FACULDADE DE TECNOLOGIA DE AMERICANA

Curso Superior de Tecnologia em Segurança da Informação

Marcel Rosolen

TÉCNICAS DE ATAQUES À MEMÓRIA KERNEL: MELTDOWN E SPECTRE

Americana, SP
2018

FACULDADE DE TECNOLOGIA DE AMERICANA

Curso Superior de Tecnologia em Segurança da Informação

Marcel Rosolen

TÉCNICAS DE ATAQUES À MEMÓRIA KERNEL: MELTDOWN E SPECTRE

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Segurança da Informação, sob a orientação do Prof. Me. Benedito Luciano Antunes de França.

Área de concentração: Segurança em sistemas de informação voltada para Arquitetura de Computadores.

Americana, SP
2018

FACULDADE DE TECNOLOGIA DE AMERICANA

Curso Superior de Tecnologia em Segurança da Informação

FICHA CATALOGRÁFICA – Biblioteca Fatec Americana - CEETEPS

Dados Internacionais de Catalogação-na-fonte

R737t ROSOLEN, Marcel

Técnicas de ataques à memória kernel: meltdown e spectre. / Marcel Rosolen. – Americana, 2018.

47f.

Monografia (Curso de Tecnologia em Segurança da Informação) - - Faculdade de Tecnologia de Americana – Centro Estadual de Educação Tecnológica Paula Souza

Orientador: Prof. Ms. Benedito Luciano Antunes de França

1 Segurança em sistemas de informação I. FRANÇA, Benedito Luciano Antunes de II. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana

CDU: 681.518.5

FACULDADE DE TECNOLOGIA DE AMERICANA

Curso Superior de Tecnologia em Segurança da Informação

Marcel Rosolen

TÉCNICAS DE ATAQUE À MEMÓRIA KERNEL: MELTDOWN E SPECTRE

Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Segurança da Informação pelo CEETEPS/Faculdade de Tecnologia – FATEC/ Americana.

Área de concentração: Segurança em sistemas de informação voltada para Arquitetura de Computadores.

Americana, 9 de Agosto de 2018.

Banca Examinadora:

Prof. Me. Benedito Luciano Antunes de França (Presidente)
Fatec Americana

Prof. Me. Clerivaldo José Roccia
Fatec Americana

Prof. Esp. Antonio Alfredo Lacerda
Fatec Americana

AGRADECIMENTOS

Primeiramente, a Deus por me prover a cada dia o presente da vida e a dádiva do futuro.

À Instituição FATEC Americana, ao corpo docente, a direção, a administração e funcionários que, pela segunda vez, me acolheram e me proporcionaram estudos que me fizeram progredir não apenas como estudante de tecnologia, mas também como ser humano.

Ao meu orientador, o Prof. Me. Benedito Luciano Antunes de França, que me ajudou não apenas na elaboração deste trabalho como também no convívio acadêmico, por seus ensinamentos, sua paciência e incentivo em todos os nossos passos.

Aos meus pais, que são minha pedra fundamental, a minha base para tudo que construí e que estou construindo. São eles os responsáveis por todo o meu sucesso.

Aos meus familiares e minha namorada pelos inúmeros e especiais incentivos e colaborações em que se dispuseram em todos os passos da minha vida.

E a todos que direta ou indiretamente fizeram parte de minha formação.

DEDICATÓRIA

Aos meus pais, familiares e namorada. Também a todos que me ajudaram nos desafios e oportunidades da vida.

RESUMO

Este trabalho apresenta o detalhamento dos microprocessadores a níveis técnicos suficientes para o entendimento de como as vulnerabilidades Meltdown e Spectre funcionam. Não obstante, inclui um completo estudo com informações sobre as falhas, desde seus criadores, fatores técnicos e implicações, até a aplicação de soluções paliativas e permanentes para sua correção. Por fim, detalharemos uma série de testes realizados com o intuito de comprovar que a aplicação das mencionadas correções incorre em perda de processamento, uma vez que seu emprego reduz a capacidade dos microprocessadores.

Palavras-chave: Meltdown; Spectre; execução dinâmica; execução especulativa; previsão de ramificação múltipla.

ABSTRACT

This work presents the microprocessor detailing at sufficient technical levels for understanding how Meltdown and Spectre vulnerabilities behave. Nevertheless, it includes a complete study with information about the failures, from their creators, technical factors and implications, to the application of palliative and permanent solutions for their correction. Finally, we will detail a series of tests carried out in order to verify that the application of the mentioned corrections incurs loss of processing, since its use reduces the capacity of the microprocessors.

Keywords: Meltdown; Spectre; dynamic execution; speculative execution; multiple branch prediction.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE GRÁFICOS

INTRODUÇÃO

Microprocessadores são considerados o cérebro de um computador. Este componente é responsável por controlar todas as operações, interpretar e executar instruções, realizar cálculos, tomar decisões e interagir com os demais componentes que formam um sistema computacional. É, portanto, um dos componentes mais complexos e importantes em um computador. Os microprocessadores estão embutidos em praticamente todos equipamentos eletrônicos que necessitem de algum controle de hardware ou processamento de dados, estando contido em computadores, celulares, consoles de jogos (videogames), televisores, carros, maquinário industrial, satélites. Por tratarem todas as informações presentes em um computador, é necessário a aplicação de vários mecanismos de segurança que garantam que estes dados não sejam comprometidos, o que é realizado mediante vários métodos construídos em software e hardware. Em pesquisas recentes, descobriram-se novos métodos para burlar as defesas dos microprocessadores, a partir de falhas de sua infraestrutura.

Este trabalho foi estruturado como se segue: no primeiro capítulo, foi dado uma visão geral sobre os microprocessadores, uma breve história de sua composição técnica. No segundo capítulo foi desenvolvido a temática central, em que se explica como estas vulnerabilidades atuam. No terceiro capítulo se refere às soluções encontradas para mitigar ou corrigir as falhas identificadas. O quarto capítulo apresenta testes de desempenho com base nas atualizações de sistema e qual o seu impacto no poder de processamento dos microprocessadores. Por fim, na Conclusão, foram demonstradas as principais informações almejadas e obtidas graças ao conteúdo estudado.

CAPÍTULO 1 - MICROPROCESSADORES

A ideia do microprocessador, uma unidade central de processamento, surgiu durante a década de 40, em uma publicação de John Von Neumann, em que se propôs a criação do EDVAC (Electronic Discrete Variable Automatic Computer), o qual teve sua construção finalizada em 1949 (ARRUDA, 2011). Até então os computadores existentes não possuíam uma unidade de processamento, função realizada por um conjunto de módulos que, com combinações de chaves e cabos, realizavam os cálculos necessários. Até então os computadores utilizavam válvulas (primeira geração de computadores) e posteriormente transistores (segunda geração de computadores) para chavear os sinais elétricos. A evolução para a terceira geração de computadores se deu com a utilização de circuitos integrados, os quais substituíram os transistores convencionais por chips com transistores e demais componentes eletrônicos miniaturizados, com destaque para os computadores IBM System/360 (SOUZA *et al*., 2016). A quarta geração de computadores se inicia com a criação do primeiro microprocessador do mundo, quando em 15 de novembro de 1971, batizada de "Intel 4004", desenvolvido por Federico Faggin, Ted Hoff e Mazor Stanley pela Intel (MOREIRA, 2011), uma das maiores empresas de desenvolvimento e produção de circuitos integrados. Possuía 2300 transistores em um espaço de 10 mícrons (10000 nanômetros) (INTEL, 2018a).

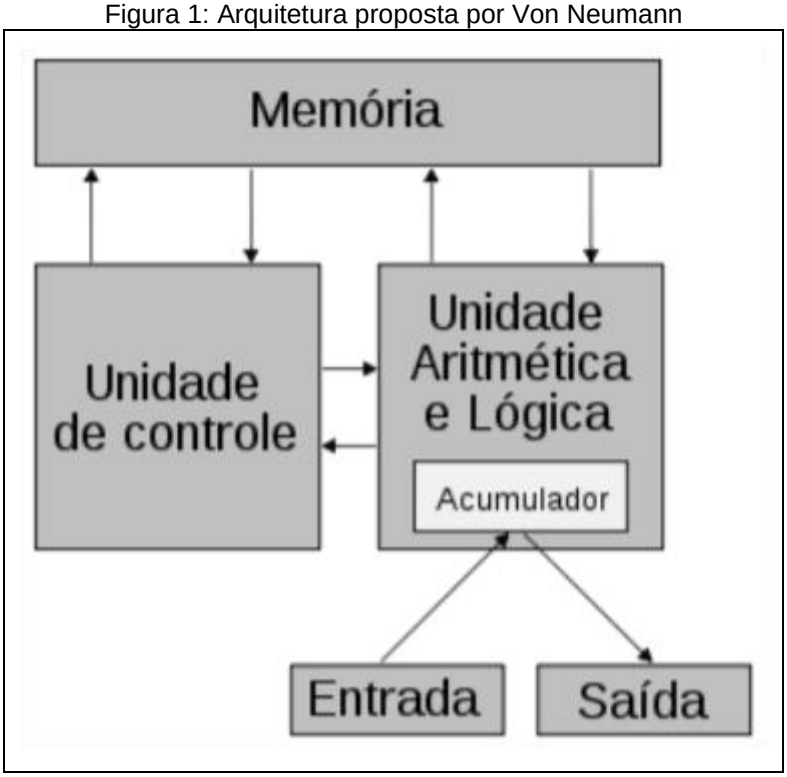
Em outubro de 1985 foi lançado o Intel 386, que estabelecia a arquitetura de 32 bits, base utilizada até os dias de hoje (MORIMOTO, 2010). Considerado o primeiro processador moderno pelo fato de conter um conjunto básico de instruções, também utilizado até os dias de hoje.

1.1. Composição técnica de um microprocessador

Para seu funcionamento, um microprocessador utiliza vários módulos integrados, cada um com suas funções específicas. Estes módulos foram propostos pela teoria de Von Neumann em 1945 (UHLMANN, 2014), o qual descrevia um

computador com uma Unidade Lógica e Aritmética (ULA), Unidade de Controle (UC), Memória, Unidades de Entrada e Saída (E/S) e Registradores (GATTO, 2016).

Figura 1: Arquitetura proposta por Von Neumann

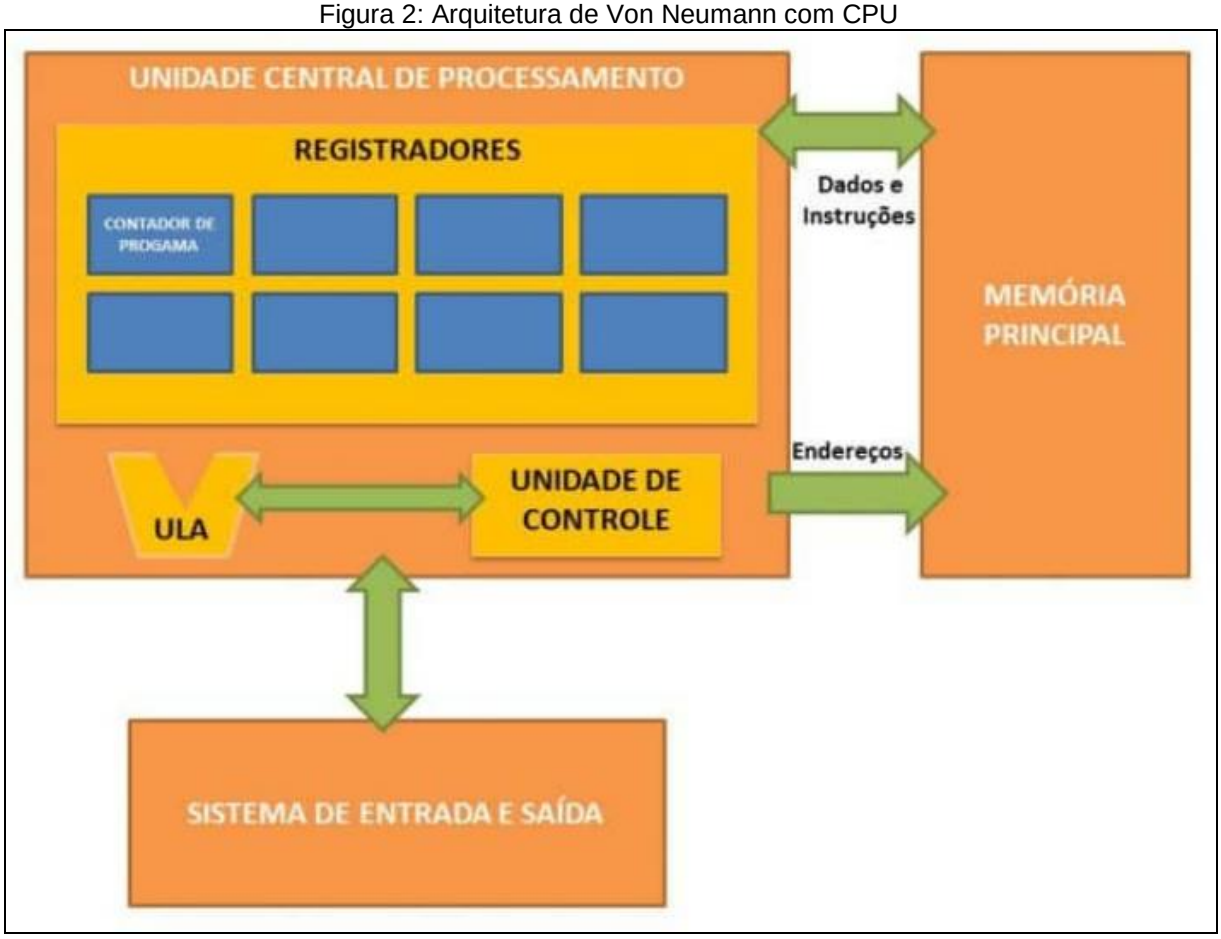


Fonte: Uhlmann, 2014, p. 6.

O microprocessador, denominado Unidade Central de Processamento (CPU - Central Processing Unit), é formado pelas unidades ULA, UC e registradores, como pode ser visto na Figura 2.

A respeito da Unidade Lógica e Aritmética, é concebida como uma unidade responsável por efetuar as operações lógicas e aritméticas. Ao passo que a Unidade de Controle é uma unidade responsável pelo controle do funcionamento do microprocessador, dos dados transitados, comunicação entre a CPU e a memória principal e unidades de E/S do computador. Na UC está inserida a programação em linguagem de máquina responsável pelo funcionamento da CPU. Os Registradores são memórias utilizadas exclusivamente pelo microprocessador para a movimentação de dados que serão necessários para o processamento solicitado. Trabalham na mesma velocidade do processador, garantindo eficiência de processamento, que não seria possível com a utilização das memórias do computador, que possuem acesso mais lento.

Figura 2: Arquitetura de Von Neumann com CPU



Fonte: Gatto, 2016

1.2. Técnicas de processamento

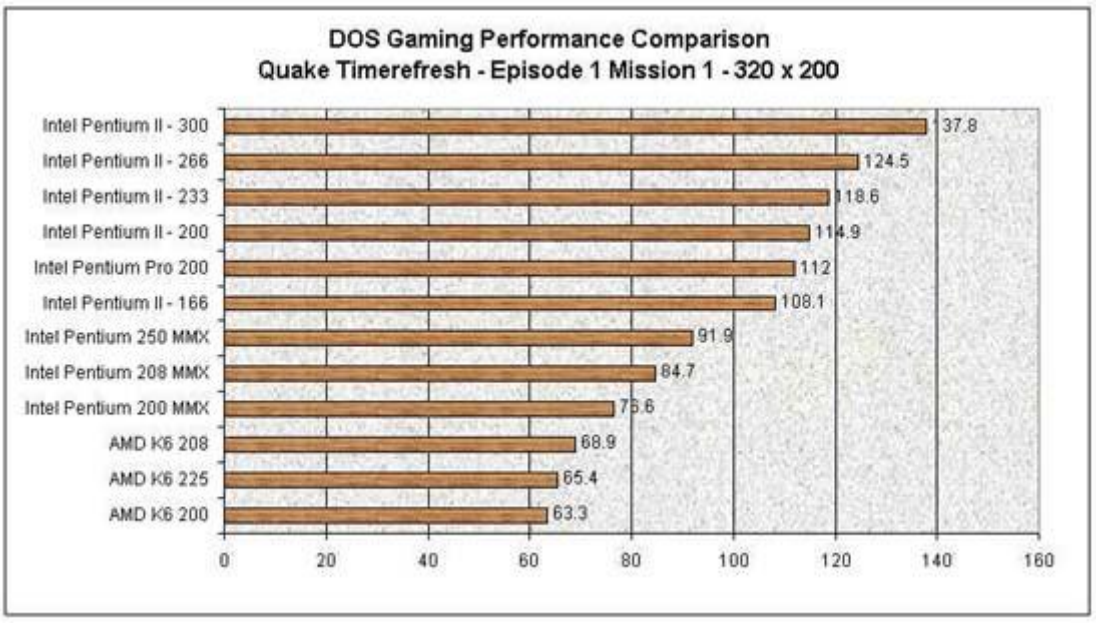
Um microprocessador utiliza várias técnicas de processamento de dados, visando uma melhoria no desempenho geral do conjunto, seja por processar maiores quantidades de dados por vez ou realizar cálculos de forma mais rápida. Entre as técnicas instrumentalizadas, podemos destacar o uso de Execução Especulativa (Speculative Execution), Previsão de Ramificação Múltipla (Multiple Branch Prediction) e Análise de Fluxo de Dados (Dataflow Analysis) (HRUSKA, 2018).

A Execução Especulativa é um conjunto de técnicas utilizadas pelo microprocessador para antecipar a execução de instruções que serão solicitadas posteriormente em uma fila de instruções, assim, quando uma determinada instrução for solicitada, o resultado da mesma já estará disponível, poupando assim

tempo de processamento. Esta técnica é utilizada porque, por vezes, algumas unidades de processamento do microprocessador estão ociosas, ou ainda porque alguma instrução pode levar a diferentes caminhos. O uso destas técnicas conferiu um maior poder de processamento comparado com a geração anterior de microprocessadores (HRUSKA, 2018).

No gráfico 1, nota-se como a família de microprocessadores Intel Pentium Pro e Intel Pentium II produziu melhores resultados em *benchmarks*, comparados a outros microprocessadores, que faziam uso de técnicas de processamento em linha, isto é, fazia uso de uma sequência de instruções que era executada, conforme a fila em que se encontravam, sem saltos ou predições.

Gráfico 1: Comparação de desempenho entre microprocessadores



Fonte: Hruska, 2018

Estas formas de atuação do microprocessador garantiram um salto no poder de processamento, porém, mantiveram guardadas falhas críticas descobertas recentemente, as vulnerabilidades *Meltdown* e *Spectre*, que serão abordadas a seguir.

CAPÍTULO 2 - MELTDOWN E SPECTRE

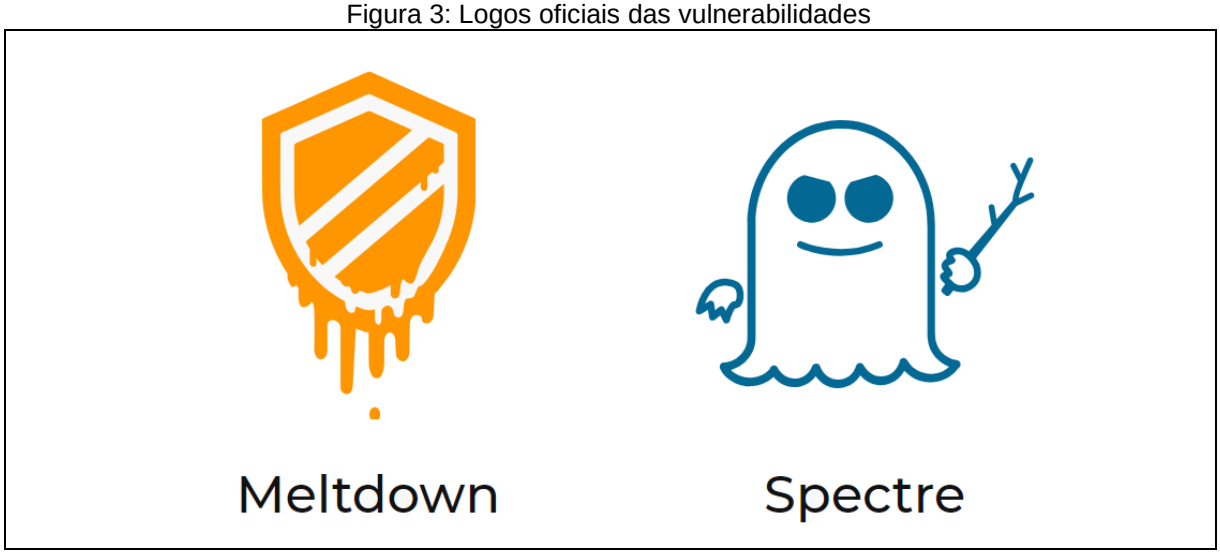
Meltdown e *Spectre* são os nomes dados a três vulnerabilidades identificadas nos microprocessadores modernos que utilizam técnicas de processamento denominadas Processamento Dinâmico, ou seja, que utilizam Execução Especulativa, Predição de Ramificação Múltipla e Análise de Fluxo de Dados. Ganharam grande notoriedade devido ao fato de estarem presentes em praticamente todos os microprocessadores utilizados na atualidade, desenvolvidos desde 1995 até os dias de hoje.

As vulnerabilidades foram descobertas por pesquisadores do Projeto Google Zero, Universidade de Graz (Áustria), Universidade da Pennsylvania (EUA), Universidade de Maryland (EUA), Universidade de Adelaide (Austrália), empresas privadas de segurança em tecnologia Cyberus Technology e Rambus (GRAZ UNIVERSITY OF TECHNOLOGY, 2018). Após a descoberta, um *website* foi criado pelos pesquisadores com a função de reunir toda a informação recolhida sobre o tema, tais como as imagens, endereços de repositórios e artigos técnicos, sob o nome de meltdownattack.com. As descobertas foram realizadas em 2017 e reportadas em 3 de janeiro de 2018, descritas a seguir:

- Meltdown:
  - Rogue Data Cache Load, registrado na CVE-2017-5754 (WILLIAMS, 2018);

- Spectre:
  - Bounds Check Bypass, registrado na CVE-2017-5753; e
  - Branch Target Injection, registrado na CVE-2017-5715 (WILLIAMS, 2018);

Figura 3: Logos oficiais das vulnerabilidades



Fonte: Graz University of Technology, 2018

A CVE (Common Vulnerabilities and Exposures) é uma base internacional de dados de acesso público, destinado ao registro de todas as falhas de segurança, disponível em https://cve.mitre.org/. Cada falha registrada possui um identificador, uma descrição e referências públicas (MITRE, 1999-2018).

De acordo com a CVE, a vulnerabilidade CVE-2017-5754, conhecida como *Meltdown*, refere-se a:

> Sistemas com microprocessadores que utilizam execução especulativa e previsão indireta de ramificação podem permitir a divulgação não autorizada de informações a um invasor com acesso de usuário local por meio de uma análise de canal lateral do cache de dados (MITRE, 1999-2018)[1].

Vulnerabilidades CVE-2017-5753 e CVE-2017-5715, conhecidas como *Spectre*, referem-se a:

> Sistemas com microprocessadores utilizando execução especulativa e previsão de ramificação podem permitir a divulgação não autorizada de informações a um atacante com acesso de usuário local por meio de uma análise de canal lateral (MITRE, 1999-2018)[2].

[1] Systems with microprocessors utilizing speculative execution and indirect branch prediction may allow unauthorized disclosure of information to an attacker with local user access via a side-channel analysis of the data cache (MITRE, 1999-2018. *Tradução nossa!)*

[2] Systems with microprocessors utilizing speculative execution and branch prediction may allow unauthorized disclosure of information to an attacker with local user access via a side-channel analysis (MITRE, 1999-2018. *Tradução nossa!*).

2.1- Meltdown

A vulnerabilidade identificada como *Meltdown* consiste em uma brecha da arquitetura dos microprocessadores, ou seja, é uma falha de hardware. Esta falha permite o vazamento do conteúdo da memória *kernel*, uma área extremamente protegida em um computador, principalmente por conter dados sensíveis. É considerada uma vulnerabilidade fácil de explorar, assim como também é fácil a correção paliativa.

> *Meltdown* quebra o isolamento mais fundamental entre os aplicativos do usuário e o sistema operacional. Este ataque permite que um programa acesse a memória e, portanto, também os segredos de outros programas e o sistema operacional (GRAZ UNIVERSITY OF TECHNOLOGY, 2018)[3].

Esta vulnerabilidade foi descoberta e reportada por três equipes de pesquisadores independentes, os quais integram a autoria do artigo técnico sobre a Meltdown, sendo eles: Moritz Lipp, Michael Schwarz, Daniel Gruss, Stefan Mangard, da Graz University of Technology; Thomas Prescher e Werner Haas, da Cyberus Technology GmbH; Paul Kocher[3], pesquisador independente; Daniel Genkin, da University of Pennsylvania and University of Maryland; Yuval Yarom, University of Adelaide e Data61, e, por último, Mike Hamburg, da Rambus, Cryptography Research Division (LIPP *et al.,* 2018).

De acordo com os pesquisadores citados, apenas os microprocessadores com arquitetura Intel, a partir de 2010, são afetados por esta vulnerabilidade, e potencialmente pode afetar os demais fornecedores (AMD, ARM, etc).

Basicamente, a vulnerabilidade permite com que um código recupere dados da memória *kernel* e a carregue nos registradores do microprocessador para posterior visualização. Como existem barreiras de segurança que impedem tal ação, por se tratar de memória protegida, é utilizado o processamento dinâmico, ou seja, a execução especulativa, para burlar as isolações criadas pelo microprocessador e

[3] Meltdown breaks the most fundamental isolation between user applications and the operating system. This attack allows a program to access the memory, and thus also the secrets, of other programs and the operating system (GRAZ UNIVERSITY OF TECHNOLOGY, 2018. *Tradução nossa!).*

sistema operacional. Com os dados salvos no cache do microprocessador, outro código é capaz de realizar sua leitura, sem qualquer verificação de segurança.

> *Meltdown* (...) baseia-se na observação de que, quando uma instrução causa uma *exception*, as instruções a seguir são executadas fora de ordem antes de serem finalizadas. (...) O *Meltdown* explora uma vulnerabilidade específica para muitos processadores Intel e alguns processadores ARM, o que permite que certas instruções executadas especulativamente ignorem a proteção da memória. Combinando esses problemas, o *Meltdown* acessa a memória *kernel* a partir do espaço do usuário. Esse acesso causa uma desvio, mas antes que o desvio seja emitido, as instruções que seguem o acesso vazam o conteúdo da memória acessada por meio de um canal encoberto do cache (KOCHER, *et al*., 2018)[4].

A vulnerabilidade pode ser explorada seguindo três passos:

Passo 1: o conteúdo de um endereço de memória protegida é carregado em um registrador;

Passo 2: uma instrução transitória acessa o conteúdo armazenado em cache;

Passo 3: utilizando uma técnica de ataque denominada *Flush+Reload* para determinar qual espaço em cache foi acessado, e onde a informação protegida se encontra.

Ao executar estes passos em diferentes endereços de memória, pode-se extrair a memória *kernel*, incluindo toda a memória física.

O primeiro passo para a realização deste ataque consiste em carregar os dados da memória principal para os registradores do microprocessador. Neste processo, o microprocessador realiza traduções de endereços virtuais para endereços físicos e também verificações de permissão de acesso para os endereços solicitados, uma das formas de se proteger o conteúdo da *kernel*. Os sistemas operacionais atuais mapeiam toda a memória *kernel* no espaço de endereçamento virtual de todos os processos de usuários, assim quando um processo necessita realizar transações que necessitam de funções da *kernel*, a mesma já estará

[4] Meltdown (...) relies on the observation that when an instruction causes a trap, following instructions are executed out-of order before being terminated. (...) Meltdown exploits a vulnerability specific to many Intel and some ARM processors which allows certain speculatively executed instructions to bypass memory protection. Combining these issues, Meltdown accesses kernel memory from user space. This access causes a trap, but before the trap is issued, the instructions that follow the access leak the contents of the accessed memory through a cache covert channel (KOCHER, *et al*., 2018. *Tradução nossa!).*

disponível para acesso. Como o acesso a estes endereços é protegido, como já mencionado, qualquer tentativa de acesso irá disparar uma *exception* (exceção), uma vez que não existe o nível de permissão requerido para realizar tal leitura. Uma *exception* é um desvio na linha de comandos, uma vez identificada alguma falha, como por exemplo um chamado a um endereço inválido, uma divisão por zero ou um acesso indevido à memória. A vulnerabilidade explora a execução dinâmica do microprocessador, a qual é realizada no mínimo espaço de tempo entre o acesso indevido à memória protegida e a execução da *exception*. Em um código, quando se solicita acesso à memória *kernel* em uma linha de programação, as linhas posteriores já foram lidas e executadas pelo microprocessador, como parte da execução dinâmica que não aguarda o fim da execução de uma linha para prosseguir para a linha seguinte.

No exemplo a seguir, dado pelos pesquisadores, vemos como o acesso à memória *kernel* é realizada na linha 4, porém as demais linhas podem ser executadas 5-7 antes mesmo da *exception* ser disparada.

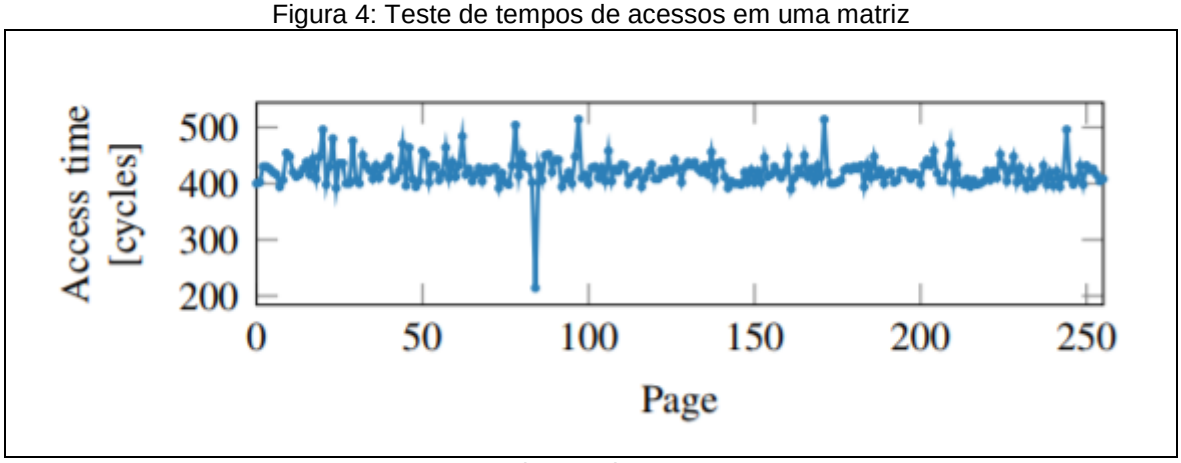
```
1 ; rcx = kernel address
2 ; rbx = probe array
3 retry:
4 mov al, byte [rcx]
5 shl rax, 0xc
6 jz retry
7 mov rbx, qword [rbx + rax] (LIPP et al., 2018).
```

O segundo passo usa os comandos executados de forma dinâmica pelo microprocessador, chamados instruções transitórias, as quais realizam o processamento utilizando dados da memória protegida, como podem ser vistas nas linhas 5-7 do exemplo acima. São estas instruções responsáveis por transmitir os dados protegidos para o atacante. Os dados adquiridos são então transportados para uma matriz alocada na memória.

O terceiro passo recebe os dados transmitidos, através de ataques como o *Flush+Reload* apontado para a matriz criada em memória. Ao fazer uso deste ataque, procura se determinar qual o tempo de acesso de cada uma das páginas da matriz; a página que apresentar tempo diferenciado das demais será a página que corresponderá diretamente ao valor vazado pelo segundo passo.

Um exemplo de *Flush+Reload* é visto a seguir, realizado pelos pesquisadores, onde a página com valor = 84 possui tempo de acesso diferente das demais páginas, evidenciando que neste local se encontra dados no cache do microprocessador.

Figura 4: Teste de tempos de acessos em uma matriz



Fonte: Lipp *et al.,* 2018, p. 5

2.2- Spectre

As vulnerabilidades identificadas como *Spectre* buscam se aproveitar das falhas introduzidas pela computação dinâmica, tal qual acontece com *Meltdown*, em que a falha se encontra na arquitetura dos microprocessadores. O foco dos ataques *Spectre* é o de acessar informações da área de endereçamento de memória do usuário entre diferentes aplicativos, o que a princípio não é possível devido bloqueios de segurança do sistema operacional. Esta vulnerabilidade é considerada mais complexa, tanto para sua exploração, como para sua correção, em comparação com a *Meltdown*.

> *Spectre* quebra o isolamento entre diferentes aplicativos. Permite a um atacante enganar programas que seguem boas práticas, a vazar seus segredos. De fato, as verificações de segurança propostas pelas boas práticas aumentam a superfície de ataque e podem tornar os aplicativos mais susceptíveis ao *Spectre* (GRAZ UNIVERSITY OF TECHNOLOGY, 2018)[5].

[5] Spectre breaks the isolation between different applications. It allows an attacker to trick error-free programs, which follow best practices, into leaking their secrets. In fact, the safety checks of said best practices actually increase the attack surface and may make applications more susceptible to Spectre (GRAZ UNIVERSITY OF TECHNOLOGY, 2018. *Tradução nossa*!).

Estas vulnerabilidades foram descobertas e reportadas pelas seguintes equipes de pesquisadores independentes, os quais integram a autoria do artigo técnico sobre a *Spectre*, sendo eles: Paul Kocher, pesquisador independente; Jann Horn, do Google Project Zero; Anders Fogh, da G DATA Advanced Analytics; Daniel Genkin, da University of Pennsylvania and University of Maryland; Daniel Gruss, Moritz Lipp, Stefan Mangard e Michael Schwarz, ambos da Graz University of Technology; Werner Haas e Thomas Prescher, da Cyberus Technology; Mike Hamburg, da Rambus, Cryptography Research Division; Yuval Yarom, da University of Adelaide e Data61 (KOCHER *et al.,* 2018). Elas estão presentes nos microprocessadores Intel, AMD e ARM, tendo em vista que testes foram realizados com sucesso, em contraste com a *Meltdown*, em que apenas microprocessadores Intel e algumas opções de microprocessadores ARM foram afetadas.

Basicamente, um ataque se inicia com a fase de configuração, em que o atacante realiza operações que confundem o microprocessador a explorar, posteriormente, uma execução especulativa incorreta, inclusive, induzindo a execução especulativa, como por exemplo manipular os estados dos caches de dados. Durante a fase inicial se prepara o canal de transmissão, em que se realiza a limpeza dos caches. Por seguinte, são executadas instruções utilizando a execução dinâmica, que transferem as informações confidenciais de um aplicativo, a partir das ações do usuário vítima, ou ainda pela execução especulativa de código malicioso. Ao final, os dados são extraídos utilizando ataques de *timing*, tal qual acontece com *Meltdown*. É realizado um ataque *Flush+Reload* ou *Evict+Reload*, que analisa os tempos de acesso às linhas de cache monitoradas.

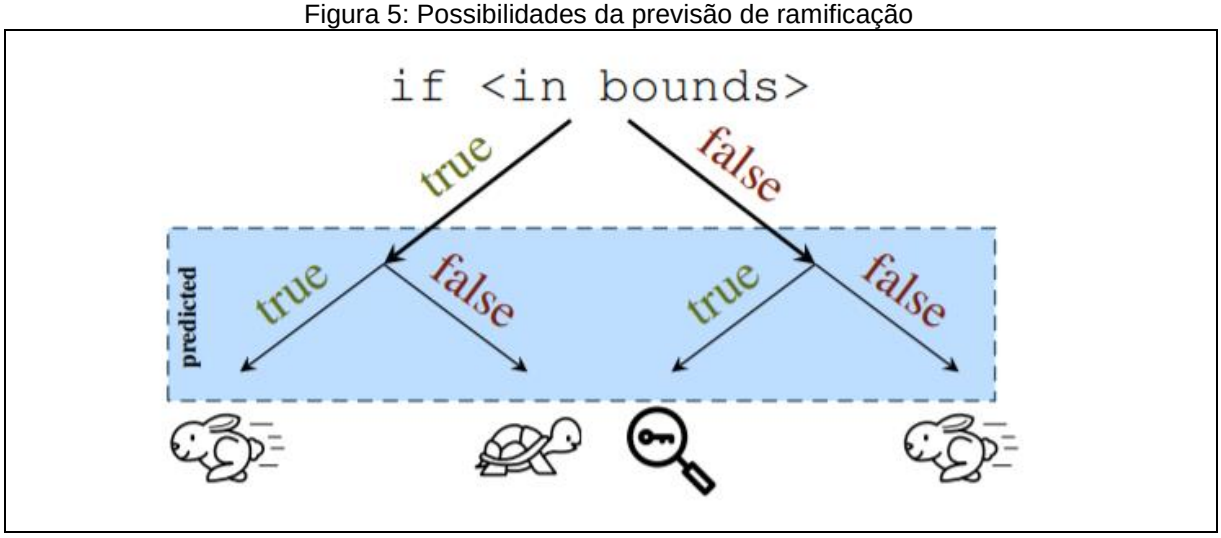
Devido ao *Spectre* ser a descoberta de duas vulnerabilidades, se tem duas variantes para um mesmo ataque, a Exploração de Erro de Previsão de Ramificação Condicional, o qual se refere a *Bounds Check Bypass*, e o Envenenamento de Ramificação Indireta, o qual se refere a *Branch Target Injection*. A primeira variante toma como exemplo o seguinte trecho de código:

```
if (x < array1_size)
y = array2[array1[x] * 4096];
```
(KOCHER *et al.,* 2018).

Este código, pertencendo a uma aplicação legítima, recebe o valor inteiro *x* de uma fonte não confiável. A primeira atuação do microprocessador é a verificação, por motivos de segurança, dos limites entre a aplicação e a fonte do inteiro *x*. Uma verificação na qual *x* está fora dos limites (não pertence à aplicação) pode retornar uma *exception*, o que causa o encerramento da execução. Porém, antes mesmo do microprocessador verificar esta dependência de limites para a execução do *if*, a execução dinâmica, no caso, a Predição de Ramificação Múltipla, calcula a possibilidade dos limites entre as duas partes, e realiza ou não a execução de seus códigos subsequentes, uma vez que não há dependências, pois já se conhece o resultado de *if*, assim como o valor de *x* e os endereços das *arrays* também são conhecidos, ao passo que, ao realizar a verificação de limites, ou seja, se o x é válido, pode possuir diversas dependências, até sua efetiva ação.

Supondo que a aplicação receba o valor de *x* do atacante (portanto, fora dos limites da aplicação), faz com que o código *array1[x]* aponte para um byte secreto *k*. Em operações anteriores, a aplicação recebe valores válidos para *x*, que levam a previsão de ramificação a acreditar que as verificações de limites deverão retornar o resultado positivo sempre para esta variável. Este ataque é exemplificado na Figura 5.

Figura 5: Possibilidades da previsão de ramificação



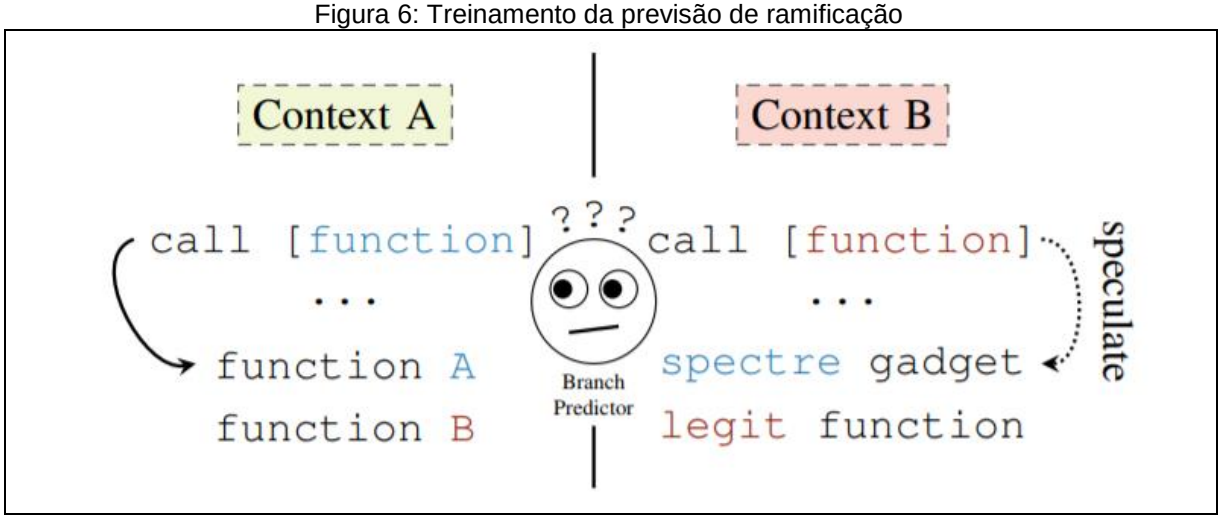
Fonte: Kocher *et. al.*, 2018, p. 6

Após a computação dos códigos, em que *array1[x]* retorna *k*, e *array2[k * 4096]* são executados, o resultado da verificação de limites já é conhecido, e, ao

determinar que o valor *x* se encontra fora dos limites, toda a execução dinâmica previamente executada é retornada ao estado inicial, porém, *array2* permanece em cache. Por fim, se calcula a localização em cache desta *array*, utilizando ataques como a *Flush+Reload*, e então se recupera o valor de *k*.

A segunda variante tende a enganar a previsão de ramificação a executar determinado ramo em todas as ocorrências. Isto se dá pelo 'treinamento' deste componente a sempre pegar o mesmo caminho, como pode ser observado no exemplo da Figura 6.

Figura 6: Treinamento da previsão de ramificação



Fonte: Kocher *et. al.*, 2018, p. 6

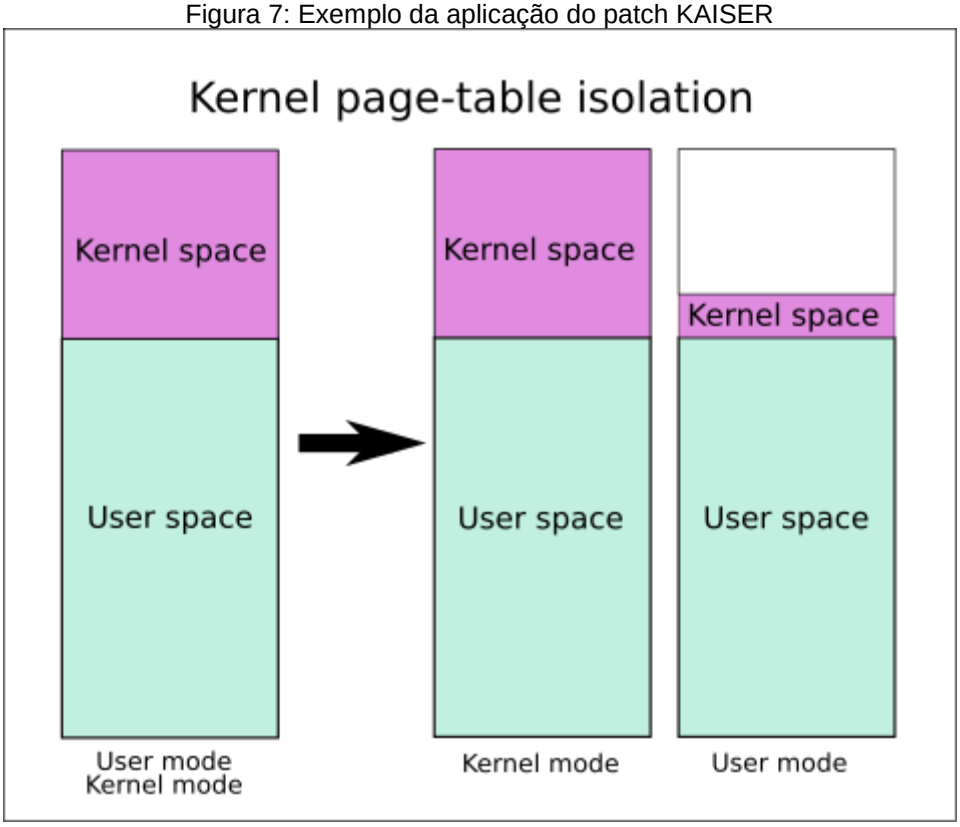
Mais especificamente, em um contexto A, o atacante mostra ao previsor de ramificação determinado caminho. Ao se verificar em um contexto B, que contém código malicioso, o previsor de ramificação simplesmente segue a lógica na qual acredita ser a correta, por haver histórico desta tomada de decisão. Seguindo as mesmas premissas dos outros ataques, a execução dinâmica do código malicioso deverá carregar as informações da memória em registradores e daí para o cache até ser recuperado pelo atacante.

CAPÍTULO 3 - MITIGAÇÕES E SOLUÇÕES

Uma vez que as vulnerabilidades identificadas são providas por falhas de arquitetura, ou seja, falhas de hardware, é muito difícil uma correção permanente para todos os microprocessadores afetados. As correções em forma física só podem acontecer na troca de microprocessadores, e um *recall* seria inviável devido a quantidade de microprocessadores em utilização em todo o planeta. O CEO da Intel, Brian Krzanich, também confirmou que não será realizado nenhum *recall*, uma vez que soluções via software estão sendo produzidas e podem eliminar os riscos que estas vulnerabilidades podem trazer (SHANKLAND; HAUTALA, 2018). O que pode ser esperado é que gerações futuras de microprocessadores sejam desenvolvidas com arquiteturas completamente diferentes das atuais, prevenindo as vulnerabilidades estudadas, porém, uma mudança deste tamanho pode levar anos de pesquisa e desenvolvimento. Entre outras possibilidades, algumas soluções em forma de software, como descreveremos abaixo, ou a troca integral do processador, poderiam se tornar as principais medidas para evitar que se explorem estas vulnerabilidades.

Segundo os pesquisadores, o *patch* KAISER (Kernel Address Isolation to have Side-channels Efficiently Removed) nos sistemas operacionais Linux é suficiente para impedir a exploração das vulnerabilidades, pois esta medida implementa maiores e melhores isolações entre memória *kernel* e espaço do usuário (LIPP *et. al.*, 2018, p. 11; 14). Esta solução impede que a memória *kernel* seja mapeada no espaço do usuário, exceto por algumas pequenas áreas necessárias para a execução de suas aplicações, como por exemplo os manipuladores de interrupções. Este *patch* estará disponível para atualização sob o nome KPTI (Kernel Page-Table Isolation). *Patches* similares serão introduzidos para Microsoft Windows 10. MacOS e IOs possuem recursos similares (LIPP *et. al.*, 2018, p. 14). A figura 7 ilustra como a isolação do *kernel* funciona, de um ponto de vista do modo *kernel* e modo usuário.

Figura 7: Exemplo da aplicação do patch KAISER



Fonte: Wikipédia, 2018

Nos sistemas operacionais Microsoft Windows foram lançados diversas *patches* de correções nos dias subsequentes a divulgação das vulnerabilidades, como pode se verificar na Tabela 1. Estes *patches* abrangem as três vulnerabilidades.

Tabela 1: Atualizações para o sistema operacional Microsoft Windows

Versão de Sistema Operacional	**Atualização**
Windows Server, version 1709 (Server Core Installation)	KB 4056892
Windows Server 2016	KB 4056890
Windows Server 2012 R2	KB 4056898
Windows Server 2012	KB 4088880
Windows Server 2008 R2	KB 4056897
Windows Server 2008	KB 4056897
Windows 10 (RTM, 1511, 1607, 1703, 1709), Windows 8.1, Windows 7 SP1	ADV180002 (Várias atualizações)

Fonte: Autoria própria baseada em Cimpanu (2018) e Microsoft (2018) (Operating system version)

Por fim, os aplicativos também disponibilizaram atualizações para proteger seus usuários, para que não sejam usados em um ataque ou que tenham seus dados acessados. É o caso dos *web browsers*, que são uma das principais fontes de um possível ataque, por meio de um *web site* malicioso; e ainda por possuírem dados sigilosos, como por exemplo listas de senhas e informações pessoais. Entre

os principais aplicativos da categoria, o Google Chrome e o Mozilla Firefox implementaram medidas que protegem o aplicativo e os dados presentes dos supostos ataques (ABRAMS, 2018). Demais empresas e organizações que desenvolvem produtos que, de alguma forma, foram afetados pelas vulnerabilidades, também possuem avisos e *patches* de atualização publicados.

CAPÍTULO 4 - IMPACTO NO PROCESSAMENTO DE DADOS APÓS ATUALIZAÇÕES

Após as atualizações de *software* e *firmware* disponibilizadas pelos fabricantes e desenvolvedores, ficou no ar a dúvida sobre qual impacto que estas mudanças trariam no poder de processamento destas unidades. Diversos testes foram realizados por empresas e por pesquisadores independentes, procurando mostrar em diversos produtos as variações de performance entre eles.

4.1- Testes produzidos pela Intel

Poucos dias após o início das atualizações, a Intel disponibilizou um estudo inicial que indicou não esperar impactos significativos para a maioria dos usuários, ou seja, para computadores pessoais e empresariais, para tarefas corriqueiras, tais como leitura e edição de texto ou o uso de aplicativos básicos. Em testes utilizando um microprocessador Intel de oitava geração, disco rígido de estado sólido (SSD - Solid State Drive) e o aplicativo de *benchmark* SYSMark, houve uma perda de 6% ou menos, com testes variando entre 2% a 14% (INTEL, 2018c). Ainda, de acordo com testes de parceiros industriais, houve pouco a nenhum impacto na performance (INTEL, 2018b).

No dia seguinte ao do anúncio deste estudo inicial, a Intel divulgou um estudo mais aprofundado, utilizando várias configurações de microprocessadores e aplicativos de *benchmark* (SHENOY, 2018). No geral, processadores mais antigos recebem um maior impacto comparado a seus sucessores. Para microprocessadores Intel de oitava geração o impacto esperado continua em torno de 6%, para aplicações web com operações complexas em Java Script o impacto esperado era de 10%. Aplicações intensivas com gráficos, como por exemplo jogos, ou processamento de dados, como por exemplo análises financeiras, possuem impacto mínimo. Para microprocessadores Intel de sétima geração o impacto é similar, com média de 7%, e para microprocessadores Intel de sexta geração, 8%. O resultado consolidado pode ser verificado na Tabela 2. Resultados pormenorizados estão disponíveis no Anexo A.

Tabela 2: Resultado consolidado de testes realizados pela Intel.

	Carga de Trabalho	**Intel® Core™ i7-8700K**	**Intel® Core™ i7-8650U**	**Intel® Core™ i7 7920HQ**	**Intel® Core™ i7 6700K**
Performance Relativa (Sistemas totalmente corrigidos / Sistemas não corrigidos a 100%)					
SYSmark 2014 SE - Geral	Escritório	94%	95%	93%	92%
PCMark 10 - Geral	Criação	96%	96%	97%	96%
3DMark Sky Diver - Geral	Jogos	100%	99%	100%	101%

Fonte: Autoria própria baseada em Shenoy (2018)

Apesar dos testes indicarem um impacto mínimo, com algumas raras variações de utilização que podem variar entre 5% até 30% no desempenho, especulam-se, como veremos, que microprocessadores mais antigos (quinta geração e anteriores) sofreriam mais com as atualizações propostas. Testes independentes, após a disponibilização de *patches* para as três vulnerabilidades, foram realizados por diferentes publicações especializadas em tecnologia. Nos *benchmarkings* verificados, a grande maioria admitiu perda de processamento na ordem de 1% a 4%, alguns poucos testes ultrapassaram 5% ou mais de impacto, e ainda alguns testes mostraram -2% a 0%. Houveram ainda alguns testes específicos que alcançaram 10% a 48%. A seguir serão demonstrados os testes realizados por duas publicações independentes, as quais apontam os dados informados anteriormente, sendo elas produzidas pela Phoronix (Item 4.2) e pela Techspot (Item 4.3).

4.2- Testes produzidos pela Phoronix

O Portal Phoronix, especializado em *benchmarking* em ambientes Linux, realizou diversos testes aos dias subsequentes à disponibilização dos *patches* de correção, utilizando aplicativo de *benchmark* próprio, como verificado a seguir. Entre os vários testes realizados, destacam-se os 1) testes com a inclusão dos *patchs* para *Meltdown*, 2) testes com as correções para *Spectre*, 3) testes em ambientes virtualizados, 4) testes com todas as aplicações corretivas e 5) com *hardware* de gerações anteriores. Na análise dos *benchmarkings*, pode-se verificar que os testes que requerem muita utilização de E/S são os mais afetados pelas atualizações, em que se explicam perdas de 30%, conforme Figura 12. Para atividades com foco no espaço do usuário, houveram mínimas alterações de performance. No geral, cargas

de trabalho que não se envolvem muito com a utilização do *kernel*, possuem perda de performance reduzida em comparação com cargas de trabalho que as utilizam mais. A utilização de computadores com *hardware* mais antigos mostrou ainda maior perda de performance. Para jogos e utilização gráfica, praticamente não houve perda (*Cf.* LARABEL, 2018, p. 1-6).

Entre as diversas baterias de testes desenvolvidas e publicadas pela Phoronix, escolheu-se para ilustrar esta pesquisa, os resultados divulgados em 11 de janeiro de 2018, pelo motivo destes testes utilizarem medidas corretivas para todas as vulnerabilidades (testes 4), e também por utilizar microprocessadores de gerações passadas (testes 5), a denominar:

ThinkPad T61 utilizando CPU Intel® Core™2 Duo T9300 (2008);
ThinkPad W510 utilizando CPU Intel® Core™ i7-720QM (2009); e
ThinkPad X1 Carbon utilizando CPU Intel® Core™ i7-5600U (2015).

A título de informação, os dados concernentes às Figuras 8, 9, 10, 11, 12, 13, 14 e 15 foram detalhadas neste espaço, visto que a inserção das explicações logo abaixo de cada uma das Figuras faria com que o espaço gráfico das laudas não suportasse a quantia de duas Figuras por lauda.

Nos testes com a utilização de SQLite, como demonstrado na Figura 8, verificam-se que os microprocessadores mais antigos tiveram baixa na performance na ordem de 3% a 10%, em comparação com o mesmo equipamento sem a adoção das medidas corretivas. Em comparação, pode-se verificar na Figura 9, que o mesmo teste, com microprocessador Intel® Core™ i7-4790K, obteve perda máxima de 2%.

Em testes de tempo de compilação de *kernel*, no entanto, todos os resultados apontaram perda de 2% a 4%, conforme constatado na Figura 10, inclusive com a aplicação ao processador atual, na Figura 11.

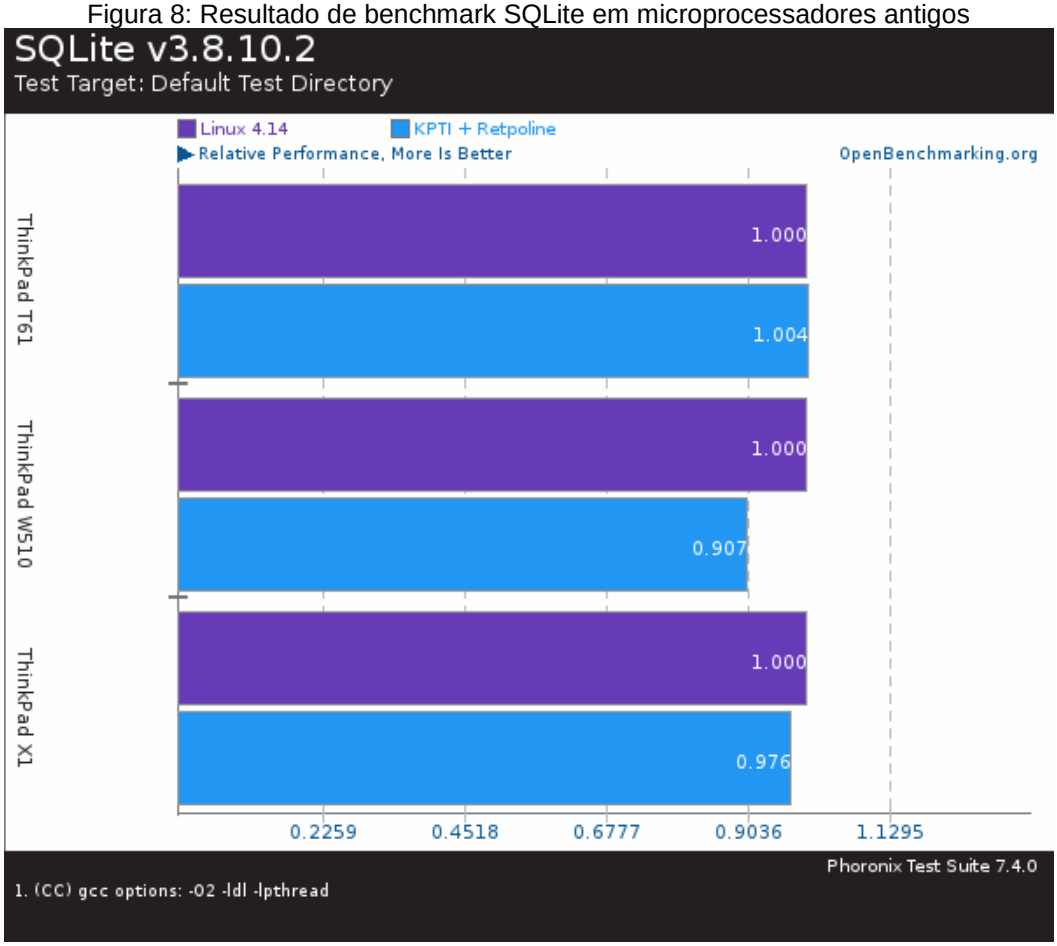
Em um dos testes mais díspares, o *benchmark* Redis v3.0.1, nos testes GET e SET, mostrou que a aplicação das correções promoveu 7% a 38% de perda de

desempenho em processadores antigos, conforme pode ser visto na Figura 12. No processador atual, não houve perda, conforme explicitado na Figura 13.

Em outro teste, que simula o serviço de servidor de páginas *web*, outra vez os microprocessadores antigos foram mais afetados (Figura 14) do que o microprocessador atual (Figura 15), com média de 15%.

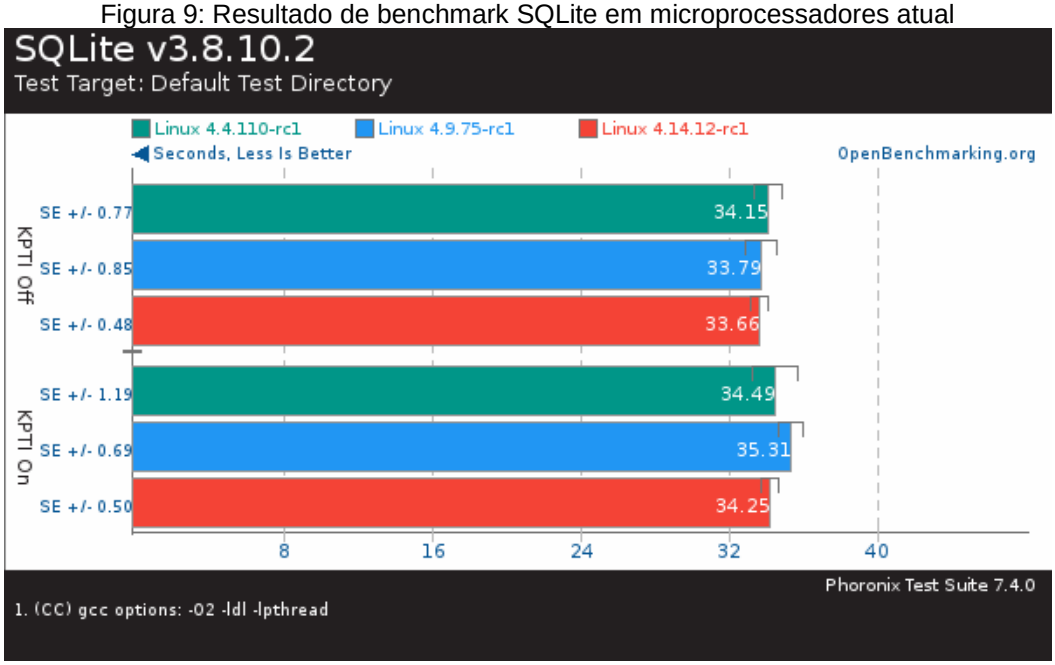
Salientam-se que os testes desenvolvidos pelo pesquisador independente Michael Larabel, pela Phoronix, desenvolvidos neste corrente ano, fundamentaram-se em uma máquina, com processador atual, com as distribuições Linux versões 4.14, 4.4 e 4.9 (vide Referências, artigo postado em 4 de janeiro), ao passo que os demais Testes, Larabel fez uso de três processadores antigos (notebooks com configurações diferentes), conforme já explicado anteriormente, usando distribuição Linux versão 4.14 (vide Referências, artigo postado em 11 de janeiro).

Figura 8: Resultado de benchmark SQLite em microprocessadores antigos



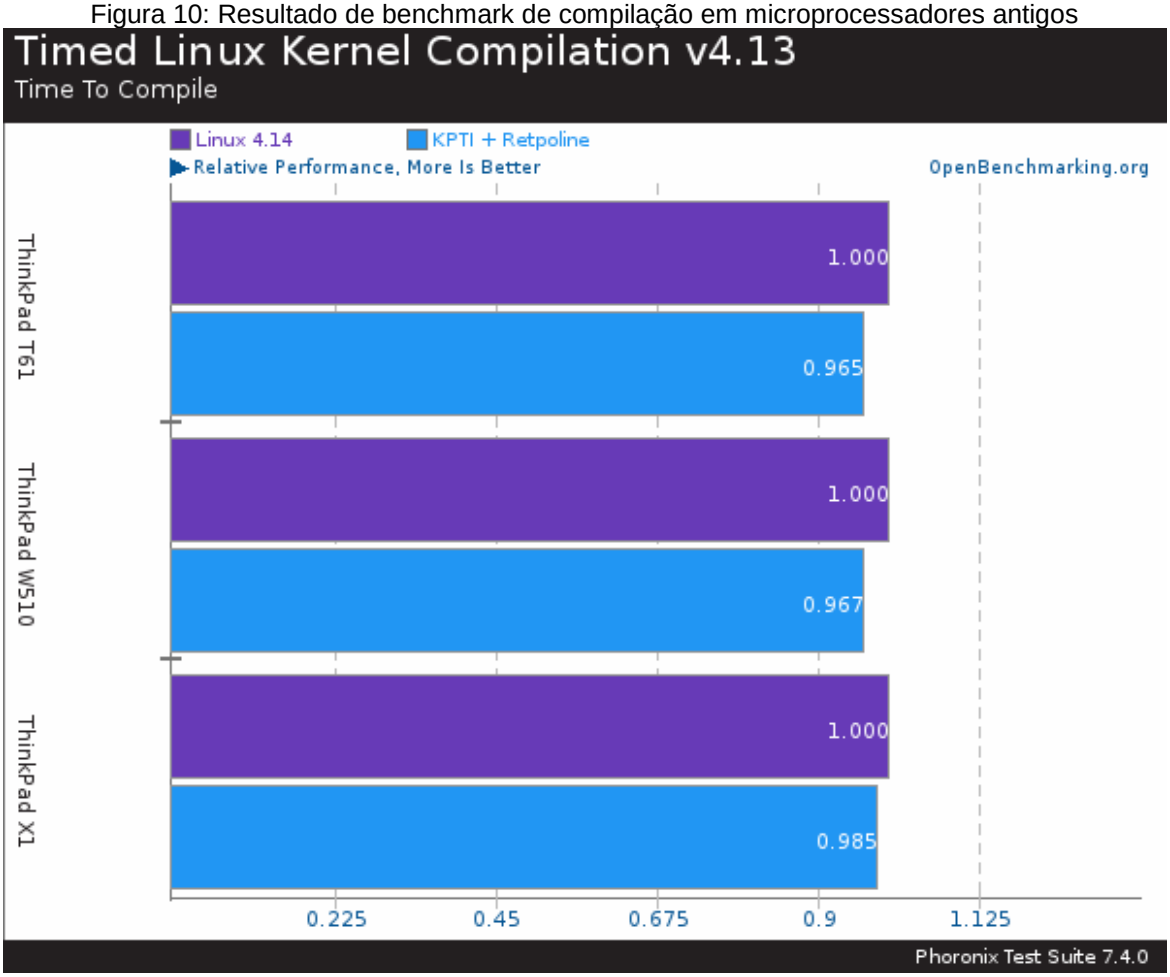
Fonte: Larabel, 2018b, p. 2

Figura 9: Resultado de benchmark SQLite em microprocessadores atual



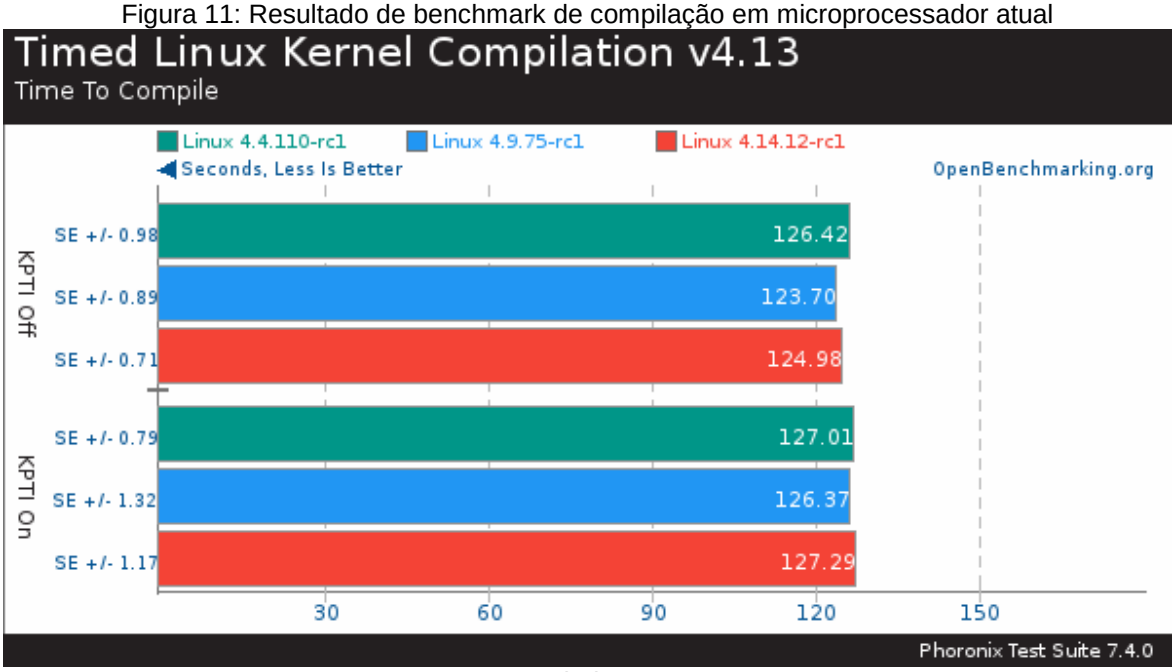
Fonte: Larabel, 2018a, p. 2

Figura 10: Resultado de benchmark de compilação em microprocessadores antigos



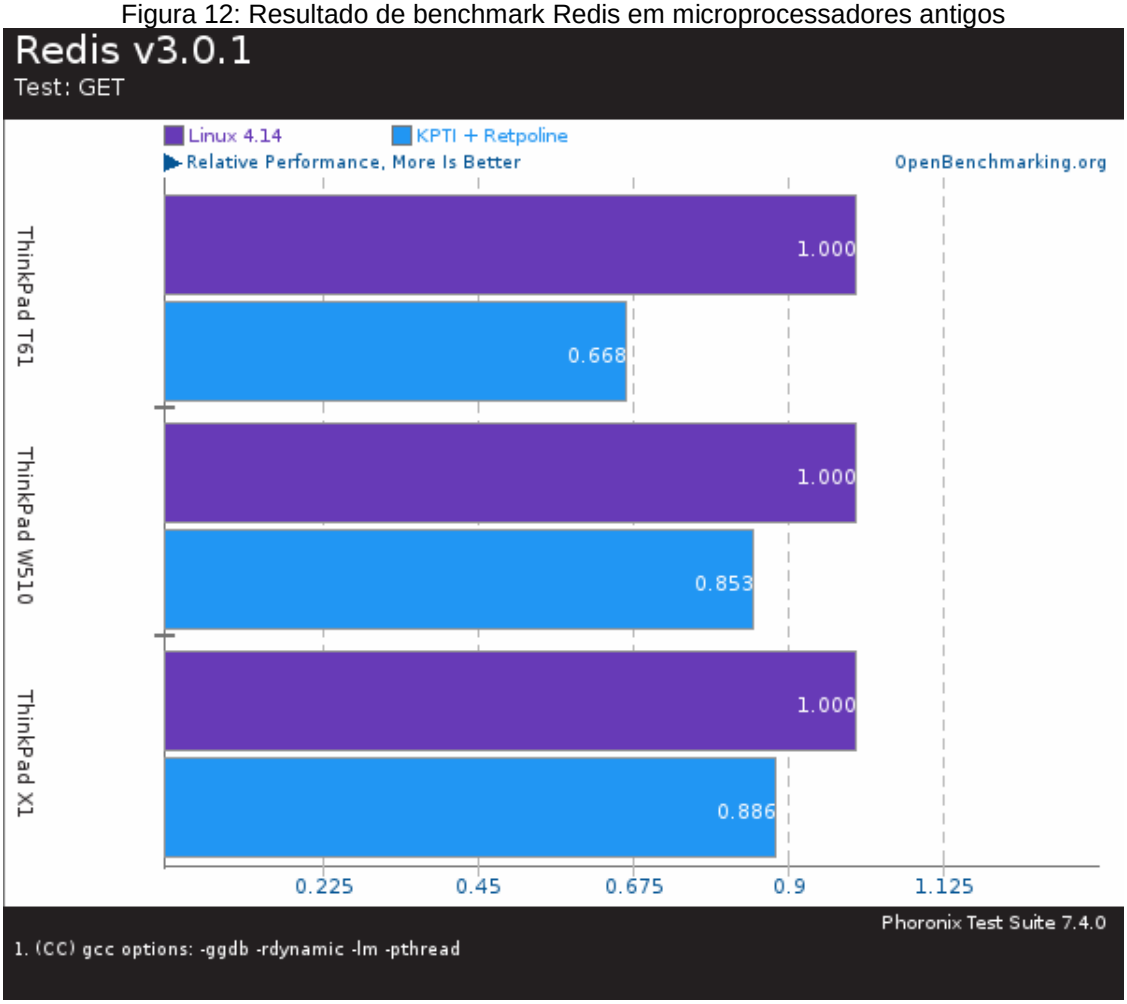
Fonte: Larabel, 2018b, p. 3

Figura 11: Resultado de benchmark de compilação em microprocessador atual



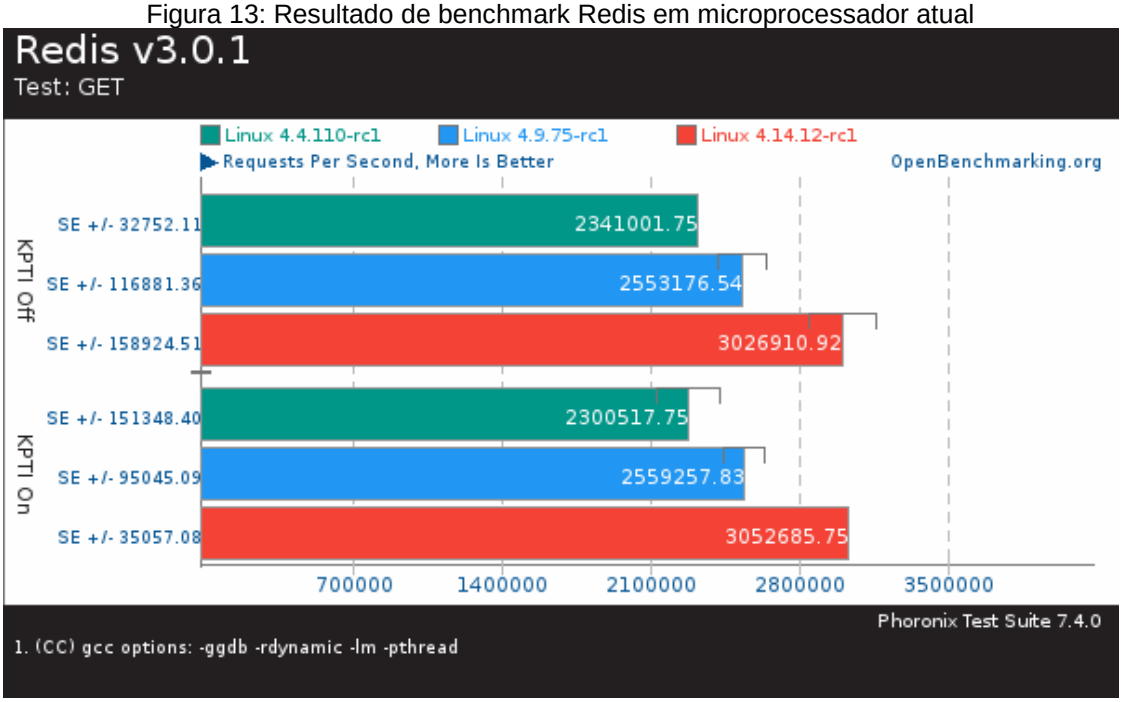
Fonte: Larabel, 2018a, p. 2

Figura 12: Resultado de benchmark Redis em microprocessadores antigos



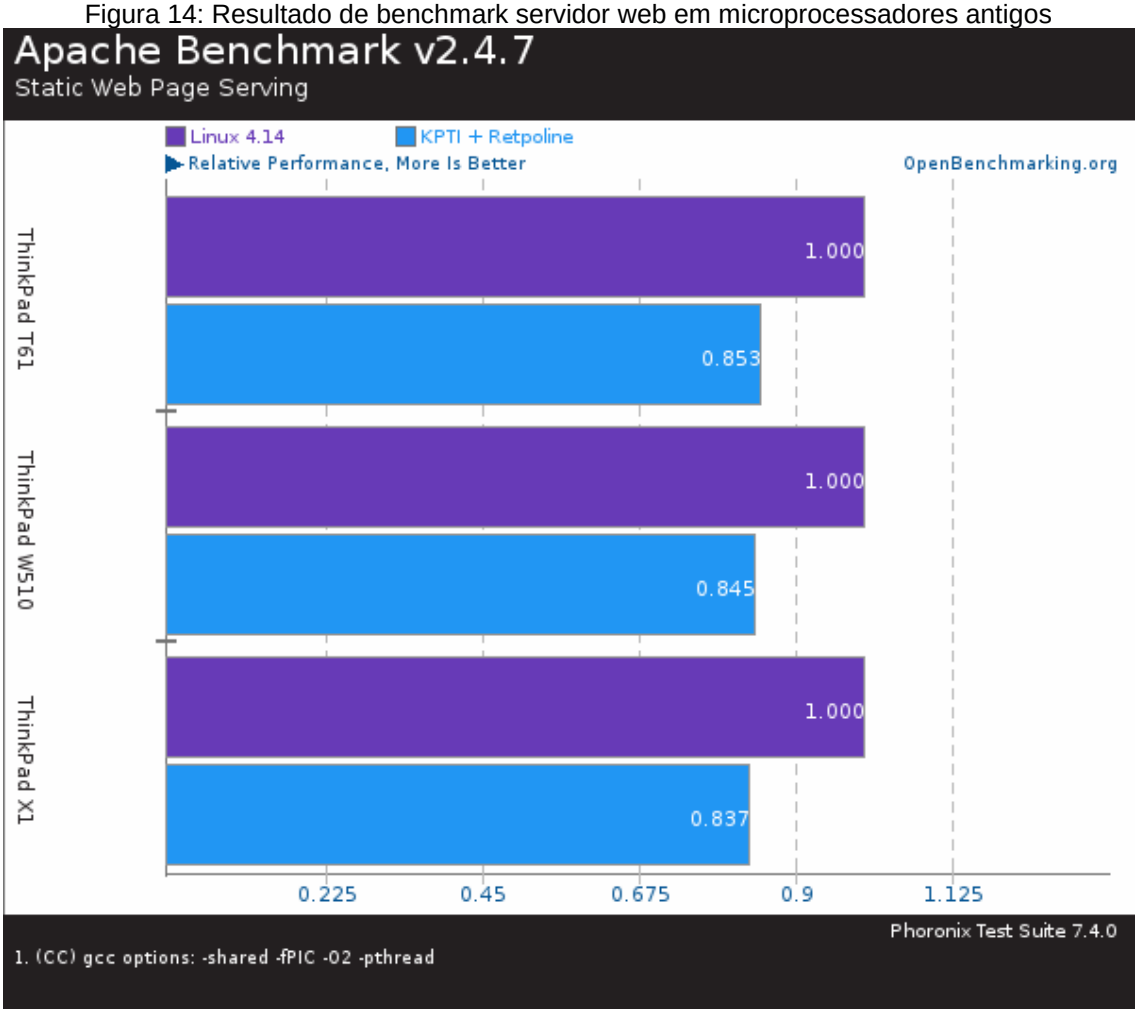
Fonte: Larabel, 2018b, p. 5

Figura 13: Resultado de benchmark Redis em microprocessador atual



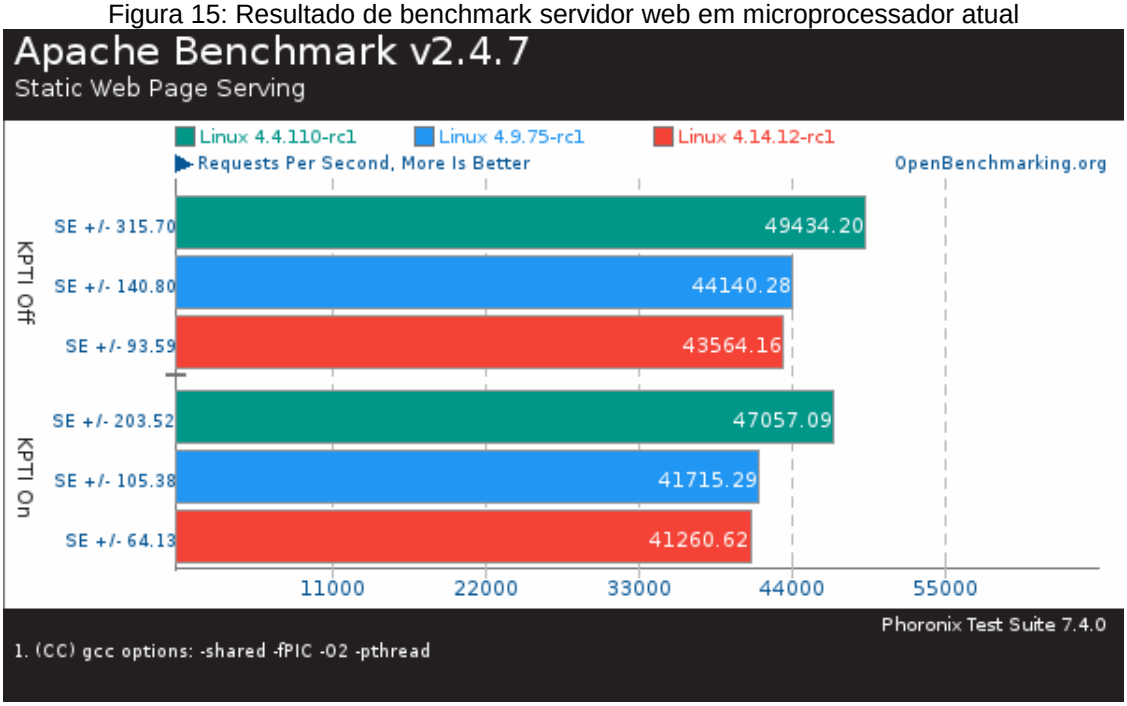
Fonte: Larabel, 2018a, p. 3

Figura 14: Resultado de benchmark servidor web em microprocessadores antigos



Fonte: Larabel, 2018b, p. 6

Figura 15: Resultado de benchmark servidor web em microprocessador atual



Fonte: Larabel, 2018a, p. 3

4.3- Testes produzidos pela Techspot

Em outro portal, o Techspot, foi realizado uma série de testes que apresentaram os mesmos resultados, com perda de 1% a 5% na maioria dos testes. Estes, foram realizados com a aplicação do *patch* para *Meltdown*. Em uma segunda bateria de testes, com aplicações das correções para todas as vulnerabilidades, novamente não se identificaram problemas na maioria dos testes. Os testes foram baseados em 1) *benchmarks* de armazenamento, 2) *benchmarks* de criação e renderização de conteúdo 3D, 3) *benchmarks* de produtividade e 4) *benchmarks* de jogos (*Cf.* WALTON, 2018, p. 1-3). Entre as baterias de testes utilizadas, optamos por considerar a mais atual por apresentar um *benchmark* completo com correções para todas as vulnerabilidades. Em seguida, apresentaremos os resultados obtidos. Para estes resultados foram utilizados os microprocessadores Intel® Core™ i3-8100 e Intel® Core™ i7-8700K, ambos de oitava geração.

Novamente, a título de informação, os dados concernentes às Figuras 16, 17, 18, 19, 20 e 21 foram detalhadas neste espaço, visto que a inserção das explicações logo abaixo de cada uma das Figuras faria com que o espaço gráfico das laudas não suportasse a quantia de Figuras por lauda.

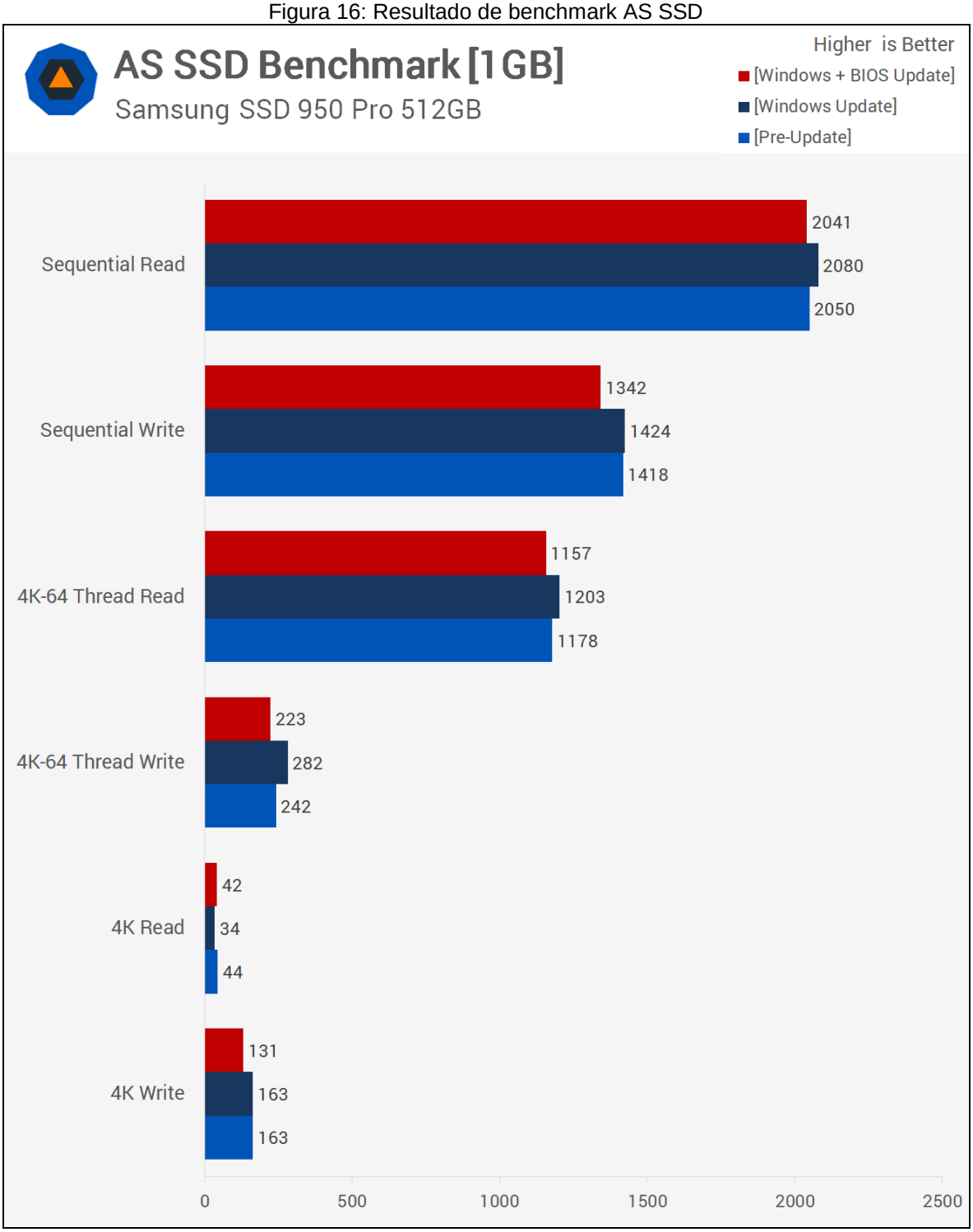
Os testes de armazenamento mostraram as maiores baixas entre os *benchmarks* disponíveis. Cada um dos *benchmarks* mostrou variações entre si, mas todos apontaram ao menos uma baixa de performance da ordem de 20%. Verifica-se na Figura 16 a perda de 19% escrita 4K (4K Write), na Figura 17 ocorreram perdas de 10%, 28% e 40%. No intuito de enfatizar esta perda alarmante de 40%, comenta o pesquisador independente Steven Walton:

“Eu fiz este teste dezenas de vezes depois de várias redefinições para tentar resolver se era apenas algum tipo de falha. Infelizmente esta é a figura que recebi” (WALTON, 2018, p. 3)[6].

[6] I ran this test dozens of times after multiple resets to try and work out if it was just some kind of glitch. Unfortunately this is the figure I kept receiving (WALTON, 2018, p. 3. *Tradução nossa!).*

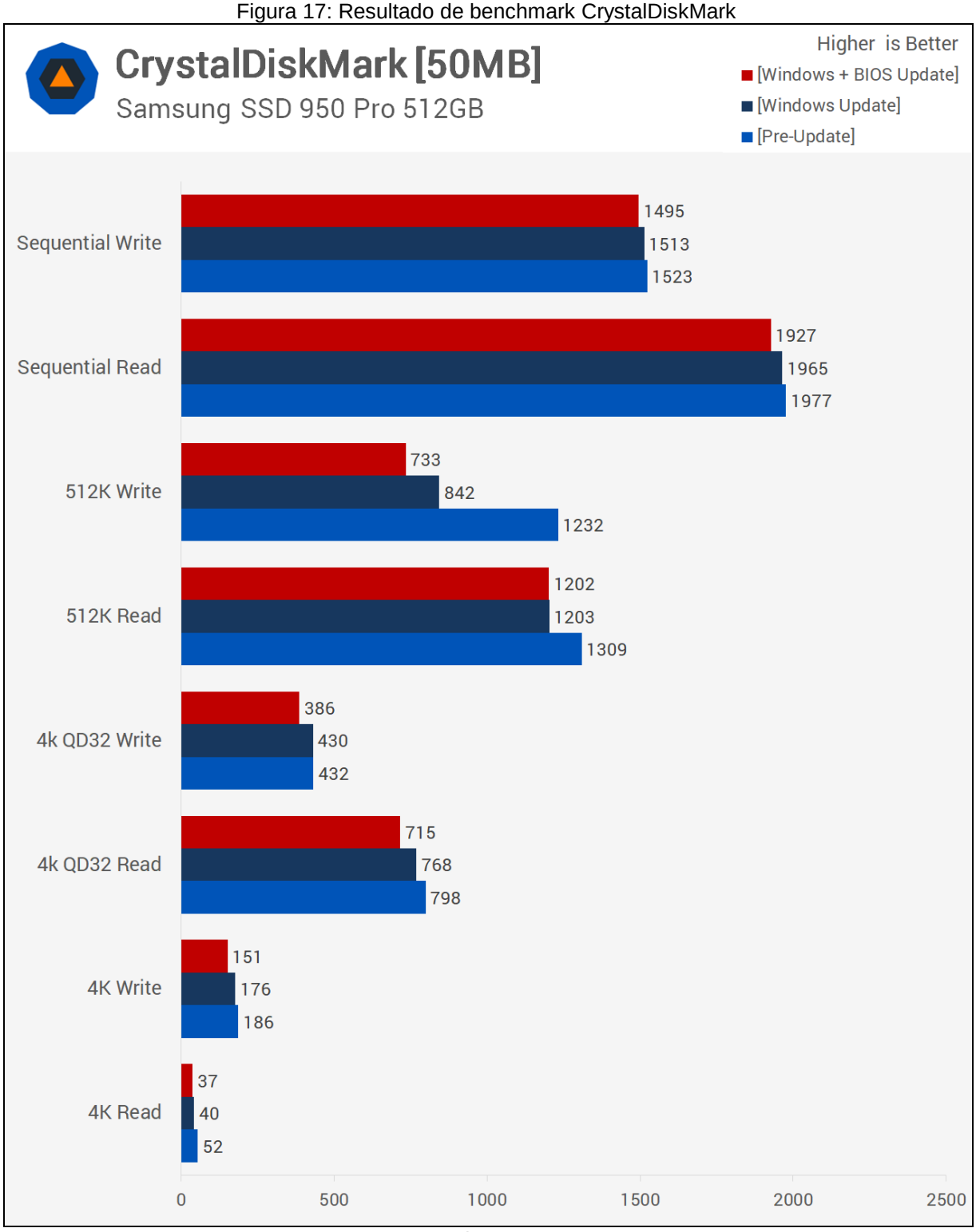
Nos *benchmarks* de produtividade e conteúdo 3D, as perdas não superaram mais que 3%, verificado nos exemplos das Figuras 18 e 19. Por fim, os *benchmarks* com vários jogos em diferentes qualidades de gráficos e *benchmarks* sintéticos do Geekbench também mostraram perdas máximas de 4%, conforme constatadas nas Figuras 20 e 21.

Figura 16: Resultado de benchmark AS SSD



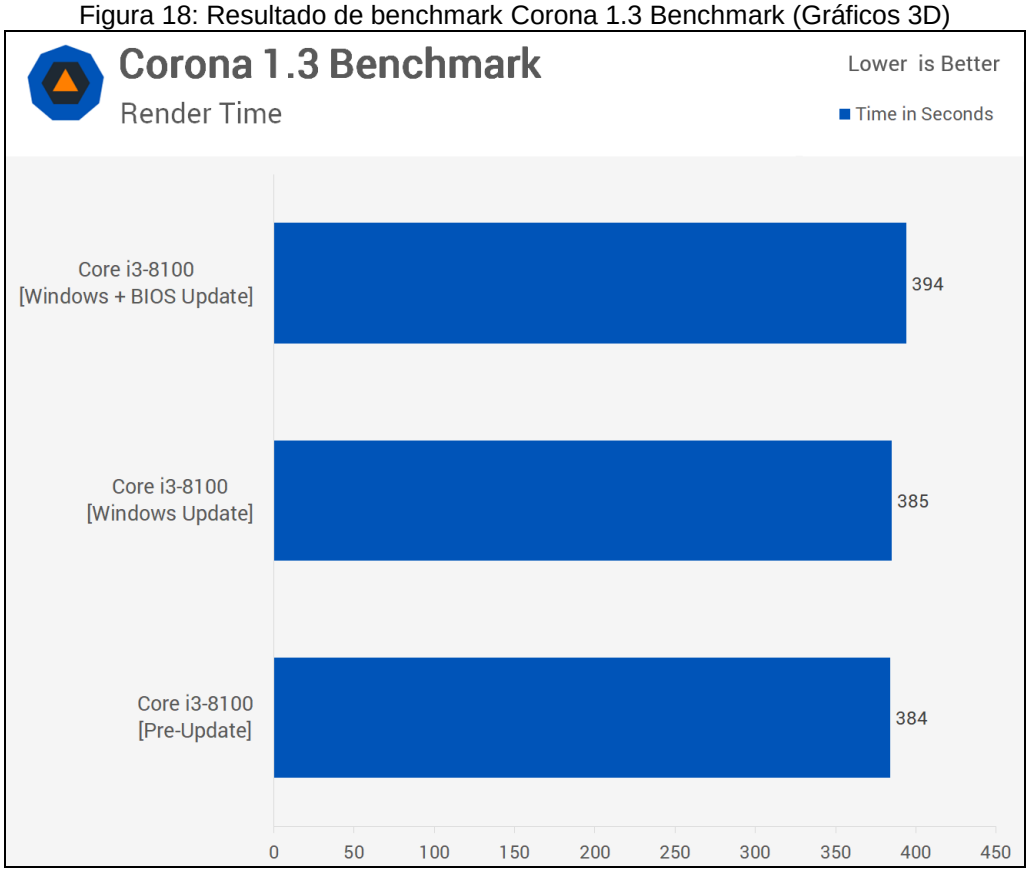
Fonte: Walton, 2018, p. 3

Figura 17: Resultado de benchmark CrystalDiskMark



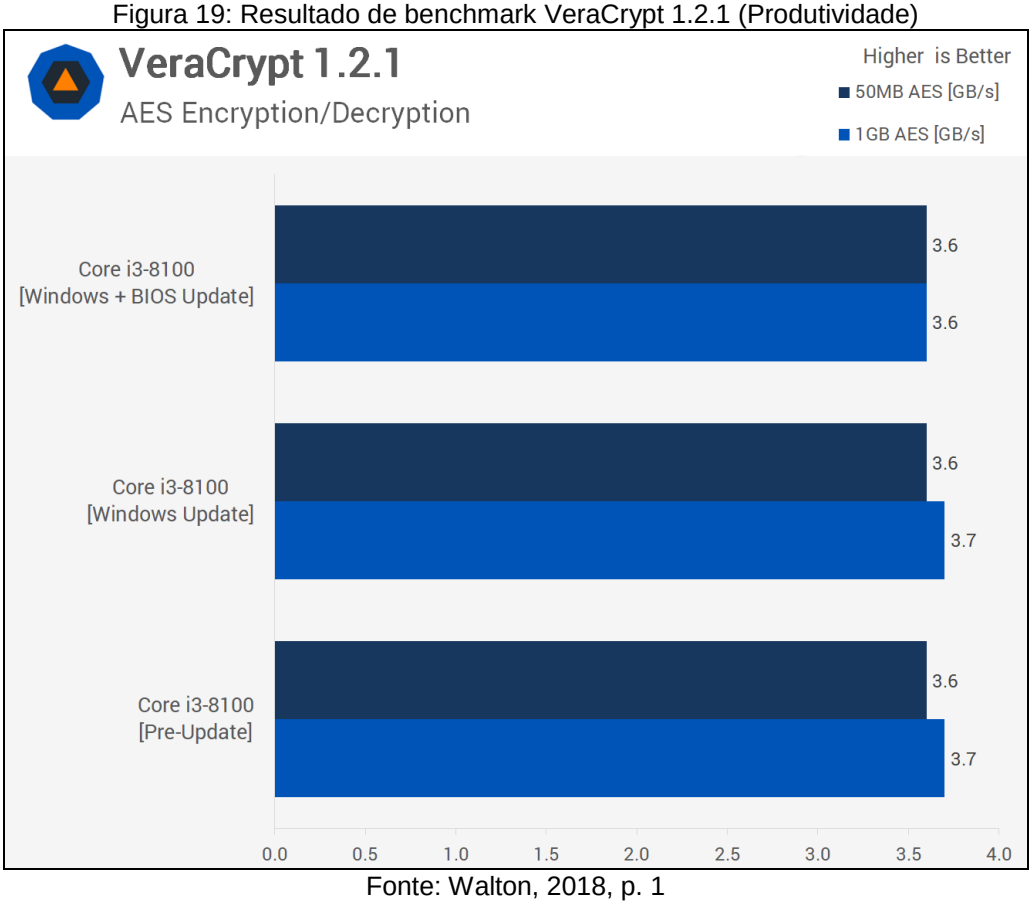
Fonte: Walton, 2018, p. 3

Figura 18: Resultado de benchmark Corona 1.3 Benchmark (Gráficos 3D)



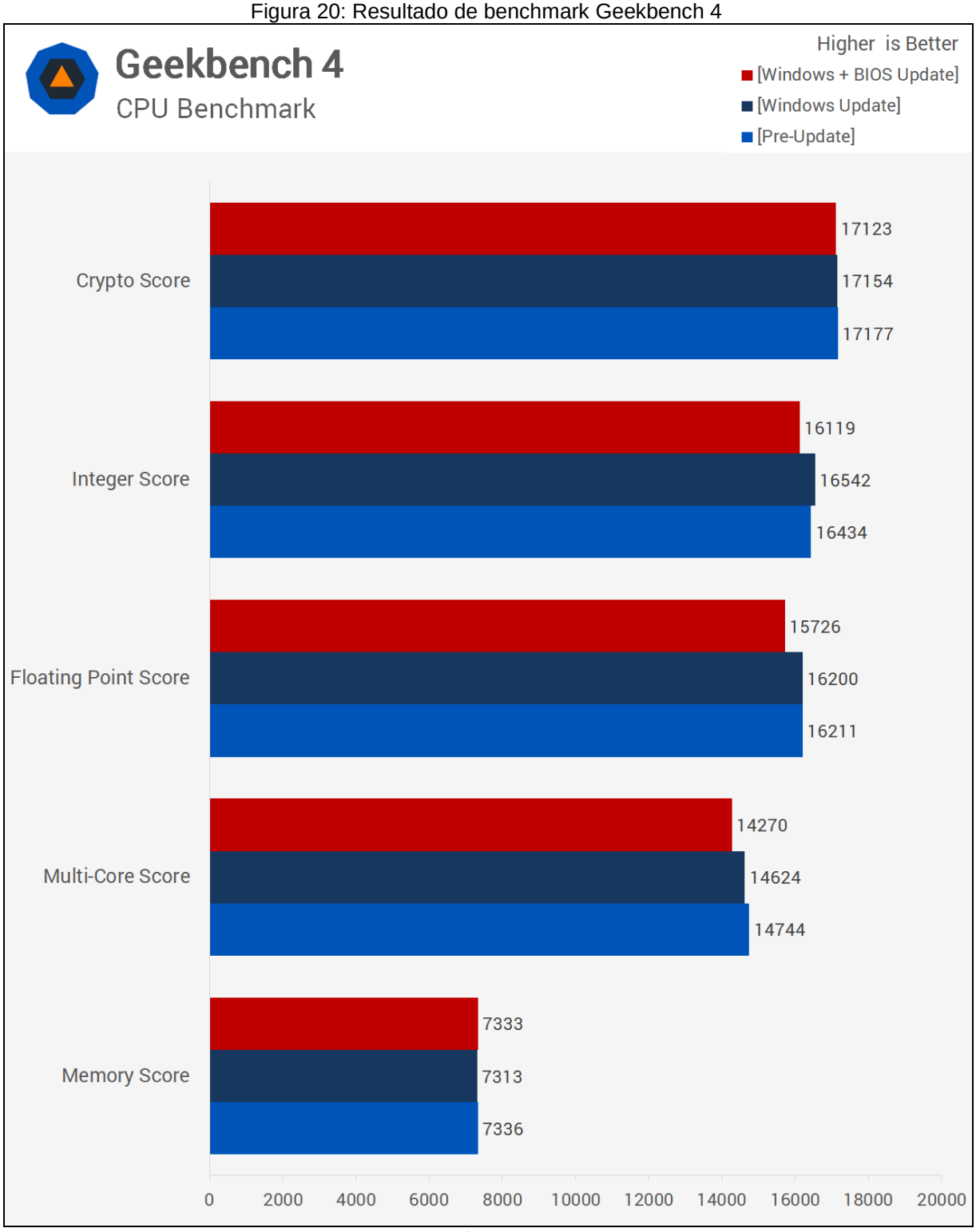
Fonte: Walton, 2018, p. 1

Figura 19: Resultado de benchmark VeraCrypt 1.2.1 (Produtividade)



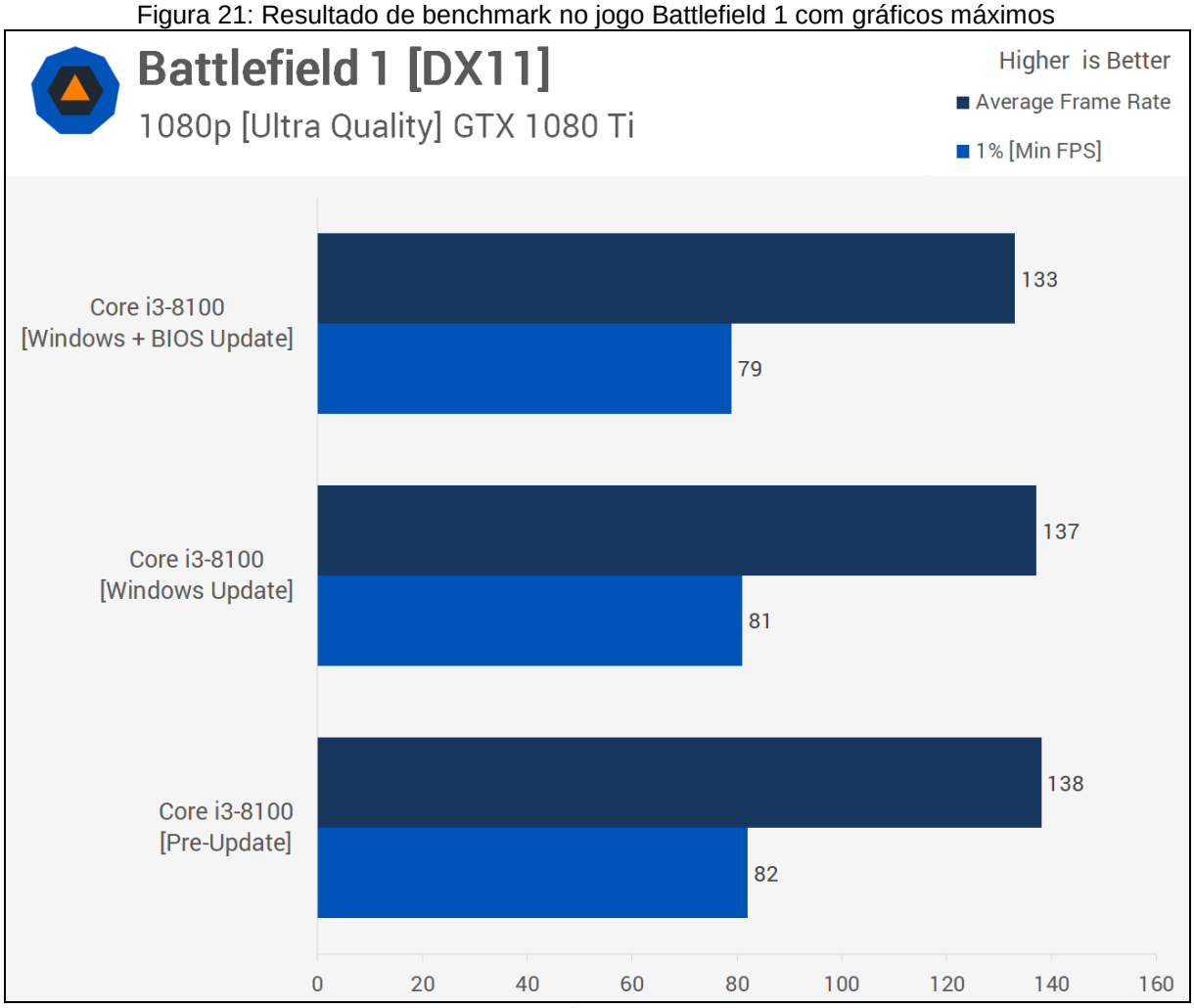
Fonte: Walton, 2018, p. 1

Figura 20: Resultado de benchmark Geekbench 4



Fonte: Walton, 2018, p. 2

Figura 21: Resultado de benchmark no jogo Battlefield 1 com gráficos máximos



Fonte: Walton, 2018, p. 2

CONCLUSÃO

As vulnerabilidades apresentadas no início do ano causaram várias mudanças no setor de tecnologia, a começar pelas próprias desenvolvedoras que precisaram adequar seus produtos em face das novas descobertas, mais ainda para as fabricantes de microprocessadores, entre elas a Intel, particularmente a maior afetada, que agora enfrenta diversas ações judiciais que procuram a reparação pelos produtos com defeito (WARREN, 2018). Tal qual uma fabricante de automóveis realiza *recalls* para a troca de equipamentos em seus veículos, questiona-se se a Intel não deveria ser obrigada a fazer o mesmo com seus microprocessadores. De fato, esta ação não deve ser considerada em seus planos, uma vez que trocar todos os processadores com as falhas identificadas seria praticamente inviável, a nível global. Na história, a Intel já realizou *recall* de produtos notadamente com defeito, quando em 1994 identificou microprocessadores Pentium P5 com falhas raras em cálculos com ponto flutuante, denominada FDIV Bug, foi o primeiro problema de hardware em um computador a ganhar notoriedade mundial, e claro, custando milhões para a fabricante (ATHOW, 2014). No entanto, a questão aludida não foi objeto de análise deste Trabalho de Conclusão de Curso, mas pode provocar outros estudantes a investigarem a questão do ressarcimento e/ou indenização em decorrência deste desrespeito aos direitos do consumidor, haja vista que os prejudicados não se fixam apenas ao solo norteamericano.

Se um *recall* não vai acontecer, soluções por meio de *software* é a principal aposta do setor, objeto de análise central deste Trabalho. Estas atualizações são fáceis de entregar a nível global, e, de fato, já foram feitas na primeira semana da divulgação das vulnerabilidades, com complementações nas semanas subsequentes. Estas atualizações criam barreiras para que as vulnerabilidades não sejam exploradas. Como efeito colateral, elas implicam perda de poder de processamento em diversos graus, uma vez que alteram o modo como o computador interage com a BIOS (Basic Input/Output System - Sistema Básico de Entrada/Saída) e a memória *kernel,* e como a execução dinâmica trabalha para entregar performance ao microprocessador. Como validado em vários *benchmarks*, estas perdas são reais, e, apesar de, na absoluta maioria das utilizações residenciais e empresariais não apresentarem mais que 4% a 8%, em alguns casos,

especificamente, chegam a alarmantes 48% a perda de performance proporcionada pela adoção das correções. Em resposta a estes resultados, a Intel garante que está produzindo soluções para que as vulnerabilidades sejam corrigidas e que o impacto destas correções seja mínimo para os usuários.

Por fim, verificou-se que *hardware* e *software* mais antigos podem admitir maiores perdas de rendimento, uma vez que os sistemas atualizados tratam de melhor forma as atualizações recebidas. Portanto, é fundamental e consenso do setor, que se mantenham as atualizações de sistema operacional, *firmware* e aplicativos mais recentes disponíveis.

Ainda se assegura que estas linhas produzidas, elaboradas no decorrer deste primeiro semestre letivo, não esgotam a potencialidade que esta temática pode provocar, mas, conforme enfatizado, pode ser explorada no intuito de validar os testes já realizados pela Phoronix e pela Techspot, a questão jurídica dos direitos do consumidor, bem como novas variantes destas vulnerabilidades, tais como *MeltdownPrime* e *SpectrePrime*, entre outras.

REFERÊNCIAS

ABRAMS, Lawrence. List of Meltdown and Spectre Vulnerability Advisories, Patches, & Updates. Artigo postado em 3 de janeiro de 2018. **Blepping Computer**, 2003-2018. Disponível em <https://www.bleepingcomputer.com/news/security/list-of-meltdown-and-spectre-vulnerability-advisories-patches-and-updates/>. Acesso em: 20 jul. 2018, às 10h.

ARRUDA, Felipe. A história dos processadores. Artigo postado em 16 de junho de 2011. **Tecmundo**, 2018. Disponível em: <https://www.tecmundo.com.br/historia/2157-a-historia-dos-processadores.htm>. Acesso em: 3 jul. 2018, às 11h.

ATTOW, Desire. Pentium FDIV: The processor bug that shook the world. **Tech Radar**, 2018. Artigo postado em 30 de outubro de 2014. Disponível em: <https://www.techradar.com/news/computing-components/processors/pentium-fdiv-the-processor-bug-that-shook-the-world-1270773>. Acesso em: 25 jul. 2018, às 16h.

CIMPANU, Catalin. Microsoft Releases Emergency Updates to Fix Meltdown and Spectre CPU Flaws. Artigo postado em 4 de janeiro de 2018. **Blepping Computer**, 2003-2018. Disponível em: <https://www.bleepingcomputer.com/news/microsoft/microsoft-releases-emergency-updates-to-fix-meltdown-and-spectre-cpu-flaws/>. Acesso em: 20 jul. 2018, às 11h.

GATTO, Elaine Cecília. Arquitetura de John Von Neumann. Artigo postado em 18 de outubro de 2016. **Embarcados**, 2007-2018. Disponível em: <https://www.embarcados.com.br/arquitetura-de-john-von-neumann/>. Acesso em: 4 jul. 2018, às 21h35min.

GRAZ UNIVERSITY OF TECHNOLOGY, 2018. **Meltdown and Spectre:** Vulnerabilities in modern computers leak passwords and sensitive data. Disponível em: <https://meltdownattack.com/>. Acesso em: 6 jul. 2018, às 11h.

HRUSKA, Joel. What is Speculative Execution? Artigo postado em 10 de janeiro de 2018. **Extremetech**, 1996-2018. Disponível em: <https://www.extremetech.com/computing/261792-what-is-speculative-execution>. Acesso em: 4 jul. 2018, às 11h.

INTEL. The Story of the Intel® 4004. **Intel**, 2018a. Disponível em: <https://www.intel.com/content/www/us/en/history/museum-story-of-intel-4004.html>. Acesso em: 3 jul. 2018, às 11h.

_____. Industry Testing Shows Recently Released Security Updates Not Impacting Performance in Real-World Deployments. Artigo postado em 4 de janeiro de 2018. **Intel**, 2018b. Disponível em: <https://newsroom.intel.com/news-releases/industry-testing-shows-recently-released-security-updates-not-impacting-performance-real-world-deployments/>. Acesso em: 21 jul 2018, às 12h.

_____. Intel Offers Security Issue Update. Artigo postado em 9 de janeiro de 2018. **Intel**, 2018c. Disponível em: <https://newsroom.intel.com/news/intel-offers-security-issue-update/>. Acesso em: 21 jul. 2018, às 12h.

KOCHER, Paul; *et al*. **Spectre attacks**: exploiting speculative execution. 2018, 19p. Disponível em:<https://spectreattack.com/spectre.pdf>. Acesso em: 6 jul. 2018, às 11h.

LARABEL, Michael. Linux KPTI Tests Using Linux 4.14 vs. 4.9 vs. 4.4. Artigo postado em 4 de janeiro de 2018. **Phoronix**, 2004-2018a. Disponível em: <https://www.phoronix.com/scan.php?page=article&item=linux-kpti-pcid&num=1>. Acesso em: 22 jul de 2018, às 12h.

_____. KPTI + Retpoline Linux Benchmarking On Older Clarksfield / Penryn ThinkPads. Artigo postado em 11 de janeiro de 2018. **Phoronix**, 2004-2018b. Disponível em: <https://www.phoronix.com/scan.php?page=article&item=pre-pcid-kptiretpoline&num=1>. Acesso em: 22 jul de 2018, às 12h.

LIPP, Moritz; *et al*. **Meltdown**. 2018, 16p. Disponível em: <https://meltdownattack.com/meltdown.pdf>. Acesso em: 6 jul. 2018, às 11h.

MICROSOFT, 2018. **Windows Server guidance to protect against speculative execution side-channel vulnerabilities:** Operating system version. Disponível em: <https://support.microsoft.com/en-us/help/4072698/windows-server-guidance-to-protect-against-the-speculative-execution>. Acesso em: 18 jul. 2018, às 11h.

MITRE, 1999-2018. **CVE-2017-5754**, 2017. Artigo postado em 2 de janeiro de 2017. Disponível em: <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-5754>. Acesso em: 17 jul. 2018, às 11h.

_____. **CVE-2017-5753**, 2017. Artigo postado em 2 de janeiro de 2017. Disponível em: <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-5753>. Acesso em: 17 jul. 2018, às 11h.

_____. **CVE-2017-5715**, 2017. Artigo postado em 2 de janeiro de 2017. Disponível em: <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-5715>. Acesso em: 17 jul. 2018, às 11h.

_____. **Common Vulnerabilities and Exposures**, 2018. Disponível em: <https://cve.mitre.org>. Acesso em: 10 jul. 2018, às 10h.

MOREIRA, Eduardo. Intel 4004, o primeiro processador da história, comemora 40 anos de idade. Artigo postado em 16 de novembro de 2011.**Techtudo**, 2000-2018. Disponível em: <http://www.techtudo.com.br/artigos/noticia/2011/11/intel-4004-o-primeiro-processador-da-historia-comemora-40-anos-de-idade.html>. Acesso em: 3 jul. 2018, às 11h.

MORIMOTO, Carlos E. Processadores, da pré-história ao Pentium 4. Artigo postado em 29 de dezembro de 2010. **Guia do hardware,** 1999-2018. Disponível em:

<https://www.hardware.com.br/guias/historia-processadores/386-era-32bits.html>. Acesso em: 5 jul. 2018, às 16h20min.

SHANKLAND, Stephen; HAUTALA, Laura. Nope, no Intel chip recall after Spectre and Meltdown, CEO says. Artigo postado em 4 de janeiro de 2018. **CNET**, 2018. Disponível em <https://www.cnet.com/news/meltdown-spectre-intel-ceo-no-recall-chip-processor/>. Acesso em: 19 jul. 2018, às 12h.

SHENOY, Navin. Intel Security Issue Update: Initial Performance Data Results for Client Systems. Artigo postado em 10 de janeiro de 2018. **Intel**, 2018. Disponível em <https://newsroom.intel.com/editorials/intel-security-issue-update-initial-performance-data-results-client-systems/>. Acesso em: 21 jul. 2018, às 12h.

SOUZA, Antonio Carlos dos Santos; *et al*. *Uma Introdução a Computação:* História e Ciência. São Paulo : **Ixtlan**., 2016.Disponível em: <http://www.labrasoft.ifba.edu.br/wp-content/uploads/2014/03/UmaIntroducaoAComputacao-HistoriaCiencia.pdf>. Acesso em: 3 jul. 2018, às 11h.

UHLMANN, Erwin Alexander. Arquitetura de Computadores. **Instituto Siegen.** Guarulhos, 2014. 25p. Disponível em: <http://institutosiegen.com.br/wp-content/uploads/2018/03/Arquitetura-de-Computadores.pdf>. Acesso em: 4 jul. 2018, às 21h35min.

WALTON, Steven. Patched Desktop PC: Meltdown & Spectre Benchmarked. Artigo postado em 7 de janeiro de 2018. **Techspot**, 2018. Disponível em: <https://www.techspot.com/article/1556-meltdown-and-spectre-cpu-performance-windows/>. Acesso em: 22 jul. 2018, às 13h.

WARREN, Tom. Intel facing 32 lawsuits over Meltdown and Spectre CPU security flaws. Artigo postado em 16 de fevereiro de 2018. **The Verge**, 2011-2018. Disponível em: <https://www.theverge.com/2018/2/16/17020048/intel-spectre-meltdown-class-action-lawsuits>. Acesso em 25 jul. 2018, às 16h.

WILLIAMS, Chris. Meltdown, Spectre: The password theft bugs at the heart of Intel CPUs. Artigo postado em 4 de janeiro de 2018.**The Register**, 1998-2018. Disponível em: <http://www.theregister.co.uk/2018/01/04/intel_amd_arm_cpu_vulnerability/>. Acesso em: 6 jul. 2018, às 11h.

WIKIPEDIA, 2018. **Kernel page-table isolation**. Disponível em <https://en.wikipedia.org/wiki/Kernel_page-table_isolation>. Acesso em: 19 jul. 2018, às 11h.

ANEXO A - TABELA DE BENCHMARK DA INTEL

Benchmark	Workload Description	8th Generation Desktop Intel® Core i7 8700K Processor	8th Generation Mobile Intel® Core™ i7-8650U Processor	7th Generation Mobile Intel® Core™ i7 7920HQ Processor	6th Generation Desktop Intel® Core™ i7 6700K Processor		
CPU Code Name		Coffee Lake	Kaby Lake	Kaby Lake	Skylake	Skylake	Skylake
OS		Windows 10	Windows 10	Windows 10	Windows 10	Windows 7	Windows 7
Storage		SSD	SSD	SSD	SSD	SSD	HDD
Introduction Date		Q4'17	Q3'17	Q1'17	Q3'15	Q3'15	Q3'15
		Relative Performance (Fully Mitigated System / Non Mitigated System at 100%)					
SYSmark 2014 SE Overall	Windows Application-based Office Productivity, Data/Financial Analysis and Media Creation.	94%	95%	93%	92%	94%	100%
SYSMark 2014 SE Office Productivity		95%	95%	95%	90%	93%	96%
SYSMark 2014 SE Media Creation		96%	97%	96%	96%	97%	97%
SYSMark 2014 SE Data/Finance Analysis		97%	98%	98%	103%	99%	106%
SYSMark 2014 SE Responsiveness		88%	86%	86%	79%	89%	101%
PCMark 10 - Overall	Windows application based benchmark covering essentials, content creation and productivity	96%	96%	97%	96%	96%	96%
PCMark 10 - Essentials		96%	96%	97%	96%	93%	95%
PCMark 10 - Productivity		96%	94%	95%	94%	97%	97%
PCMark 10 - Digital Content Creation		98%	98%	98%	99%	97%	97%
3DMark Sky Diver - Overall	DX11 Gaming performance	100%	99%	100%	101%	100%	100%
3DMark Sky Diver - Graphics		100%	99%	100%	101%	100%	100%
3DMark Sky Diver - Physics		99%	98%	100%	99%	97%	99%
3DMark Sky Diver - Combined		100%	99%	100%	101%	100%	101%
WebXPRT 2015 Notw: Windows 10 on Edge Browser Windows 7 on IE Browser	Web applications using six usage scenarios: Photo Enhancement, Organize Album, Stock Option Pricing, Local Notes, Sales Graphs, Explore DNA Sequencing.	92%	90%	93%	90%	95%	92%

The benchmark results reported above may need to be revised as additional testing is conducted. The results depend on the specific platform configurations and workloads utilized in the testing, and may not be applicable to any particular user's components, computer system or workloads. The results are not necessarily representative of other benchmarks and other benchmark results may show greater or lesser impact from mitigations.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

For more information about benchmarks and performance test results, go to www.intel.com/benchmarks.

KBL-R U 4+2 Configuration:
Processor: Intel® Core™ i7-8650U Processor (KBL-R U 4+2) PL1=15W TDP, 4C8T, Turbo up to 4.2GHz **Memory:** 2x4GB DDR4-2400 1Rx8 Samsung M471A5143EB1
Storage: Intel® 600p m.2 NVME SSD **Display Resolution**: 1920x1080 **OS:** Windows* 10 Build RS3 16299.15. Power policy set to AC/High Perf for all benchmarks **Graphics driver**: 15.60.4901_whql **RST:** 15.9.1.1018_pv-RFfix
CFL-S 6+2 95W Configuration:
Processor: Intel® Core™ i7-8700K Processor (CFL-S 6+2), PL1=95W TDP, 6C12T, Turbo up to 4.7GHz **Memory:** 2x8GB DDR4-2666 2Rx4 HyperX HX426C15FBK2/16 **Storage**: Intel® 600p M.2 NVMe SSD **Display Resolution**: 1920x1080 **OS:** Windows 10 Build RS3 16299.15. Power policy set to AC/HighPerf for all benchmarks **Graphics driver**: 15.60.4877_Whql, **RST:** 15.9.1.1018_pv-Rffix
KBL-H 4+2 45W Configuration:
Processor: Intel® Core™ i7-7920HQ Processor (KBL-H 4+2), PL1=45W TDP, 4C8T, Turbo up to 4.1GHz **Memory:** 2x4GB DDR4-2400 1Rx8 Samsung M471A5143EB1 **Storage**: Intel® 600p M.2 NVMe SSD **Display Resolution**: 1920x1080 **OS:** Windows 10 Build RS3 16299.15. Power policy set to AC/HighPerf for all benchmarks **Graphics driver**: 15.60.4877_Whql, **RST:** 15.9.1.1018_pv-Rffix
SKL-S 4+2 91W Configuration:
Processor: Intel® Core™ i7-6700K Processor (SKL-S 4+2), PL1=91W TDP, 4C8T, Turbo up to 4.2GHz **Memory:** 2x8GB DDR4-2400 [running at 2133] 2Rx8 G.Skill Ripjaws F4-2400C15D-16GVR **Storage**: Intel® 600p M.2 NVMe SSD **Display Resolution**: 1900x1200 **OS:** Windows 10 Build RS3 16299.15. Power policy set to AC/HighPerf for all benchmarks **Graphics driver**: 10.18.15.4256, **RST:** 14.6.0.1029
SKL-S 4+2 91W Configuration:
Processor: Intel® Core™ i7-6700K Processor (SKL-S 4+2), PL1=91W TDP, 4C8T, Turbo up to 4.2GHz **Memory:** 2x8GB DDR4-2400 [running at 2133] 2Rx8 G.Skill Ripjaws F4-2400C15D-16GVR **Storage**: Intel® 540s Series 240GB SATA SSD **Display Resolution**: 190x1200 **OS:** Windows 7 Build 7601 Service Pack 1. Power policy set to AC/HighPerf for all benchmarks **Graphics driver**: 10.18.15.4256, **RST:** 14.6.0.1029
SKL-S 4+2 91W Configuration:
Processor: Intel® Core™ i7-6700K Processor (SKL-S 4+2), PL1=91W TDP, 4C8T, Turbo up to 4.2GHz **Memory:** 2x8GB DDR4-2400 [running at 2133] 2Rx8 G.Skill Ripjaws F4-2400C15D-16GVR **Storage**: Western Digital Black Edition 1TB 7200RPM SATA HDD WD1003FZEX **Display Resolution**: 1900x1200 **OS:** Windows 7 Build 7601 Service Pack 1. Power policy set to AC/HighPerf for all benchmarks **Graphics driver**: 10.18.15.4256, **RST:** 14.6.0.1029