Kötü amaçlı yazılım ile yazılım hatası arasındaki fark nedir?

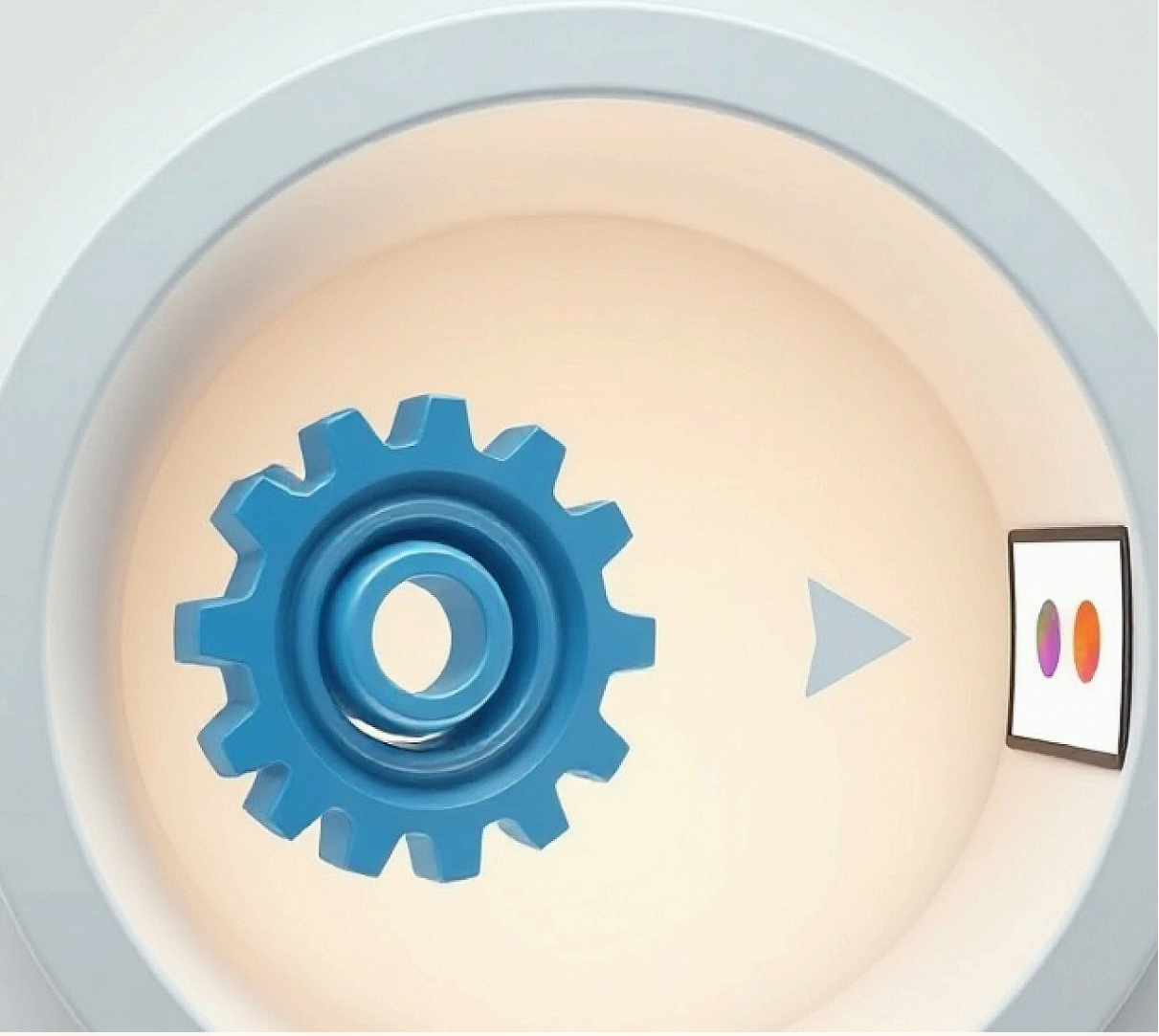


www.yzokul.com.tr

Kötü amaçlı yazılım ile yazılım hatası arasındaki fark nedir?



Şu soruyu görüyorsun: kötü amaçlı yazılım ile yazılım hatası arasındaki fark nedir? Kısa yanıt: hata kazara oluşan bir kusurdur, kötü amaçlı yazılım ise zarar vermek için bilerek yazılır. Bir kusur, programın yanlışlıkla yanlış davranmasına yol açar. Kötü amaçlı yazılım ise zararlı bir hedef gözetilerek oluşturulur. Fark, yalnızca programın nasıl davrandığında değil, niyet ve amaçtadır.



Her program belirli bir işi yapmak için kurulur; örneğin sayıları toplamak veya bir sayfa göstermek gibi. Bu amaçlanan iş, karşılaştırma yaptığımız istenen davranıştır. Bir programı kullandığında, tasarlandığı şeyi yapmasını beklersin. İstenen davranış, programın doğru çalışıp çalışmadığını yargılamak için ölçüttür. Amaçlanan işi bilmeden bir şeyin ters gidip gitmediğini anlayamazsın.



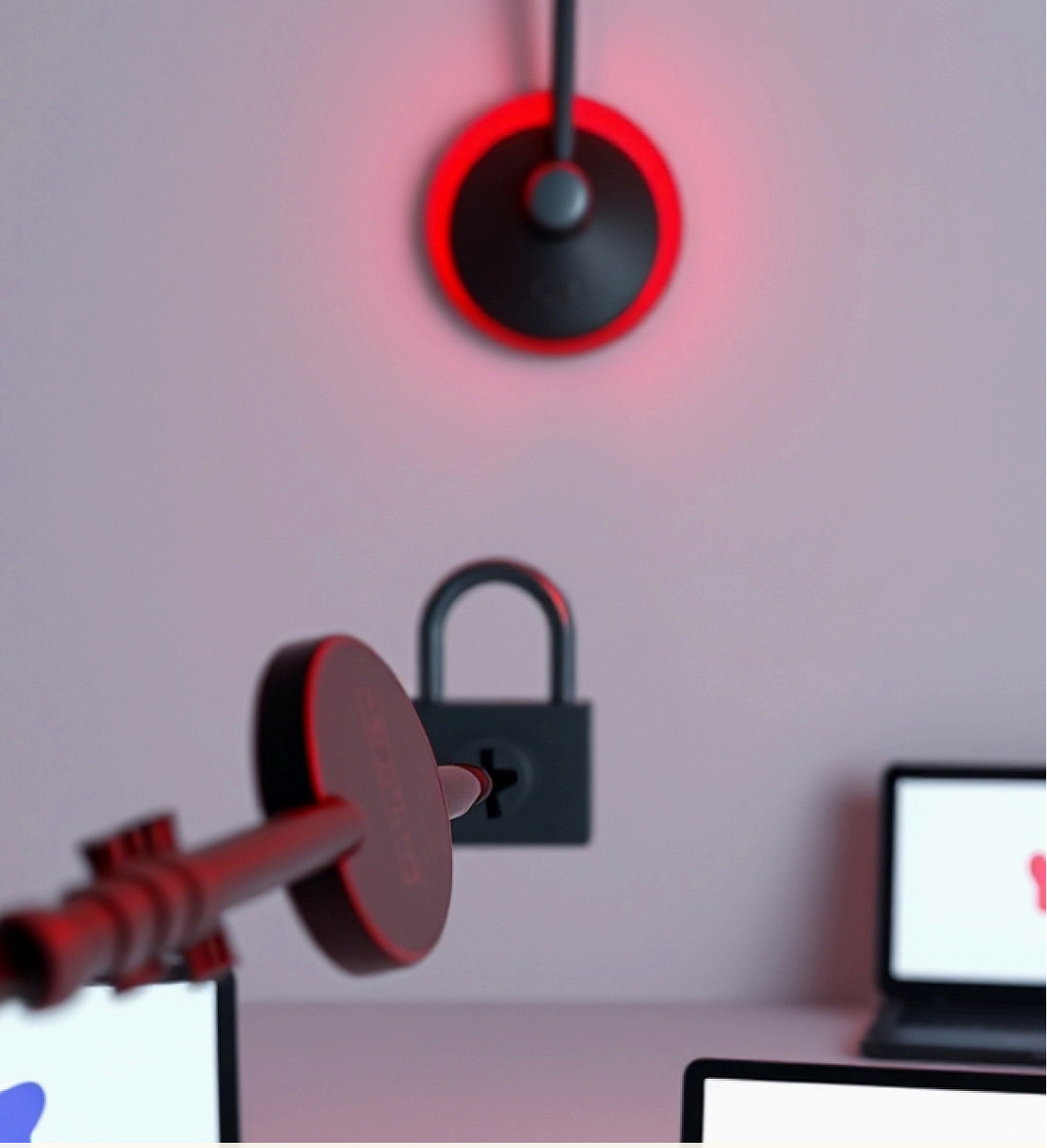
Amaçlanan işi yapmayan bir program bir sorun gösteriyor demektir. Örneğin iki sayıyı topladığında yanlış sonuç gösterebilir. İstenen davranıştan bu tür istenmeyen her sapmaya yazılım hatası deriz. Hata, koddaki teknik bir kusurdur. Programın olması gerektiği gibi davranmadığı anlamına gelir.



Hatalar kazara oluşur. Kod yazılırken yapılan yanlışlardan kaynaklanır, bir plandan değil. Kimse programın yanlış davranmasını istememiştir; bu yalnızca bir yazım yanlışı, mantık hatası veya yanlış anlama yüzünden olmuştur. Programcı programın başarısız olmasını amaçlamamıştır. Kusur istemsizdir ve genellikle beklenmez. Bu kazara oluşma niteliği, hatanın temel özelliğidir.



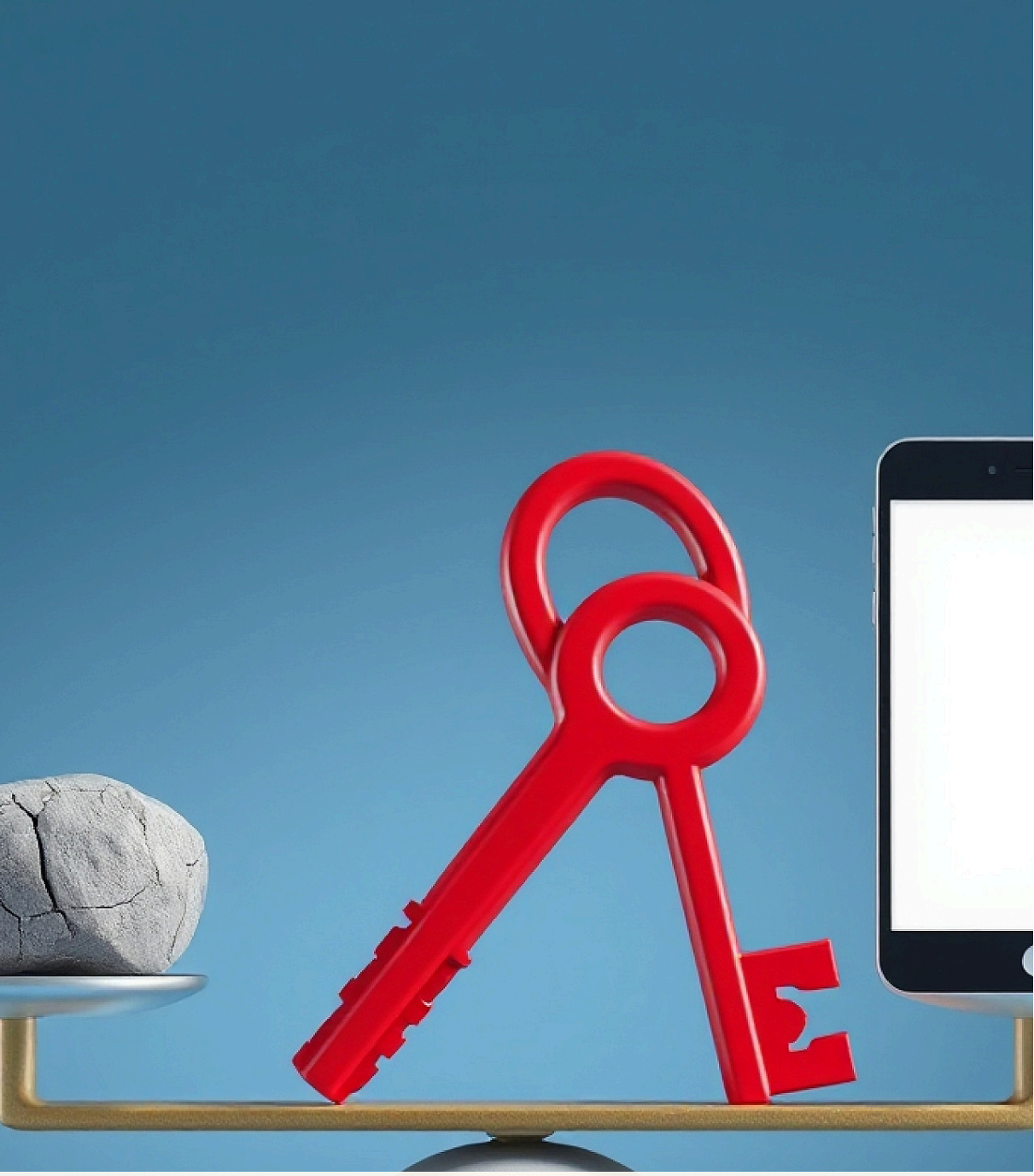
İki sayıyı toplaması gereken küçük bir hesap makinesi programı düşün. Koddaki bir yazım yanlışı yüzünden 2+2=5 veriyor. Bu basit bir hatadır: programcı kazara yanlış komutu yazmıştır. Bunun arkasında kötü bir niyet yoktur. Hesap makinesi doğru toplamak için tasarlanmıştı, ama bir kusur yanlış yanıt vermesine yol açtı. Yanlış tamamen tekniktir.



Kötü amaçlı yazılım, zarar vermek veya kötü şeyler yapmak için bilerek yazılmış koddur. Zararlı sonuç bir kaza değil, hedeftir. Birisi zararlı bir sonuca ulaşmak için onu bilerek oluşturur. Şans eseri ortaya çıkmış bir yanlış veya kusur değildir. Yazan kişi, yazılımın kendisine yarar sağlayacak veya başkalarına zarar verecek biçimde yanlış davranmasını amaçlar. Amaçlı zarar, onun tanımlayıcı özelliğidir.



Kötü amaçlı yazılımın belirli kötü hedefleri vardır. Bunlar arasında zarar vermek, yetkisiz erişim sağlamak veya başka kötü amaçlar yer alır. Her hedefi, kodu yazan kişi seçer. Örneğin veri çalmaya veya bir sistemi bozmaya çalışabilir. Zararlı niyet tasarıma yerleştirilmiştir. Yazılım meşru bir işlevi yerine getirmek için değil, bu amaçlara ulaşmak için oluşturulur.



İkisini yan yana karşılaştır: hata teknik bir kusurdur, kötü amaçlı yazılım ise seçilmiş bir amaçtır. Fark, yalnızca programın nasıl davrandığı değil, niyettir. Hata kazara olur; kötü amaçlı yazılım bilerek yapılır. Yalnızca belirtiye bakarak ikisini ayırt edemezsin. Davranışın zarar vermek için mi amaçlandığını sormalısın.



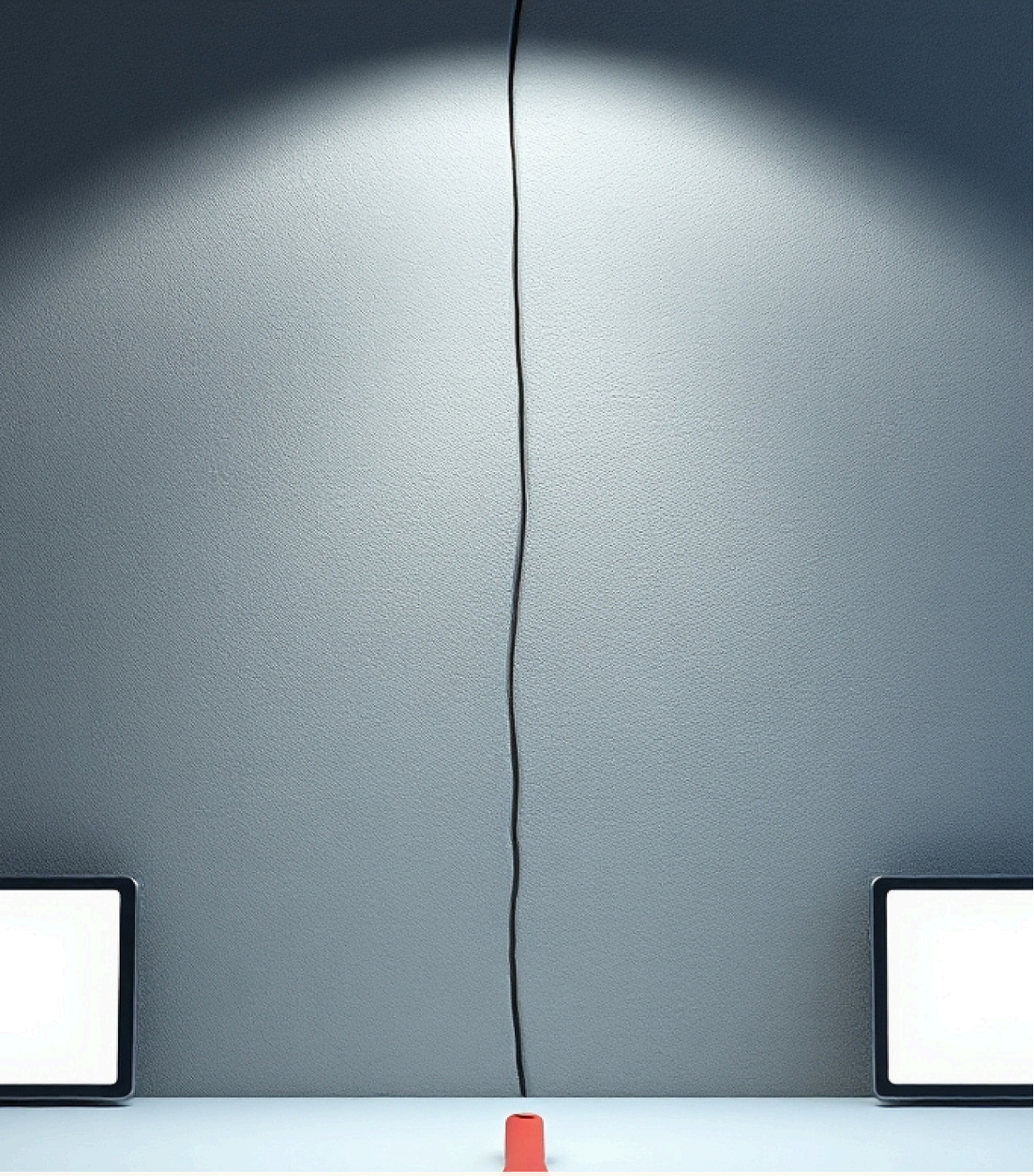
Çöken bir program mutlaka kötü amaçlı değildir. Tek başına çökme, yazılımın kötü amaçlı olduğunu göstermez. Birçok normal program sıradan hatalar yüzünden çöker. Çökme bir kusurun belirtisidir, kötü niyetin kanıtı değildir. Koddaki basit bir yanlıştan kaynaklanabilir. Yazılımın zararlı olup olmadığına karar vermek için daha fazla bilgiye ihtiyacın var.



Veri kaybı gibi tek bir kötü davranışın iki nedeni olabilir. Kazara bir hatadan veya bilerek yazılmış koddan kaynaklanabilir. Yalnızca belirti sana hangisi olduğunu söyleyemez. Davranışın arkasındaki amacı araştırmalısın. İstenmeyen bir yanlışsa hatadır. Zarar vermek için tasarlanmışsa kötü amaçlıdır.



Kötü amaçlı yazılım, içeri girmek veya zarar vermek için normal programlardaki hataları kullanabilir. Burada hata kapıdır, kötü amaçlı yazılım ise davetsiz misafirdir. Hata kazara oluşan bir zayıflıktır; kötü amaçlı yazılım bu zayıflığı kullanan kasıtlı koddur. İkisi birlikte çalışır, ama farklıdır. Hatanın kendisi kötü amaçlı değildir; yalnızca kötüye kullanılabilecek bir kusurdur.



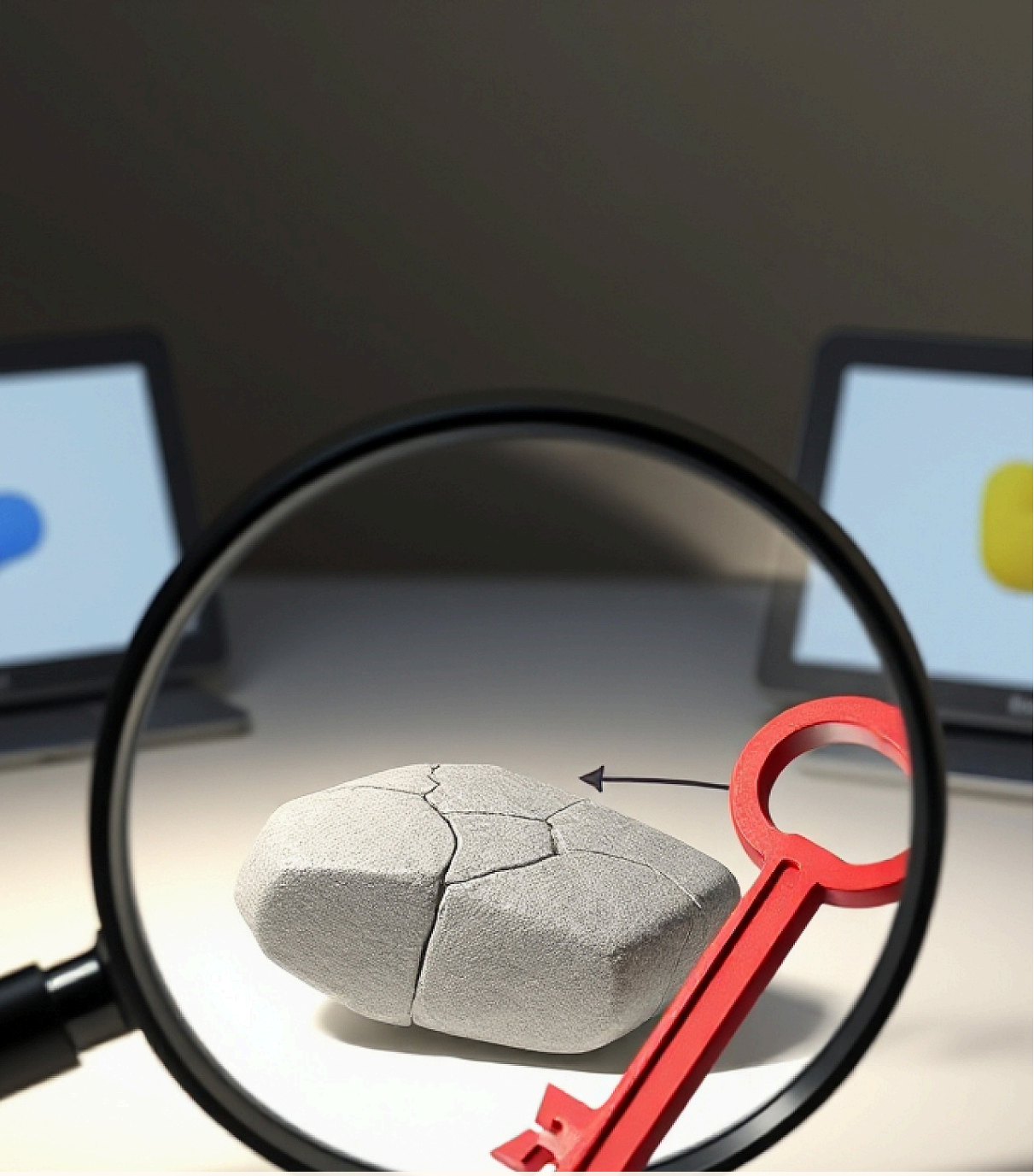
Kusurlu bir normal program ve bu kusuru kullanan ayrı bir zararlı kod düşün. Normal programda onu savunmasız bırakan bir hata vardır. Zararlı kod, bu hatadan yararlanmak için bilerek yazılmıştır. Birlikte görünürler ama farklı kavramlar olarak kalırlar. Hata bir yanlıştır; zararlı kod kasıtlıdır. Kusur ile saldırganı birbirinden ayırmalısın.



Yaygın bir yanlış, her çökmenin veya hatanın programın kötü amaçlı yazılım olduğu anlamına geldiğini düşünmektir. Çökme genellikle yalnızca bir kusurdur, bu yüzden bunun yerine amacı kontrol etmelisin. Davranışın zarar vermek için mi amaçlandığını sor. Kazara olduysa hatadır. Zarar vermek için tasarlandıysa kötü amaçlıdır. Tek bir belirtiye dayanarak sonuca atlama.



Karşıt yanlış ise hataların her zaman zararsız olduğunu varsaymaktır. Bir hata zarar vermek için yazılmaz, ama yine de gerçek sorunlara yol açabilir veya başkaları tarafından kullanılabilir. Örneğin bir hata verini kaybettirebilir veya birinin içeri girmesine izin verebilir. Hatanın kendisi kötü amaçlı değildir, ama etkileri ciddi olabilir. Kötü niyet olmasa bile hataları olası riskler olarak görmelisin.



İkisini ayırt etmek için tek bir soru sor: bu davranış birisi tarafından zarar vermek veya erişim sağlamak için mi amaçlandı? Yanıt evetse bu kötü amaçlı yazılıma işaret eder. İstenmeyen bir yanlışsa bu bir hataya işaret eder. Anahtar, amacı aramaktır. Niyet ikisini ayırır. Yalnızca belirtiyi değil, tasarımı ve sonucu da kontrol et.



Hata kazara oluşan teknik bir kusurdur, kötü amaçlı yazılım ise kötü şeyler yapmak için bilerek yazılmış koddur. Tek başına bir çökme ikisini de kanıtlamaz ve kötü amaçlı yazılım hataları kullanabilir. Amaç ile kusurun farklı kavramlar olduğunu unutma. Niyet sorusunu yol gösterici olarak kullan. Bu ayrım, sorunları doğru anlamana ve yanıtlamana yardımcı olur.