

Webinar Q & A - Managing Security Threat in SaaS: Essential "How To's"

Aspire Systems, an Outsourced Product Development firm committed to providing end-to-end product development services including SaaS enabling services, conducted a webinar on – Managing Security Threat in SaaS: Essential "How To's". SaaS Experts – Alexey Lef, Chief Technical Architect at SciQuest and Jothi Rengarajan, Senior Technical Architect at Aspire Systems shared their views on how to address security related concerns in building a SaaS solution.

Aspire witnessed excellent response for this webinar and received several interesting questions from the participants. For the benefit of registrants and other software providers, we have documented answers to all key questions that were received during the webinar. Thank you for all the questions!

[Q] How to achieve Federated Identity Management? What are the best practices to follow in implementing Federated Solution?

[A] There are many Federated Identity Frameworks provided by leading vendors such as Geneva Framework from Microsoft, OpenSSO from Sun, etc. You will be able to use them and achieve federated authentication faster. It is advisable to go with standards such as SAML rather than implementing some proprietary technology. You also have options of 3rd party identity providers such as Microsoft passport and OpenID providers which would make more sense in consumer SaaS.

[Q] How do you define uptime for mission critical business Apps and what technology is required?

[A] Before defining uptime, it might be important for us to understand what downtime means. It could simply be measuring if a system is up and running or if an important functionality of the product is available.

Uptime depends on a case to case basis and also on various factors such as if the system is available to public or not, the domain, nature of users, etc. Let's first look at what contributes to the down time.

There are 2 categories of down time:

#1 Planned down time – This is purely for maintenance activities like new releases, patch upgrades, system maintenance, back-up, etc.

#2 Unplanned down time – This is caused by unforeseen incidents like application level error, application crash, system crash, internet outage, power/backup power failure, etc.

For mission critical business applications it's advisable to define uptime at a modular level, instead of just one figure for the entire system. For example, if it's a banking application your transaction module can have a different uptime compared to personal settings module. Hence, the mission critical weightage could vary within the system. Hence, by keeping them separate it allows the flexibility to design/deploy and operate the systems accordingly.

In case of cloud providers, they provide SLAs around availability. Say if it is 99.95%, you need to check if their downtime + your downtime for maintenance is within your SLA (that you commit to your customers) limit.



Webinar Q & A - Managing Security Threat in SaaS: Essential "How To's"

Regarding your other part of the question on technology, uptime has no connection with technology. In this industry there are people who write crappy codes with sophisticated technology, as well as people who can write sophisticated code with a low end technology. Uptime is addressed through how you design, develop, test, deploy and maintain the application. In short, it's not about technology rather it's how you use it.

[Q] How can I best leverage Amazon WS to achieve security goals?

[A] By using Amazon, you can be assured on the physical security of data center.

Amazon EC2 provides web service interfaces to configure firewall settings that control network access to and between groups of instances which addresses one aspect of network security. Other security requirement such as data security and access control security has to be handled by the application.

Here is a reference to Amazon's security practice:

<http://aws.amazon.com/about-aws/whats-new/2008/09/05/amazon-web-services-security-whitepaper/>

[Q] How should vulnerabilities be managed in a SaaS-service?

[A] Vulnerabilities in SaaS can be broadly classified into the following areas:

Security Vulnerabilities for a web application – Since SaaS solutions are web based, all the vulnerabilities that any typical web application is subjected to will be applicable to SaaS solution as well. For example, SQL Injection – this is a vulnerability that is potentially possible in any web application. We have given reference for materials regarding this in the following question.

Data Security – This relates to how your data is organized, stored, retrieved and maintained. Data Security can be addressed via tenant data segmentation and tenant data encryption patterns, data audits, isolated schema, etc.

Network Security and Physical Security – This area deals with physical access of machines as well as their deployment in the network. In our webinar presentation you will find a reference to Amazon's web security practice, which clearly describes how Amazon has handled network and physical security. This will also give you an idea on what a typical data center should possess to address this vulnerability.

Access control security - Access control checks should be done strictly at product level.

[Q] Wouldn't Desktop As A Service - DAAS solve the need for high security on laptops - i.e data

[A] It does address the security but the level differs on case to case basis. DaaS addresses the point on data security, hence avoiding data lying in the laptop and user machines. However, it brings in other operational challenges. Ex: Bandwidth availability and user machines connectivity speed. DaaS may be easy to implement with few people, but (at this time) is not easy to scale out for a larger audience. In general, a lot of services are becoming available as on-demand, but are still in their early days. With the rapid growth of technology the on-demand services can very quickly become wide scale solutions



Webinar Q & A - Managing Security Threat in SaaS: Essential "How To's"

in the near future.

[Q] In the instance of integrating an in-house application with the provider's SaaS offering, how do I ensure the claims of the provider are true and my data is not getting compromised?

[A] I am assuming that you are talking about data level integration between SaaS system and your in-house system. It is advisable to use standards for the integration such as WS-*. WS-Security contains specifications on how integrity and confidentiality can be enforced on Web services messaging using which you will be able to solve the problem that you have stated. You can also utilize TLS (transport level security) such as Https in addition to the message level security.

[Q] Also, how do I implement single sign-on when I login from my corporate network?

[A] You can use the concept of federated identity for this. Please refer the answer for the 1st question.

[Q] Is it practical to build individual VPN's for corporate SaaS users to ensure some form of network security?

[A] There are several potential difficulties in implementing individual VPNs for every corporate customer.

- ✍ The customer's IT department may not have the skills required to configure/maintain a VPN router/server
- ✍ The customer's IT department may not have the resources or willingness to maintain a VPN with a SaaS vendor
- ✍ Configuring VPN in such a way that the customer's internal network is not exposed to the SaaS vendor (and potentially to other customers) may not be easy
- ✍ It may create additional complexity for remote users
- ✍ It is yet another potential point of failure

To decide whether a VPN is really necessary, I suggest that we first think about what we are trying to achieve here. I believe it comes down to two things:

- ✍ Ensure that the data is encrypted as it travels over the Internet
- ✍ Ensure that connections originate from the customer's network and not somewhere else

Data encryption is easily solved by using SSL. Even for server to server connection (e.g. integration points), there are various SSL libraries and packages available.

Origin authentication can be implemented in various ways. For server to server connections, shared secret or client SSL certificates can be used. It is a bit trickier for end-user (browser) connections. A simple option is to limit connection origin to a range of IP addresses. Spoofing an IP address is possible but difficult. It may be an acceptable solution in most cases.

Webinar Q & A - Managing Security Threat in SaaS: Essential “How To's”

[Q] Can you share any references for a Threat Model specifically created for a typical SaaS app?

[A] General web threat model still holds good for SaaS. Here are few references to it.

<http://msdn.microsoft.com/en-us/library/aa302418.aspx>

http://www.owasp.org/index.php/Main_Page

For more information :

Website : www.aspiresys.com

E-mail : janaki.jayachandran@aspresys.com

Phone : +91-44-67404000
+1-908-218-5017