não conhece. Portanto, a proteção eficaz de APIs começa com um inventário de APIs atualizado e uma avaliação da postura de segurança. Da mesma forma, à medida que você desenvolve seus recursos de monitoramento de segurança de APIs, é importante estendê-los para implementações de APIs de produtos e também que não sejam de produção. E, o mais importante, seu monitoramento e aplicação de APIs devem se estender além das ações sozinhas e considerar as relações entre as entidades envolvidas em sua atividade de APIs. Isso permitirá que você encontre vulnerabilidades e brechas de proteção e faça cumprir a conformidade com os modelos de uso de APIs pretendidos. Entender o comportamento em suas APIs permitirá que você veja qualquer abuso.

Interessado em entender mais sobre ataques a APIs e como você pode se proteger contra eles? Confira nosso detalhamento do Top 10 de APIs do OWASP.