Ameaças atores abusam de contas do desenvolvedor da NPM seqüestrada

Data: 2025-08-16 11:02:09

Autor: Inteligência Against Invaders

Uma campanha de phishing sofisticada direcionada ao mantenedor da Eslint-Config-Prettty, um pacote NPM amplamente usado com mais de 3,5 bilhões de downloads, resultou em que o código malicioso foi distribuído a milhares de projetos de desenvolvedores em todo o mundo.

O incidente, descoberto em 18 de julho por reverteringlabs 'automatizado [Detecção de ameaças](#) O sistema, destaca vulnerabilidades críticas nas práticas modernas de desenvolvimento de software, particularmente os riscos associados às ferramentas de atualização automatizada de dependência.

Ferramenta de desenvolvimento de hits de phishing

O ataque começou com um e-mail de phishing cuidadosamente criado que seriveram a equipe de suporte oficial da NPM, usando um endereço falsificado e direcionando as vítimas para uma réplica completa do site da NPM hospedado em um domínio malicioso.

O mantenedor de Eslint-Config-Grettier foi vítima desse engano, fornecendo aos atacantes credenciais para publicar versões não autorizadas de vários pacotes sob seu controle.

Poucas horas depois de obter acesso, os atacantes publicaram versões maliciosas de vários pacotes, incluindo Eslint-Config-Prettier, Eslint-Plugin-Grettier, Synckit e outros.

Esses pacotes comprometidos continham scripts pós -instalação que implantaram o Trojan de acesso remoto (rato) do Scavagger (rato) direcionando -se especificamente ao Windows.

Embora as versões maliciosas tenham sido removidas em aproximadamente duas horas, a enorme popularidade do pacote – com média 36 milhões de downloads semanais – significa que o impacto foi substancial, apesar da estreita janela de exposição.

Brecha de combustíveis para automação

O alcance do incidente foi amplificado significativamente por ferramentas de atualização de dependência automatizadas como o DependBot do Github, que criam automaticamente solicitações de tração para atualizar as versões do pacote sem intervenção humana.

Pesquisar [revelado](#) Que mais de 14.000 pacotes declararam incorretamente o Eslint-Config-Greptetts como uma dependência regular, e não como uma dependência de desenvolvimento, tornando-os vulneráveis à instalação automática durante as construções de rotina.

As vítimas notáveis incluíram projetos das principais organizações, com o reversolabs identificando

46 arquivos pacote-lock.json contendo o hash da versão maliciosa.

Entre eles estava um projeto de código aberto de propriedade da Microsoft, demonstrando como até grandes empresas de tecnologia podem ser afetadas por ataques da cadeia de suprimentos.

A natureza automatizada dessas atualizações significava que muitas equipes de desenvolvimento incorporaram, sem saber, o código malicioso em seus ambientes de construção, potencialmente expondo tokens do GitHub e outras credenciais sensíveis aos invasores.

A empresa de gerenciamento de frotas de bicicleta Dott exemplificou essa vulnerabilidade, com seus sistemas automatizados detectando, aprovando e mesclando a atualização de dependência maliciosa sem a supervisão humana.

Enquanto os corredores hospedados no Github fornecem alguma proteção por meio de máquinas virtuais isoladas, as organizações que usam corredores auto-hospedados enfrentam maiores riscos de compromisso persistente.

Aulas de segurança de desenvolvimento

Este incidente ressalta o paradoxo do gerenciamento automatizado de dependência: enquanto ferramentas como as equipes de ajuda dependente permanecem atuais com [patches de segurança](#) eles também criam novos vetores de ataque que os atores maliciosos podem explorar. Especialistas em segurança recomendam várias medidas de proteção:

- **Implementar atrasos de atualização:** Evite entrar em atualizações de dependência, a menos que resolva problemas críticos de segurança, pois a maioria dos compromissos é detectada em dias.
- **Categorização adequada de dependência:** Claramente, separe as dependências de produção das dependências de desenvolvimento para minimizar instalações desnecessárias.
- **Configure construir fluxos de trabalho com segurança:** Configure ambientes de construção para evitar a instalação de dependências desnecessárias para os sistemas de produção.
- **Requer revisão de segurança manual:** Evite a fusão automática de atualizações de dependência sem os processos adequados de verificação e aprovação.

O compromisso representa parte de uma tendência crescente de ataques da cadeia de suprimentos visando pacotes populares de código aberto.

Com a crescente sofisticação das campanhas de phishing e a ampla adoção de ferramentas de desenvolvimento automatizadas, as organizações devem equilibrar a conveniência da automação com práticas de segurança robustas para proteger seus ambientes de desenvolvimento de ameaças semelhantes.

AWS Security Services:10-Point Executive Checklist - [Download for Free](#)