Título: ***New Cache Designs for Thwarting Software Cache-based Side Channel Attacks***
Referência: ***Zhenghong Wang and Ruby B. Lee, "New Cache Designs for Thwarting Software Cache-based Side Channel Attacks", International Symposium on Computer Architecture (ISCA'07), June 2007***
Autor do resumo: ***Vitor Monte Afonso – RA046959***

A confidencialidade dos dados hoje é assegurada usando-se principamente criptografia. Dessa forma, qualquer pessoa não autorizada que obtiver os dados criptografados não consiguirá entendê-lo. Os algoritmos criptográficos modernos estão protegidos de análise matemática e até de ataques de força bruta usando o poder computacional moderno. Entretando existe outro tipo de ataque, conhecido como ataque secundário (side channel attack) que utiliza outros meios, como tempo de execução e consumo de energia, para descobrir informações sobre a chave criptográfica. Ataques secundários na cache (cache-based side channel attack) impactam uma quantidade grande de softwares, já que praticamente todos os processadores modernos utilizam cache e ataques em software são relativamente fáceis de se implementar.

Existem dois tipos principais dessa forma de ataque. Em um deles o atacante executa seu programa simultaneamente com o programa da vítima, dessa forma ele verifica as variações em seus acessos a dados para detectar acessos errados à cache (cache misses). Assim ele pode descobrir pedaços da chave criptográfica da vítima. No outro tipo de ataque o atacante pode estar remoto e observa apenas o tempo total de execução do programa. Obtendo estatisticas sobre os tempos de execução do programa o atacante pode inferir sobre a chave criptográfica.

Estes ataques baseam-se no fato de que pode-se obter informações sobre a chave devido aos erros de acesso à cache. As soluções existentes baseadas em software são específicas a algum algoritmo de cifração e consistem basicamente de reescrever o programa de forma a evitar os ataques existentes. Entretando esse tipo de defesa não evita novos ataques. Já as soluções em hardware atacam a raiz do problema mas possuem problemas de custo e performance.

No artigo são propostas duas novas soluções baseadas em hardware que mitigam a raiz do problema e possuem impacto baixo no custo e performance.

A primeira é chamada PLcache (partition-locked cache),. Nesse método as linhas importantes da cache ficam trancadas de forma a criar algo como uma “cache privada”, onde acessos de outra cache privada não podem modificá-la, evitando que vazem informações internas aos programas através da cache. A outra solução chama-se RPcache (random permutation cache), que consiste na utilização de tabelas de permutação para que as informações geradas pelos acessos errados à cache sejam aleatórios e o atacante não consiga tirar nenhum dado importante disso.

Ambos os métodos são seguros, pois atacam a raiz do problema, acabando com o vazamento de informação gerado pelos erros de acesso à cache de forma que o atacande não obtenha nenhuma informação relevante. Além disso, os resultados dos testes realizados mostram que os métodos apresentados não tem um impacto grande na performance nem no custo, e são mais eficientes do que as soluções já existentes.

Em suma, as soluções apresentadas .alcançam a segurnaça necessária e possuem pouca influência na performance.