API Security

SmartBear/Axway Webinar

September 10, 2015

* Security situation
  + Lot of recent security breaches with APIs
  + Mobile App vulnerabilities are often API vulnerabilities in disguise
* Vulnerabilities
  + Weak API Key Authentication
    - API keys in URLs
    - Amazon secures APIs using 2 keys
      * Secret key Id to perform HMAC signing
        + With detection of replay attaches
      * Access Key Id
* Depending on whom you are talking to, API security is defined in a specific way
* Order of checking
  + Access authorization
  + Parameter manipulation and data harvesting
  + Network eavesdropping
  + Disclosure
* IoT concerns
* Ability to Throttle and Quotas are important
  + Prevents miuse of DoS
* Owasp.org has the top security vulnerabilities
* Security Testing
  + Should be included as part of regression testing
  + Payload analysis
    - E.g. Tinder (mobile app, for people to meet) – though the app intended to show only that the person is “x miles away from you”, the API was delivering the exact geo-coordinates for the user
* Paying attention and respond quickly to initial information (most vulnerabilities are reported before they are exploited)

Q&A

1. What is the difference between web app security and API security?

API security is web app security???

1. Is OAuth and API Key enough?

OAuth is for delegated access (authorize an app to access an API on your behalf). Ironically you still have to secure the Oauth credentials.

1. Continuous integration environment and Ready APIs?

What type you are using. Ready API has a command line interface that allows you to integrate with continuous integration system

**Webinar: Taking the Threat Out of APIs: Top 10 Threat Protection**

October 27, 2015

**Q&A**

**Having tests repeatable. Any guidance on protecting legacy APIs that are not inherently designed for security? Without complete rewrite / re-engineering?**

the API Gateway pattern allows you to place security in front of the legacy (e.g. Plain Old XML / POX) APIs. You can also use the API Gateway pattern to create a virtualized REST API in front of the legacy XML API.

**As an API provider, how can I ensure credentials/tokens are not compromised by the client? For e.g., I can force them to use a client secret, but they might have the secret hardcoded/insecure place that will allow malicious user access to it & compromise**

good question - e.g. with many API providers they make it clear that the client must ensure that the client API keys are not compromised. So, it the responsibility is on the client. Often, an API Gateway is used at the client side to protect the keys. Many organizations use an API Gateway like the Axway API Gateway at the client side for this reason (security of API keys in the outbound direction)

**Which version oauth 1.0 or 2.0?**

For use cases that need end to end protection, I prefer signed security tokens which is older version of OAuth

**Recommendations for migrating from a username/password for an API to SAML or OAuth?**

I think either can work, see previous answer, if you need more specifics let me know

**What's the difference between using SAML vs. OAuth from a security perspective?**

SAML arguably has a stronger security profile due to digital signatures, however OAuth has wide adoption because developers find it easier to use (and they do not have to deal with XML). Important - BOTH need TLS

**Where does OpenConnect Id fit into this picture**

OpenID Connect (OIDC) provides a very userful "UserInfo" API endpoint to look up user attributes based on an OAUth Access Token. So it builds upon OAuth in a very useful way. At Axway we see it being used in this way, and is a useful part of ABAC (to obtain the attibutes used in ABAC)

**How can we take care of DDOS?**

DDOS is still taken care of at the network layer - e.g. using Akamai or AWS CloudFront. API Security doesn't replace protection for large scale DDOS

**Why is the statement - Oauth better than TLS? I was under the impression that both are needed in order to secure**

Both are needed. OAuth is at the app level, TLS is at the network level. TLS protects the channel through which OAuth passes

**what is difference between oauth and hmac message**

HMAC is for computing digests of messages, for example with Amazon AWS usage of two API Keys (Secret Key ID which is used for computing the digest, and the Access Key ID which is used for identification). OAuth is used for delegated authorization.

**What would be some security best practices for Hackathons?**

Great questions. I have seen the anti-pattern of hackathon organizers removing authentication from their APIs in order to "make it easier for them to be used in a hackathon". I don't recommend that. It is better to make use of an API Portal for the hackathon, where developers can authenticate and see sample API calls, including (for example) OAuth Access Tokens. At Axway our API Portal provides this generation of tokens, to help developers understand the security model of the APIs. A dedicated API Developer Portal for the hackathon also provides a simple "menu" of the APIs which can be used in the hackathon, with documentation, samples, etc

**Is it assumed that the API Gateway and the APIs themselves are local to each other -- IOW, a private connection between the API Gateway and the API servers?**

In my view, it’s best if they are physically and logically separate, but logical separation alone is way better than nothing

**What about JWT? How does that fit in?**

JWT (JSON Web Tokens) are often used as a way to package attributes as JSON, and associate them with an OAuth Access Token. In this way, they provide a great way to pass attributes about the user to the API itself, so that fine-grained Authorization can be performed right at the API (or, the API usage can be personalized for the end-user based on the attributes in the JWT token)

**Building a private REST API that will be used only by web apps and mobile apps developed within our org. Evaluating both SAML and OAUTH, what would you recommend?**

Either can work. As a security person I feel like I am in a defensible position with either. I would recommend looking at sample code and building out a reference implementation to see which works with your dev team. Salesforce.com has some sample code to look at for example

**If there are older proprietary or domain specific communication protocols that lack security how to add security to these communications? E.g. medical software protocols.**

First step is usually to ensure you do NOT do lowest common denominator - work toward highest level for each hop. For true legacy, sometimes out of band can work. Try to initiate or validate out of band if possible

**What is the best way to go about helping developers learn how to prevent SQL injections?**

OWASP has some great information, including example mod\_security rulesets which protect against attacks like SQL Injection at www.owasp.org [at Axway we embed mod\_security in our API Gateway to enforce these rules]

**What about HMAC that can be used without TLS?**

HMAC provides digests over messages, usually as part of authentication and ensuring message integrity (detecting if it has been tampered it). It doesn't provide encryption of the data itself. So you need TLS for the encryption part. In general it is recommended TLS 1.2 is used, as a given, even when using HMAC for digest of the message itself.

**What authentication mechanism would you recommend for api that are consumed by only internal applications? Should it still use api manager?**

In my view yes. For one thing it’s an admin convenience. For another it’s an API for security services that you would otherwise have to write yourself

**How can we leverage API gateways for context based authentication? Can we dynamically challenge for a second factor if detected that the context may have been compromised. In other words is there and authn/authz authorities that the API gateway can leverage?**

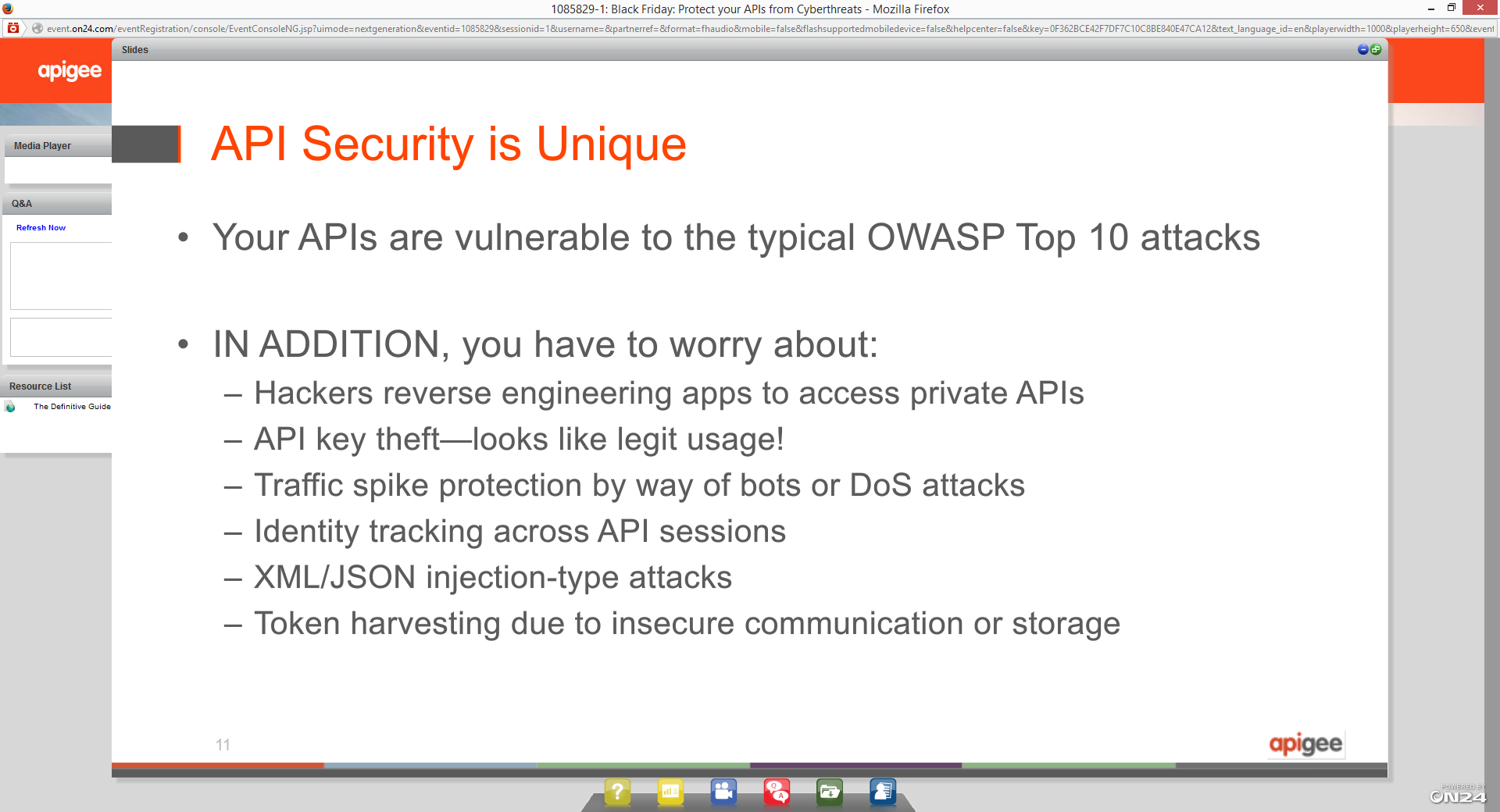
Super cool topic, risk based auth is usually a mix of fingerprinting - transactions, usage, patterns. Very powerful, but requires some back end data stores and analytics as well.

**How does this interact with other Axwway products like Sentinel?**

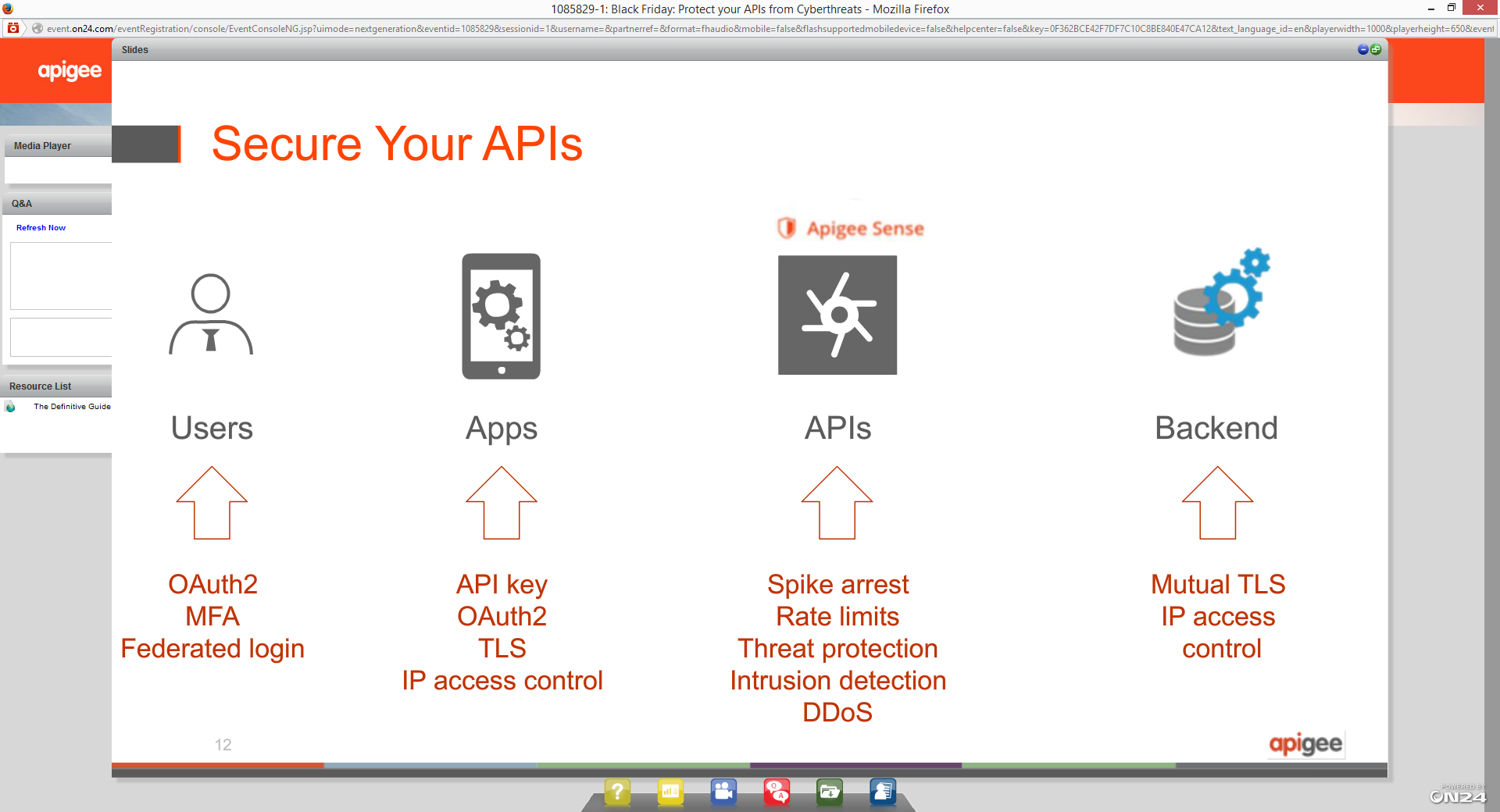
Axway Sentinel is integrated with the API Gateway, and can be used for monitoring. Axway Passport is also integrated, for customers who are using both. In addition, many Axway products have APIs of their own (e.g. Secure Transport) and the API Gateway, and Axway API Management in general, can be used to manage these APIs.

Apigee Webinar – Protect your APIs from Cyberthreats

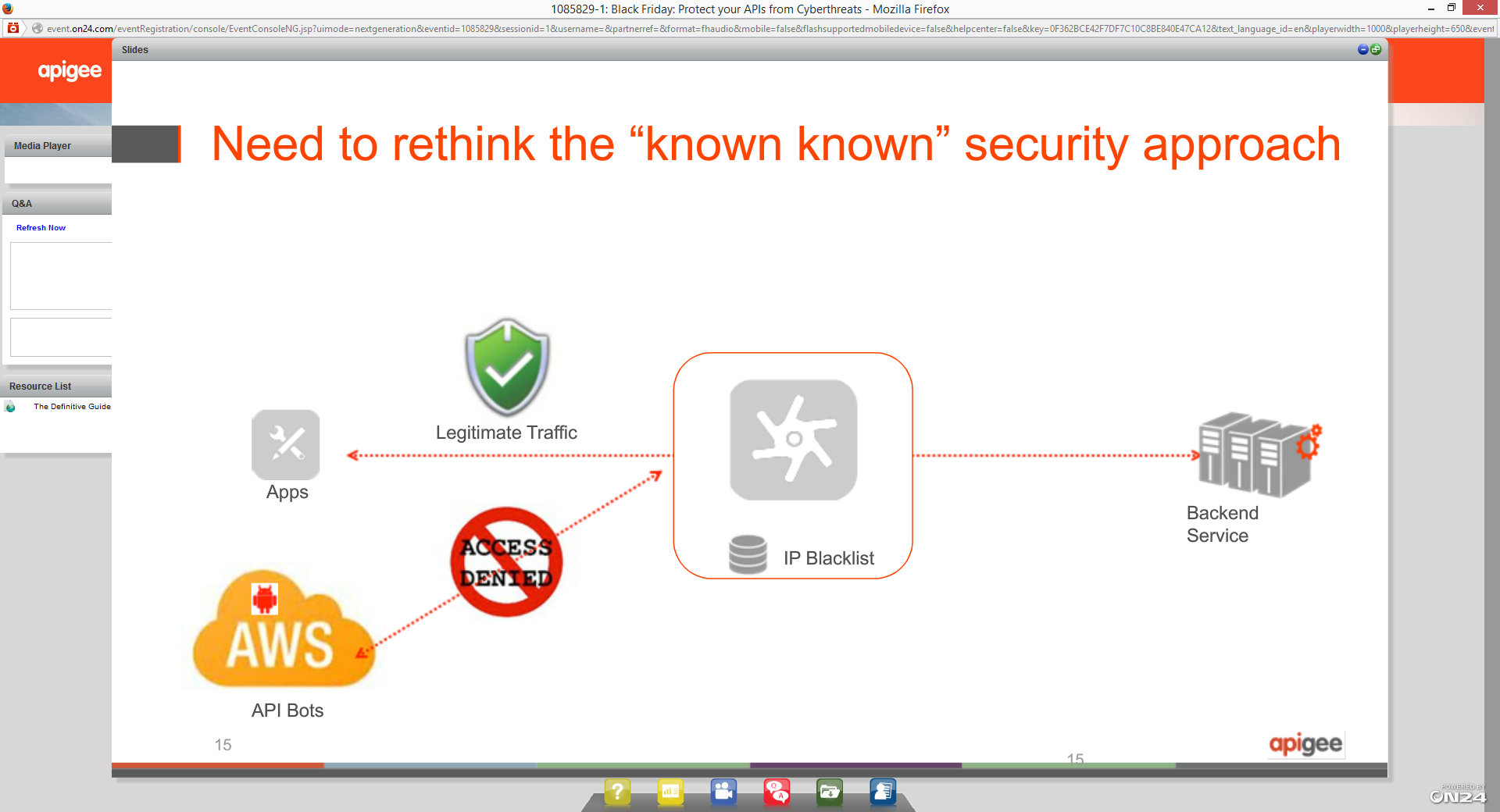
November 17, 2015



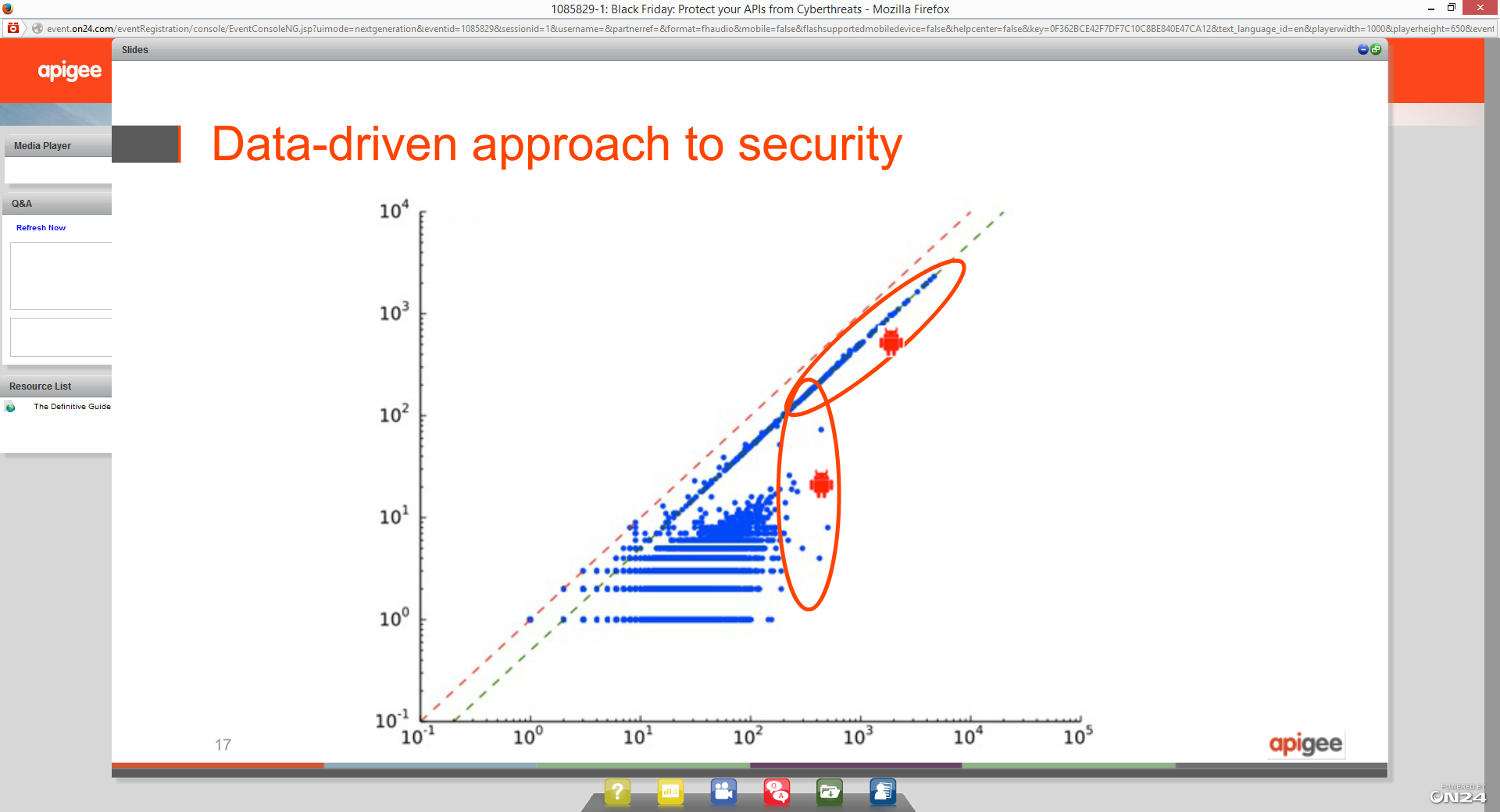
* OWASP is designed with known rules

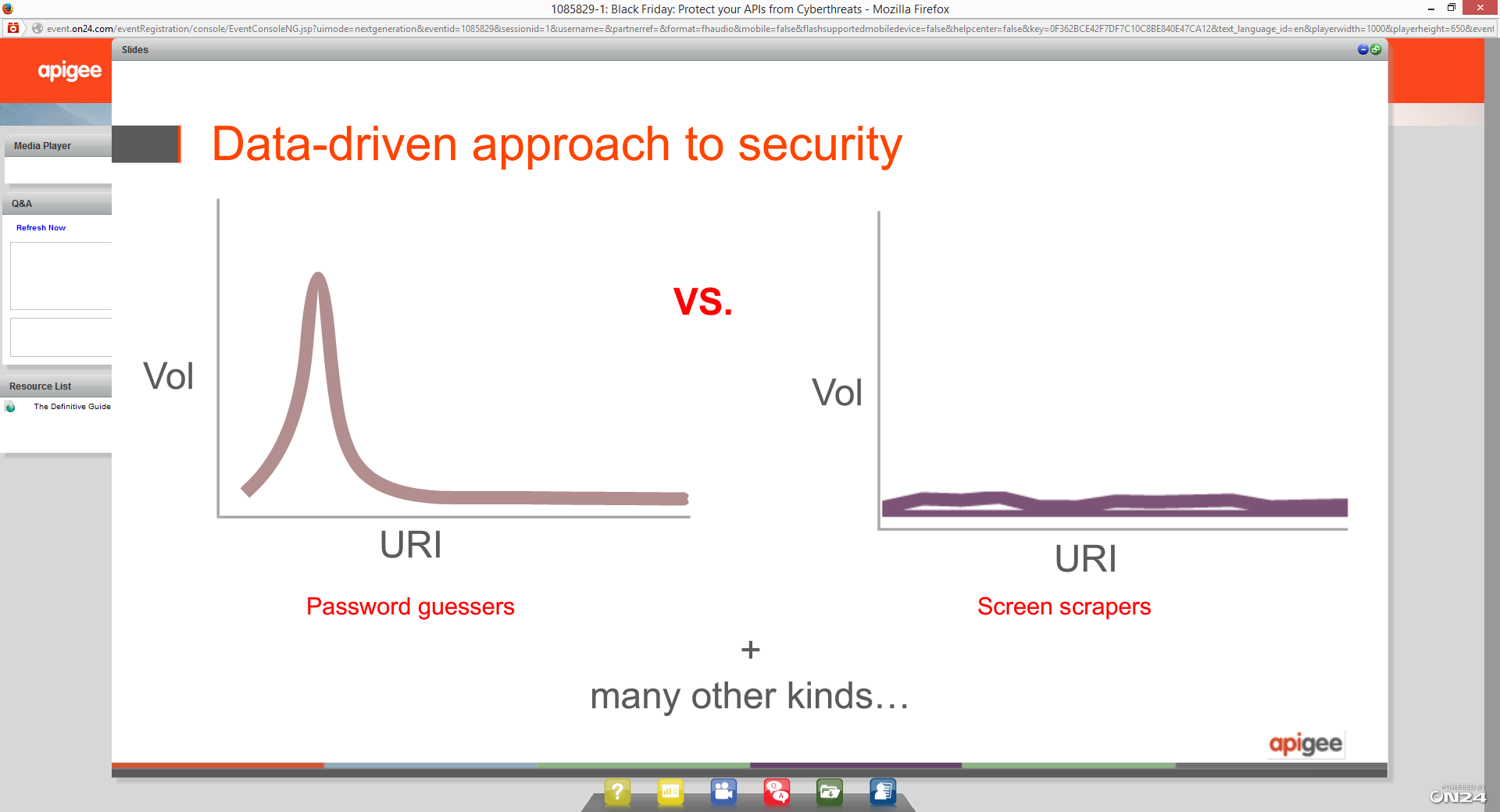


* Layers of security
  + At every point of security
  + API key based verification
  + OAuth2 – scope and authorization
  + Multi-factor authentication
* After you have set up your security policies
  + There are still threats – what if the user is coming in with the right credentials
  + Earlier model of security was “known-known”, but this is not going to work any longer and neither would it scale

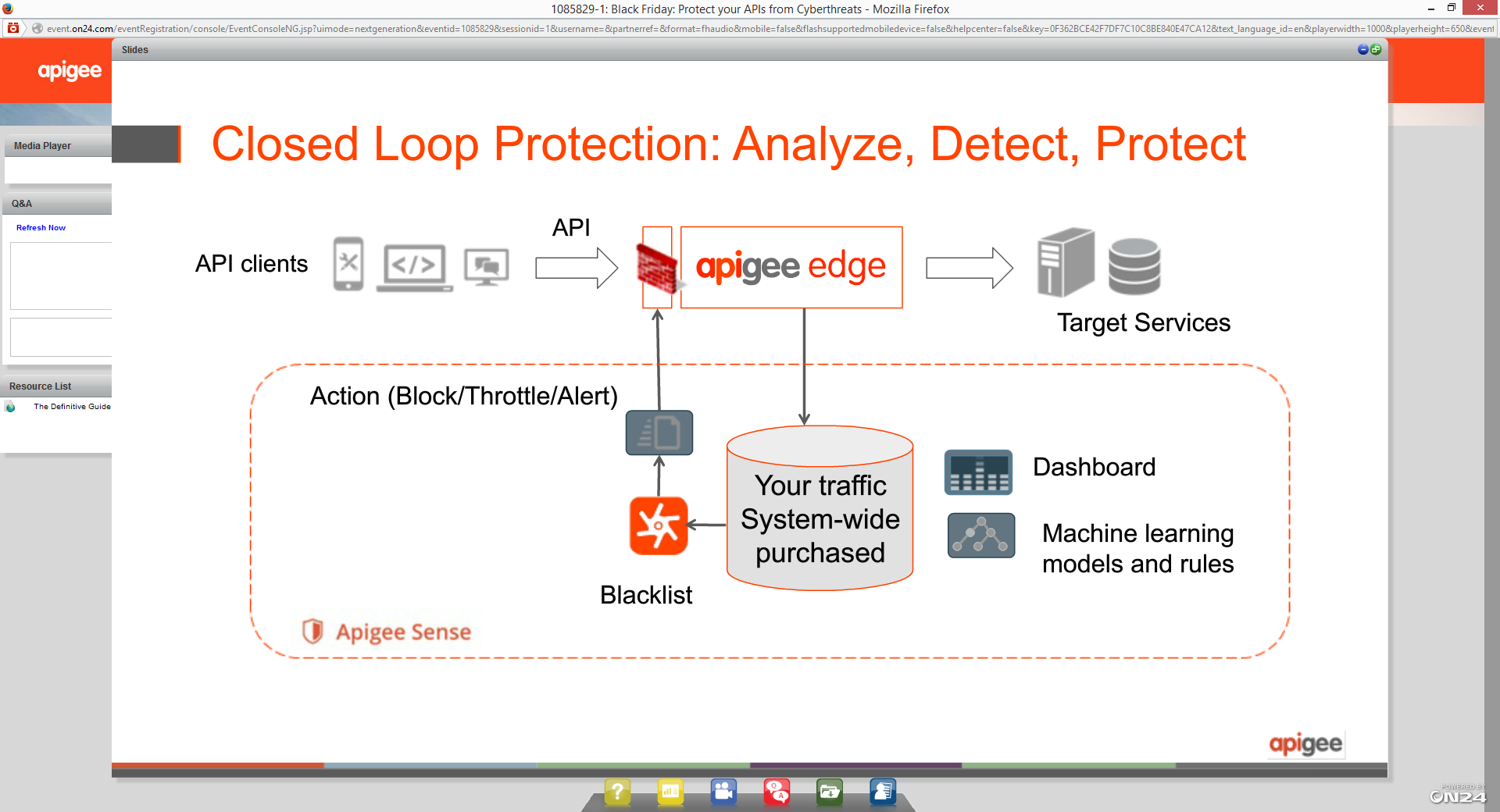


* How do you separate the good traffic from the bad traffic?
  + You need to have a set of rules and algorithms that would look at the traffic and try to figure out illegitimate/malicious traffic
    - Using API meta-data and intelligent algorithms (ML)

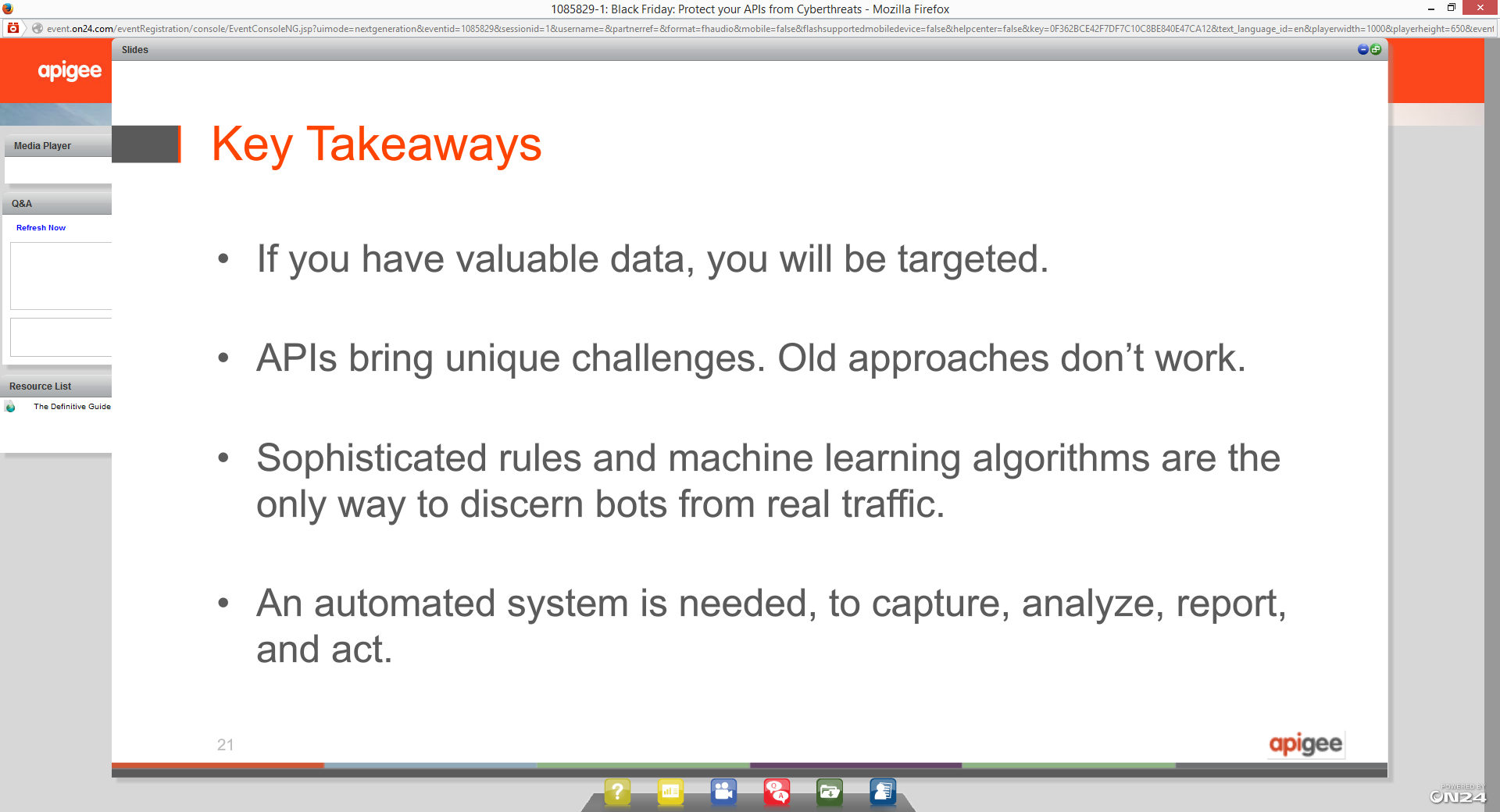


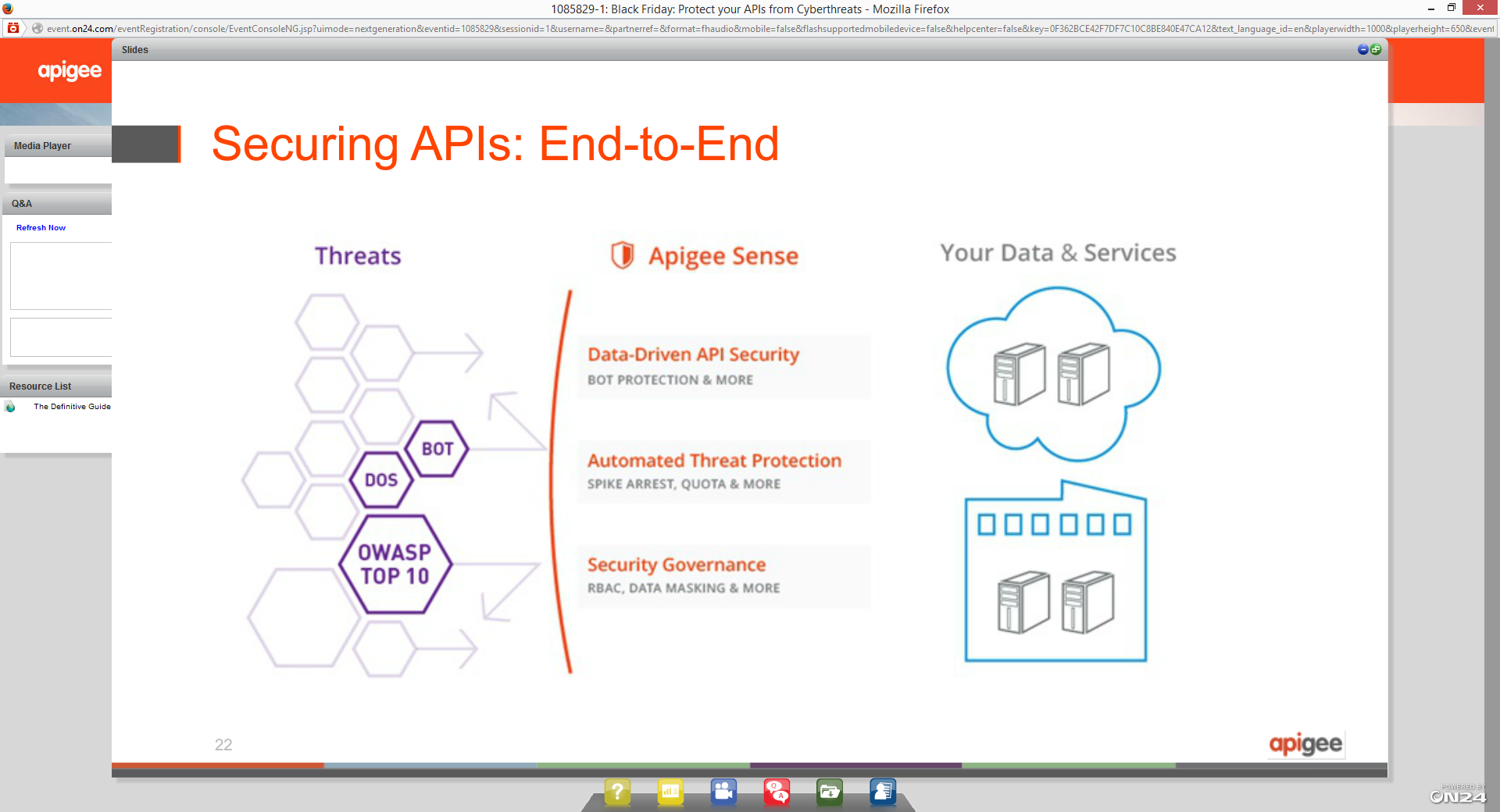


API Security: Data Driven Approach



* Key thing is that you have to do it in a way that the performance of the APIs are not impacted





Mark O’Neil, Gartner

