COLUNA

Coluna do Kurt

Backup seguro de sistemas

Criar backups é uma coisa, ter a certeza de que eles são seguros é outra. Conheça algumas dicas para garantir que o processo seja o menos doloroso possível.

Fazer cópias de segurança dos seus dados é fundamental. Se não criarmos backups utilizáveis regularmente, quaisquer interrupções, falhas de disco ou erros administrativos podem causar a perda permanente de dados. Mas, embora os backups abordem os aspectos de disponibilidade – e até certo grau de integridade – da tríade de segurança AIC (disponibilidade, integridade, confidencialidade), podem apresentar riscos significativos no que diz respeito à confidencialidade ou sigilo dos dados. Em outras palavras, quando centralizamos todos os dados no armazenamento removível (como, por exemplo, em fitas), as coisas podem ficar ruins muito rapidamente se as fitas forem extraviadas ou roubadas.

Backups criptografados

A solução, claro, é criptografar os backups. Dependendo do risco que estiver disposto a correr, uma criptografia forte pode até permitir o armazenamento de arquivos em locais potencialmente inseguros (por exemplo, em uma nuvem pública). Em geral, a maioria dos programas de backup suportam AES de 256-bit, que são extremamente fortes; portanto, com chaves corretamente geradas que estão seguras contra invasores, os dados permanecerão seguros por pelo menos algumas décadas. No entanto, devemos considerar várias outras questões ao decidir como aplicar criptografia aos backups e como gerenciar as chaves de criptografia.

Dados em trânsito e em repouso

De um modo geral, os dados são considerados "em trânsito" (quando enviados através de uma rede) ou "em repouso" (quando escritos para a mídias externas ou disco rígido). Dados em trânsito são vulneráveis à interceptação – até mesmo redes locais podem ser comprometidas e configuradas para espelhar dados para um servidor controlado pelo invasor. Isto significa que qualquer criptografia usada para proteger dados em trânsito também deve permitir ao cliente autenticar no servidor e vice-versa (por exemplo, para que o invasor não possa injetar dados maliciosos em backups que posteriormente serão utilizados para restaurar um sistema). Soluções típicas aqui incluiriam SSL com certificados ou softwares VPN, tais como o uso de certificados IPsec ou autenticação compartilhada. Dados em repouso também apresentam alguns desafios significativos já que podem possuir vida útil legível medida em décadas ou mais. Além disso, a densidade dos meios de armazenamento modernos significa que algo do tamanho de um livro de bolso pode conter praticamente todos os dados que são importantes para um usuário.

Controle de inventário é um recurso muitas vezes ignorado, mas, ter a certeza de que sabemos onde estão os backups e garantir que estejam sob controle, ou sob o controle de uma terceira parte confiável, é fundamental para garantir longa vida aos dados em repouso.

Cliente ou servidor?

Uma questão importante é onde fazer a criptografia. A maioria das soluções de backup são agora baseadas em cliente/servidor, pois quase todo mundo tem mais de um servidor para fazer backup, então precisamos de algum gerenciamento centralizado para manter tudo funcionando corretamente. Se criptografarmos os dados no cliente, os dados serão codificados de imediato e o risco de exposição é minimizado. A desvantagem, porém, é que a verificação de dados no servidor exigirá as chaves privadas para estar presente, o que poderia criar um alvo potencial para os invasores. Criptografar os dados no servidor centralizaria o gerenciamento de chave e permitiria realizar mais facilmente a desduplicação.

Gerenciamento de chave

Se perdermos as chaves dos dados criptografados ou as frases utilizadas para desbloqueá-los, também perderemos o acesso a eles. Isso também significa que, se as pessoas que

conhecem as senhas morrerem ou deixarem a empresa, também se perderá o acesso aos dados. Em outras palavras, destaque mais de uma pessoa para acessá-los. Dividir a frase (por exemplo, cada pessoa digita uma metade da frase) também pode impedir que alguém venda todos os dados para um concorrente. Rotacionar chaves também é uma boa ideia, em parte porque ajuda a reforçar as políticas para lidar com as mudanças de chave, que costumam acontecer quando as pessoas entram e saem de sua organização, mas também porque quanto mais dados criptografados utilizando uma chave única, mais sério será o compromisso atribuído à essa chave. Finalmente, certos requisitos legais podem afetar a retenção e acesso às chaves de criptografia, portanto, consulte um advogado, se necessário (estas exigências aplicam-se principalmente a informações financeiras, de saúde e informações pessoais, mas diferem amplamente por país, então é preciso manter-se informado a respeito).

Tipos de dados

Antes de falar sobre o software de backup, mencionaremos brevemente os diferentes tipos de dados que precisam de resguardo. O primeiro tipo (e mais simples) é simplesmente de onde devemos pegar o arquivo inteiro. O próximo tipo é de dados estruturados, tais como bancos de dados, serviços NoSQL e assim por diante. O desafio aqui é que não podemos pegar diretamente os arquivos porque eles não estão em um estado consistente (os dados podem estar à espera de serem gravados, transações podem estar inacabadas etc). Nestes casos, desligar o serviço e depois pegar o arquivo raramente é uma opção pois interrupções não são possíveis. Isto significa que o software de backup realmente precisa comunicar-se com o aplicativo. Para os aplicativos mais populares (como a maioria dos bancos de dados), encontrar software de backup que possa se comunicar com eles e extrair os dados não deve ser muito difícil. No entanto, para muitas soluções NoSQL modernas e de Big Data, backups ainda podem ser um desafio, então o usuário poderia pensar em arquitetar seu software para tornar os backups possíveis (por exemplo, gravando dados de objeto para um arquivo de log ou enviando objetos para um servidor de backup para ser arquivado).

Backup de software

No mundo Linux há várias opções para softwares de backup. Dois dos mais populares são o Amanda [1] e o Bacula [2]. Ambos são em código aberto, e o Amanda é suportado pelo Zmanda, que constrói uma versão comercial do Amanda com recursos adicionais. O Bacula oferece suporte para o SQLite, MySQL e PostgreSQL. O Amanda também pode suportá-los, mas precisaremos basicamente criar alguns scripts para esvaziar os bancos de dados primeiro.

É claro que existem centenas de outros programas de backup de código aberto disponíveis – a maioria deles não são muito bons (ou seja, carecem de refinamento, recursos etc.) – assim, em geral, aconselhamos o usuário a aderir ao Amanda ou ao Bacula.

O Bacula atualmente também oferece desduplicação limitada a arquivo; no entanto, o recurso é relativamente simples: o usuário especifica uma "base" de backup (por exemplo, de um servidor padrão), e depois faz o backup de servidores adicionais que executam o mesmo software, onde o Bacula irá desduplicar no nível de arquivo. A boa notícia é que vários sistemas de arquivos já oferecem suporte a desduplicação, como o Opendedup [3], então esta tecnologia deve estar disponível no Amanda e no Bacula também, em algum momento.

Backup NoSQL

Embora quase todos os serviços NoSQL possuam redundância embutida e distribuição de dados, este recurso não ajudará se o usuário cometer um erro administrativo e apagar um monte de dados ou ainda se o centro de dados inteiro ficar alagado (como aconteceu recentemente durante a passagem do furacão Sandy pelos Estados Unidos). Com os sistemas NoSQL, normalmente possuímos três opções: voltar o sistema de arquivos (por exemplo, usando snapshots LVM), replicar os dados para fora do local, e usar uma ferramenta específica para extrair todos os dados.

Conclusão

Considerando o quão popular a Big Data está se tornando, não é nenhuma surpresa que as soluções e estratégias de backup estejam correndo atrás do prejuízo. Para soluções NoSQL, criar backups mantendo o funcionamento do sistema pode ser um desafio, logo, o planejamento para estas situações agora pode poupar muitas dores de cabeça posteriores. E, caso o leitor não esteja utilizando NoSQL ainda, as possibilidades são de que venha a utilizá-lo em um futuro próximo, portanto é melhor já tomar a frente da curva de aprendizado. Em algum momento, esperamos obter bons recursos de dumping construídos sobre programas de backup existentes, porque as coisas só devem começar a ficar maiores. ■

Mais informações

[1] Amanda: http://www.amanda.org

[2] Bacula: http://www.bacula.org

[3] Opendedup: http://www.opendedup.org