

Quando o incidente de segurança é o próprio fornecedor

O agente de programação da chinesa Z.ai enviava à nuvem da empresa, cifrado com uma chave que só ela tinha, o histórico dos projetos dos usuários, e o PL nº 2.338/2023 entrega a comunicação de um incidente assim a quem o causou

22 set. 2026 · v. 1.0 · Autorregulação · Ato privado de conformidade · China · Em vigor · art. 17 · art. 42 · art. 43 · art. 46

Fábio Cabral · ORCID 0009-0005-6669-0424 · CC-BY-4.0 · DOI 10.5281/zenodo.22902617

O QUE MUDOU

A Z.ai, empresa chinesa que desenvolve os modelos GLM, mantém um agente de programação chamado ZCode, que roda no computador do desenvolvedor e trabalha dentro das pastas dos projetos dele. Em 18 de setembro de 2026, um assinante pagante do plano de programação da empresa publicou, em blog pessoal, a análise que fez do aplicativo depois de notar que a pasta local do ZCode ocupava mais de 700 MB. Segundo esse relato, enquanto o usuário estivesse conectado, o ZCode empacotava o espaço de trabalho inteiro, com o histórico completo do Git, o cache de arquivos grandes e os registros locais de operação, cifrava o pacote e o enviava a um repositório de armazenamento da empresa na Alibaba Cloud.

Três detalhes do relato importam para o que segue. A chave capaz de abrir o pacote ficava só no servidor da empresa, de modo que nem o usuário nem o próprio aplicativo conseguiam decifrá-lo. Os dois controles de privacidade da tela de configurações, pela leitura que o autor do relato fez do código, não interrompiam a captura nem o envio. E a política de privacidade do produto, que ele arquivou no mesmo dia, não mencionava o envio do espaço de trabalho. O relato também registra um limite. O projeto comercial de 313 MB do autor falhou 564 vezes por exceder o tamanho máximo e nunca saiu da máquina. Saiu, e foi aceito pelo servidor, um repositório público pequeno, de 538 arquivos.

A empresa reconheceu o envio no mesmo dia, em comunicado aos usuários escrito em chinês, que chega a esta nota pelo relato do engenheiro. Segundo ele, o comunicado atribuiu o envio à função de indexação da base de código e admitiu que ela vinha ligada por padrão no período inicial do lançamento.

Em 21 de setembro, a conta oficial do ZCode no X publicou declaração em inglês. A empresa pede desculpas, dá a correção por concluída, abre o código do ZCode no GitHub sob licença Apache 2.0 e anuncia um canal permanente de notificação de vulnerabilidades, com recompensa proporcional à gravidade. Afirma que os dados de código referidos pela comunidade não estão retidos e nunca foram usados para treinar modelos. Informa ainda que, depois da correção, convidou duas entidades para avaliar a segurança, a Academia Chinesa de Tecnologia da Informação e Comunicação, instituto de pesquisa ligado ao governo chinês, e a empresa de segurança NSFOCUS. Pela declaração, as duas confirmaram que o repositório de armazenamento está vazio ou foi apagado e que a versão 3.14.0 do aplicativo não tem mais o caminho de envio. Os relatórios integrais, diz o texto, serão publicados em breve. Até a data desta nota, não foram.

No mesmo dia o engenheiro atualizou o relato depois de ler o código aberto. O repositório publicado tem apenas duas versões registradas, uma inicial vazia e outra com todo o código, de modo que o

histórico que mostraria como o envio foi retirado não está lá. E a avaliação posterior, observa ele, prova que o armazenamento foi esvaziado depois de 20 de setembro, mas não reconstitui o destino do que foi enviado antes do dia 18.

Nenhuma das fontes consultadas noticia atuação de autoridade pública, chinesa ou de outro país. Nada indica quantos usuários foram alcançados, nem se há brasileiros entre eles.

APRECIÇÃO

O que segue é apreciação do autor, e pode ser dispensada por quem queira citar apenas os fatos acima.

O Projeto de Lei nº 2.338/2023 tem resposta escrita para incidente de segurança, e ela está no art. 42. O agente de inteligência artificial comunicará à autoridade setorial a ocorrência de grave incidente de segurança, e o caput descreve as hipóteses pela consequência, risco à vida e à integridade física, interrupção de operações críticas de infraestrutura, graves danos à propriedade ou ao meio ambiente, graves violações a direitos fundamentais, à integridade da informação, à liberdade de expressão e ao processo democrático. Em volta dele estão o art. 17, que manda os agentes garantirem a segurança dos sistemas, e o art. 43, que os mantém sujeitos à legislação de cibersegurança.

Aplicado ao ZCode, o desenho tropeça antes de chegar à pergunta que interessa. O código de um cliente só entra no rol por analogia, como dano à propriedade, e ainda restaria demonstrar a gravidade. O dever de comunicar só nasce, pelo § 1º, depois que a autoridade setorial fixar prazo e critérios, e o dever de segurança do art. 17 vale nos termos de regulamento, de modo que, se o projeto virasse lei amanhã, nada seria devido no dia seguinte. Falta, além disso, a própria autoridade setorial de uma ferramenta de programação vendida do exterior.

A pergunta que interessa é de modelo. O art. 42 foi escrito à imagem da Lei Geral de Proteção de Dados Pessoais, e o seu § 2º reproduz quase palavra por palavra o § 2º do art. 48 da Lei nº 13.709/2018, a autoridade verificará a gravidade do incidente e poderá, caso necessário, determinar providências para reverter ou mitigar os seus efeitos. Os dois dispositivos partem da mesma cena, em que alguém de fora ataca o sistema, e quem o opera é o primeiro a saber e o encarregado de avisar. No ZCode a cena é outra. O envio não foi falha nem ataque, foi função do produto, desenhada, ligada por padrão e mantida pelo fornecedor. Quem descobriu foi o usuário, com as ferramentas do próprio computador, e quem comunicou foi quem causou, depois que a descoberta já era pública.

A LGPD tem um parágrafo que o projeto não copiou, e é nele que o modelo se inverte. O § 3º do art. 48 manda considerar, no juízo de gravidade, a prova de que os dados afetados foram tornados ininteligíveis para terceiros não autorizados a acessá-los. A cifra do ZCode fazia exatamente isso. Ninguém de fora conseguia ler o pacote, nem sequer o dono do código. Conseguia a empresa, porque a chave era dela. A medida que o direito brasileiro de proteção de dados trata como atenuante é, neste caso, o instrumento que dava ao fornecedor acesso exclusivo ao trabalho do cliente. O terceiro não autorizado, aqui, é o fornecedor.

Há uma ironia que diz o que alcança o caso no Brasil hoje. O código-fonte, que é o que o cliente perdeu de vista, não é dado pessoal. Mas o histórico do Git guarda, em cada versão registrada, o nome e o e-mail de quem a fez. Se houver brasileiros entre os afetados e se a Z.ai responder como controladora, o que incide é o art. 48 da LGPD, pelo resíduo de dado pessoal, e não o marco geral de inteligência artificial, pelo código.

Resta a pergunta de quem confere. A empresa escolheu as duas avaliadoras, fixou o objeto da

avaliação, que foi o estado posterior à correção, e anunciou os resultados antes de publicar os relatórios. Nem o art. 42 nem o art. 46, II, "c", que atribui à autoridade competente a edição dos procedimentos de comunicação de incidentes graves, dizem quem verifica o que foi comunicado. O caso não é isolado no ano. No incidente de julho de 2026, em que agentes da OpenAI alcançaram a infraestrutura de produção da Hugging Face, a apuração também foi conduzida por quem causou e validada por consultoria contratada por ele.

O caso mostra ainda o que funcionou, e isso precisa constar para que a crítica não seja lida como pedido de mais controle. A correção veio em três dias, sem autoridade nenhuma, por pressão de um usuário que pôde abrir o aplicativo e ver o que ele fazia. Isso só foi possível porque o ZCode roda na máquina de quem o usa. O mesmo envio, feito por serviço que processa tudo no servidor, não deixaria rastro onde o usuário pudesse olhar. O art. 42 não responde a nenhum dos dois cenários, porque entrega a comunicação a quem, em ambos, tem mais razão para não fazê-la. É o fornecedor como terceiro não autorizado.

MÉTODO

Leitura integral de duas fontes em inglês, congeladas em 22 de setembro de 2026 com sha256 registrado. A declaração da Z.ai no X foi obtida pelo espelho público api.fxtwitter.com, que devolve o texto integral da postagem, porque o X não serve o texto a cliente sem sessão. O relato do engenheiro foi extraído da página publicada, com o HTML original arquivado ao lado. A licença Apache 2.0 e a data de criação do repositório, 20 de setembro de 2026, foram conferidas na interface pública do GitHub.

As afirmações técnicas sobre o funcionamento do ZCode são do relato do engenheiro, obtidas por engenharia reversa, e estão atribuídas a ele no texto. Não foram reproduzidas. A empresa não contestou que o envio ocorreu. O comunicado da empresa de 18 de setembro, escrito em chinês, não foi lido pelo autor, e o seu conteúdo está referido pelo relato do engenheiro. Os arts. 17, 42, 43 e 46 do Projeto de Lei nº 2.338/2023 foram lidos na versão do autógrafo do Senado. O art. 48 da Lei nº 13.709/2018 foi lido no texto compilado do Planalto, congelado na mesma data.

PONTAS SOLTAS

- O número de usuários alcançados e a presença de brasileiros entre eles não constam de nenhuma fonte consultada.
- Os relatórios integrais da Academia Chinesa de Tecnologia da Informação e Comunicação e da NSFOCUS não tinham sido publicados até a data desta nota. O que a nota diz sobre as avaliações é o resumo feito pela própria empresa.
- A vinculação da Academia ao governo chinês está descrita em perfis institucionais e na cobertura, e não foi conferida em ato oficial.
- A ativação por padrão está referida pelo relato do engenheiro, que resume o comunicado em chinês, e não pela declaração em inglês de 21 de setembro, que não toca no ponto.
- A aplicação territorial do Projeto de Lei nº 2.338/2023 e da Lei nº 13.709/2018 a fornecedor sem estabelecimento no Brasil não foi examinada, e a qualificação da Z.ai como controladora é hipótese.

FONTES DOCUMENTAIS ARQUIVADAS

Declaração da Z.ai sobre o ZCode no X, 21/09/2026, texto integral em inglês (arquivado em 22 set.

2026)

sha256 a2fd1ea28354264e7ed4535a5cb70a44a78f8ddae3937587d34a21137991789f

Obra de terceiro: preservada no acervo com hash, não republicada.

ferstar, Inside ZCode, Silently Uploading Your Entire Git History to the Cloud, 18/09/2026, com atualizações de 19 e 21/09/2026 (arquivado em 22 set. 2026)

sha256 af223353ea08123229b66098f843f6a18ba55cff6917c4230f0a6441d58cb8c7

Obra de terceiro: preservada no acervo com hash, não republicada.

Fonte oficial: https://x.com/zcode_ai/status/2101844704933621971

COMO CITAR

CABRAL, Fábio. Quando o incidente de segurança é o próprio fornecedor. *Observatório IA + Direito*, nota nº 12, 22 set. 2026. Disponível em: <https://fabiocabral.pro/observatorio/2026-09-quando-o-incidente-e-o-proprio-fornecedor>. Acesso em: [data]. DOI: 10.5281/zenodo.22902617.