

MÉTODOS DE REDUNDÂNCIA		MÉTODOS DE REDUNDÂNCIA
O que são <b>Tolerance Faults</b> ?		<b>Tolerance Faults:</b> ou <b>Tolerância á Falhas</b> , são ferramentas utilizadas pelos SGSI para garantir a redundância de funcionamento do SGSI. Essas ferramentas visam garantir que um sistem a continue funcionando mesmo que haja falta de energia, uma falha de hardware, uma falha de software e etc.
Como podemos <b>classificar o nível de redundância</b> que uma empresa precisa?		Podemos usar a " <b>Regra dos 9 da Redundância</b> ", a regra dos nove previne o quanto uma empresa deve estar segura de acordo com a sua disponibilidade de serviço, os nove já começa no 90% em um ano, assim:  - Se uma empresa consegue garantir <b>90% (1 nove)</b> de redundância em 1 ano, vai ficar 10% do tempo com seus serviços interrompidos, isso significa: 36,5 dias parada no ano / 3 dias e 1 hora parada por mês / 2 horas e 40 min por dia;  - Se uma empresa consegue garantir <b>99% (2 noves)</b> de redundância em 1 ano, vai ficar 1% do tempo com seus serviços interrompidos, isso significa: 3,65 dias parada no ano / 7,3 horas por mês / 14,4 minutos por dia;  - Se uma empresa consegue garantir <b>99,9% (3 noves)</b> de redundância em 1 ano, vai ficar 0,1% do tempo com seus serviços interrompidos, isso significa: 8 horas e 45 minutos parada no ano / 43 minutos por mês / 1,44 minutos por dia;  - Se uma empresa consegue garantir <b>99,99% (4 noves)</b> de redundância em 1 ano, vai ficar 0,01% do tempo com seus serviços interrompidos, isso significa: 52,56 minutos parada no ano / 4,38 minutos por mês / 8,64 segundos por dia;
- reservado para a questão acima -		De acordo com a nessidade do nosso sistema em ser redundante, por exemplo: - Nosso sistema, ou alguns departamentos dele podem ser interrompidos sem ocasionar em percas significativas para a nossa empresa? Caso sim, podemos trabalhar com um <b>sistema de 90% á 99% de disponibilidade</b> ; - O nosso sistema só pode ficar indisponível por pouquíssimo tempo, pois vários serviços dependem dele, por isso precisamos corrigir em pouquíssimos minutos uma perca de disponibilidade? Nesse caso, nosso <b>sistema deve ter 99,9% á 99,9999% de disponibilidade</b> ; - O nosso sistema não pode parar de jeito nenhum? Dele dependem vidas? Medicamentos? Nesse caso devemos trabalhar de <b>99,9999% á 99,9999999% de disponibilidade (O que chamamos de Nove Noves)</b> ;
Como podemos usar a Regra dos 9 para <b>saber o quanto o nosso sistema deve ser redundante</b> ?		Quando os administradores de um sistema <b>querem garantir que a redundância do seu sistema continue acontecendo mesmo que todas as medidas de redundância internas falharem</b> , eles aplicam a "Replicação de Redundância", que funciona como uma carta na manga. Ela consiste em manter um sistema externo operando e garantindo a continuidade de funcionamento do sistema quando todas as medidas de segurança falharem. Isso pode ser feito internamente, dentro da própria empresa, por exemplo, de uma filial para outra, como pode ser feito entre uma empresa e outra empresa externa especializada neste tipo de serviço.
O que é a <b>Replicação de Redundância</b> ?		Os sistemas que oferecem garantias de Replicação de Redundância devem oferecer o serviço SLA (Service Level Agreement - Acordo de Nível de Serviços), que é um serviço que traz as seguintes garantias:  - <b>MTTF (Mean Time To Failure):</b> é uma taxa que mostra ao cliente o tempo médio que o sistema leva para perceber que aconteceu um problema no sistema que ele quer proteger; - <b>MTTR (Mean Time To Repair):</b> essa taxa mostra o tempo médio que o sistema espera até que o defeito seja reparado; - <b>MTBF (Mean Time Between Failure):</b> essa taxa mostra o tempo médio entre o alerta de um possível defeito e a ocorrência do defeito aconteça; - <b>MTTSR (Medium Time To Service Repair):</b> Essa taxa mostra o tempo médio esperado para que o reparo atual aconteça sem que o sistema caia por completo;
Que <b>garantias</b> devem ser implementadas num sistema de Replicação de Redundância?		
- reservado para a questão acima -		