Thermal Attacks on Storage Systems

Nathanael Paul Sudhanva Gurumurthi David Evans
University of Virginia, Department of Computer Science
Charlottesville, VA
{nate, gurumurthi, evans}@cs.virginia.edu

Abstract

Disk drives are a performance bottleneck for data-intensive applications. Drive manufacturers have continued to increase the rotational speeds to meet performance requirements, but the faster drives consume more power and run hotter. Future drives will soon be operating at temperatures that threaten drive reliability. One strategy that has been proposed for increasing drive performance without sacrificing reliability is throttling. Throttling delays service to I/O requests after the disk temperature exceeds a set threshold temperature until the temperatures drops. In this paper, we explore the possibility that a malicious attacker with the ability to issue disk read requests may be able to exploit throttling to carry out a denial-of-service attack on a storage system. Our results reveal that damaging attacks are possible when throttling is used, and argue for the use of variable speed disks as a less vulnerable thermal management alternative.

Keywords: storage systems, security, thermal management, denial-of-service attacks.

1 Introduction

As performance demands from different data-intensive applications, such as databases, scientific applications, and web servers increases, the disk has a larger impact on performance. Rotational speeds support higher data transfer rates, but the faster speeds cause a cubic increase in the temperature with more power consumption. Current drives are designed to operate reliably under a certain temperature, known as the *thermal envelope*. Exceeding the thermal envelope decreases the drive's reliability and can have serious consequences on data integrity and system availability.

To control the heat in a disk, future drive designs are expected to employ dynamic thermal management (DTM). The two main proposed techniques are throttling and variable speed disks. Throttling temporarily stops servicing I/O requests once the disk temperature exceeds a certain limit and makes the disk idle (during which time no I/O can be done) until the temperature drops to a cooler temperature. A variable speed disk is a disk that can spin at two (or more) different speeds. Slower rotational speeds dissipate less heat due to the cubic relationship between disk RPM and viscous heating. If the drive exceeds the thermal envelope, the drive can revert to a slower rotation speed to cool itself down while continuing to service I/O requests. Although throttling may be an easier design option, it can drastically slow down I/O response times [16]. The main problem with throttling is that the process of cooling takes longer than heating. Once throttling is executed, reaching the throttling point after resuming service is easier, since the disk cools slower than it heats up.

Our work shows that throttling creates the possibility of a denial-of-service (DoS) attack by deliberately constructing a workload that raises the temperature of the drive enough to trigger the throttling shut-down. Our new thermal attack on future storage systems is unrecognized by current Intrusion Detection Systems (IDS) since the malicious activity is solely comprised of read requests. AV scanner signatures and heuristics also miss the malicious activity, since they primarily look for presence of code signatures and types of accesses to certain files and not at low level characteristics of disk I/O traffic. We present results of simulated experiments that show the potential impact of maliciously constructed traffic on a drive that performs DTM using throttling.

Next, we provide background on thermal properties of disk drives. Section 3 discusses the experimental setup and workloads used. We describe the thermal denial-of-service attacks in Section 4 followed by our argument for eliminating throttling in Section 5. We conclude by arguing that an architectural change supporting variable-speed disks is a more promising solution for DTM than throttling. Our real-system experiments under elevated temperatures indicate that higher temperatures do not cause immediate failures and are more of a long-term reliability issue, and therefore excursions above the thermal envelope can be tolerated early in the lifetime of a disk drive, whereby allowing for high performance.

2 Background

Because disk drives are a major performance bottleneck, vendors attempt to maximize throughput. One way this is currently done is making the drive spin faster, but this increases power consumption. The heat from the power dissipation forces the drive to operate at a higher temperature, influencing the long-term reliability of the drive.

Other groups have studied thermal issues in storage systems including [2, 26, 27, 9, 16]. Dadvar and Skadron [4] have found thermal related attacks on CPUs and Hasan et al. [11] recently showed SMT-related DoS attacks. In 2003 Govindavajhala and Appel [8] presented an approach to attack a Java Virtual Machine using external heat to cause a bit flip that could be exploited to compromise virtual machine security.

Our DoS attack is against the disk drive itself and not the CPU. The attack is made possible if a throttling mechanism is used in the disk drive to control heat [1, 21]. Gurumurthi et al. initially proposed variable speed disks [9], and others tried differing approaches of energy management in network servers by using laptop disks and traditional spin-down based techniques [2]. Their findings indicate variable-speed disks are the best in saving energy. Zhu et al. propose cache management schemes to lower energy consumption [26, 27] while others have studied thermal properties in storage systems [17].

2.1 Disk Drive Properties

Figure 1 depicts a modern disk drive with its disk arm moving along a circular arc above a spinning platter (current speeds are 5400, 7200, 10K, and 15K Rotations Per Minute or RPM). The separation between the head and the platter is typically only a few nanometers. Modern drives can have multiple platters, with each platter holding data on both the top and bottom surfaces. A separate disk head is used to read and write on each surface, but all disk arms are attached to a central pivot point. The concentric circles on the platter are called *tracks*. Since the outer tracks are larger in circumference, they can store more bits than the inner ones.

The raw transfer rate of a disk drive is known as the Internal Data Rate (IDR). It depends on the linear density and the rotation speed. The IDR has increased by 40% each year over the last 15 years [14], due to significant growth in both the linear density and rotation speed. However, due to significant hurdles in scaling the linear density, the rotation speed needs to increase faster in order to maintain typical IDR growth.

Power dissipation as heat is generated in the disk drive and is given by the power equation:

$$Power = num\_platters \times size\_platters^{4.6} \times rotations\_per\_minute^{2.8}$$

Increasing the rotation speed leads to nearly a cubic increase in the heat that is generated. Traditionally, in order to attain the desired IDR, the platters have been shrunk (which leads to a nearly fifth power drop in the power), and this margin has been exploited to boost the rotation speed. However, with linear density growth slowing down, the rotation speed needs to be scaled even higher, but this tradeoff is less effective even at very small platter sizes [10].

Disks are designed to operate reliably only when the temperature is below a threshold known as the *thermal envelope*. Higher temperatures can cause the disk to have higher off-track errors or even lead to a head crash [13]. In order to meet the performance demands of data intensive applications, drive manufacturers are starting to design disk drives that operate below the envelope under moderate I/O loads but could exceed this threshold under certain patterns of heavy usage. Two different disks have been reported to already operate above the thermal

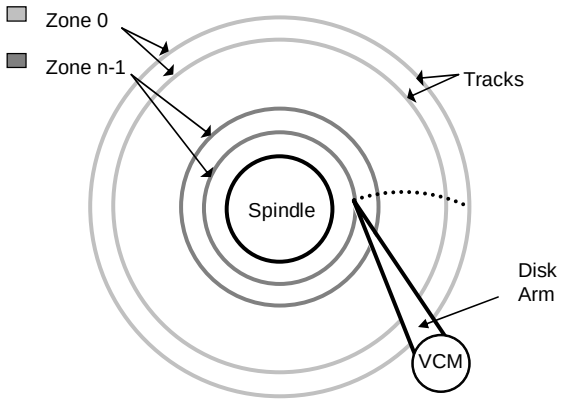

Figure 1. Top view of Disk Platter

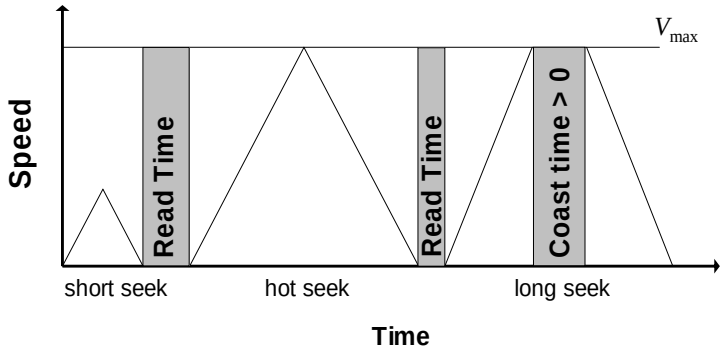

Figure 2. Short, Hot, and Long Seeks

envelope when exercised by I/O benchmarks [20]. Furthermore, disks are usually housed in a tower or rack-mount unit where they are in close proximity to other heat producing components like the processor or even other disks, further increasing their temperature.

2.2 Hot Seeks

The temperature at which a disk drive operates depends on a combination of the external ambient temperature and workload factors. The workload factor that affects temperature is the nature of the disk seeks. A seek occurs when an I/O request is made to read or write a block of data, and the block is not on the track that is currently under the head. During the seek operation, the Voice Coil Motor (VCM) assembly powers the disk arm on and off to reach the desired destination on the disk. The power consumed by the VCM is dissipated as heat. When the VCM is powered on to accelerate or decelerate the disk arm, power is consumed. Otherwise, the VCM is off and the disk cools down. The drive's temperature will be maximized if the VCM is powered on as much as possible.

We model seeks with a Bang-Bang triangular model [15], where the acceleration time equals the deceleration time. The physical actions of a seek are an acceleration phase where the VCM is powered on and accelerates the disk arm, a possible coasting period with the VCM off, and a deceleration phase when the VCM current is reversed for a braking effect. The coast phase only occurs for seeks where the maximum velocity (V_{max}) is reached. Immediately after a seek operation, the disk head must settle over the track to read/write the data (head settle time), and the may have to wait for the requested block to reach the head's location (rotational latency).

We categorize seeks as short, hot, and long seeks, as shown in Figure 2. During a *short seek*, the disk arm accelerates and decelerates without hitting the maximum velocity of the disk arm, V_{max}. For a *long seek*, the VCM is turned off during the coast period of time when the disk arm travels at V_{max} speed before decelerating to the next disk block. The hot seek generates the most heat, since the arm is powered on for the maximum amount of time without any coast time.

2.3 Dynamic Thermal Management

In the past, drives could never reach the thermal envelope even with the worst-case I/O traffic. However, this is expected to change due to several problems associated with magnetic recording density and mechanics, whereby an aggressive scaling up of the drive RPM would be required to get high data transfer rates from the disks [10]. This would require a relaxing of the thermal design constraints, whereby the disks would operate under the thermal envelope under average-case I/O conditions rather than the worst-case.

In order to avoid exceeding the thermal envelope at run-time, the disk would have to be provisioned with Dynamic Thermal Management (DTM) mechanisms. Current systems already provide DTM for disks at the software-level [12]. One proposed DTM mechanism is *throttling* – shutting down the drive when it exceeds a set temperature threshold for a period of time to allow the drive to cool. Throttling has two main drawbacks: slower performance and vulnerability to DoS attacks. This work focuses on attack issues in a drive with throttling.

A DoS attack can be performed on a disk drive if an attacker can consistently raise the temperature to the throttling point by requesting a series of worst-case I/O requests. This attack is created by generating a sequence of hot seeks with minimal idleness between consecutive seek operations. The disk would then ramp up to the designated throttling temperature where it shuts down to cool for a small period of time. After cooling to a safe temperature, which we call the *thermal safety* temperature, the disk resumes servicing more malicious I/O making the disk quickly throttle again. The attack

continues while the disk continues to alter between throttling and servicing attack requests. In Section 4, we present results showing such a sequence of requests can be constructed, even by a remote attacker. First, we describe our experimental setup.

3 Experimental Setup and Workloads

The simulator that we use captures the dynamic behavior of the storage system when servicing I/O requests from a workload. The simulator consists of a performance model and a thermal model. The performance model is based on the widely used Disksim simulator [7]. Disksim simulates all activities in the storage system that could affect performance such as disk access latencies, interconnect latencies, caches, mapping of logical blocks to physical disk sectors, RAID configurations etc. The thermal model is an extension of the work done by Eibeck et al. [6]. This model evaluates the temperature distribution of the disk drive by calculating the amount of heat that is generated by the spindle and voice-coil motors, the viscous dissipation due to the spinning platters and also the associated conduction and convection phenomena. The heat equations (which are differential equations) are solved using the finite difference method. Both these models are fairly detailed and have been validated. Details regarding the design of this simulator are given in [17].

We simulated the attack on a recently introduced 500 GB Maxtor Maxline Pro hard disk drive [18]. We assumed an ambient temperature of 33.4°C, which is the temperature a drive would experience in a tower machine with one Pentium 4 processor and 1GB of RAM operating in an air-conditioned machine room at 25°C [5]. In practice, many drives operate in less ideal conditions, especially in server clusters common in grid computing.

With the hot seek time measurement, we can construct a workload of hot seeks to mount an attack. In Figure 3, the hottest temperature is reached when we use hot seeks. Each point on the graph represents the maximum temperature the drive reached after repeated requests of the seek time on the horizontal access. When seeks are not hot seeks, the drive runs cooler either because the VCM is not powered on as long as it is powered on during a hot seek, or the coast time is non-zero.

Seeks that are longer than a hot seek have a non-zero coast time. These longer seeks allow the disk time to cool. This graph shows that throttling can be caused by seeks close to a hot seek, but a hot seek is the worst kind of I/O traffic.

By generating a worst-case scenario of I/O, we can push

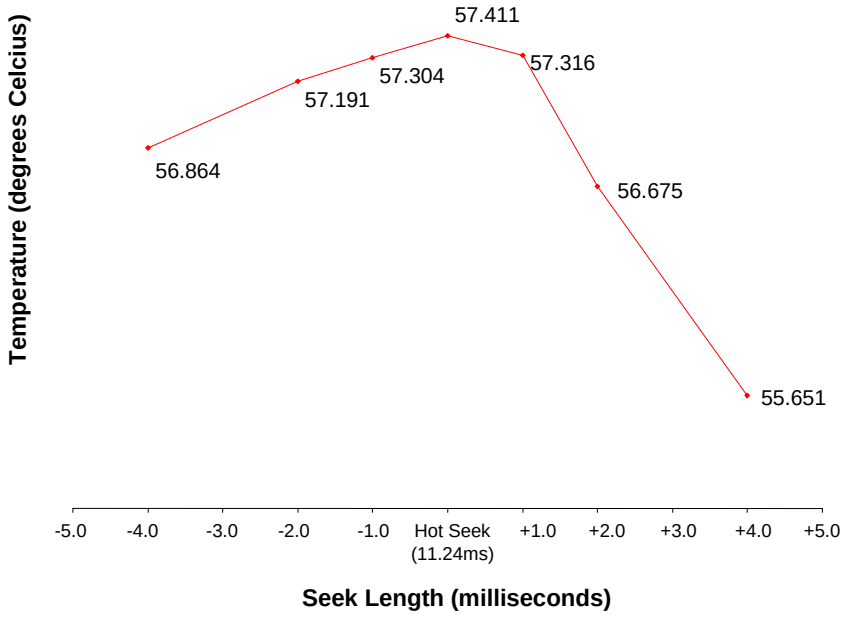

Figure 3. Seek Length and Temperature. The plotted temperature is the measured temperature after simulating a sequence of 200,000 consecutive seeks of the length given on the horizontal axis.

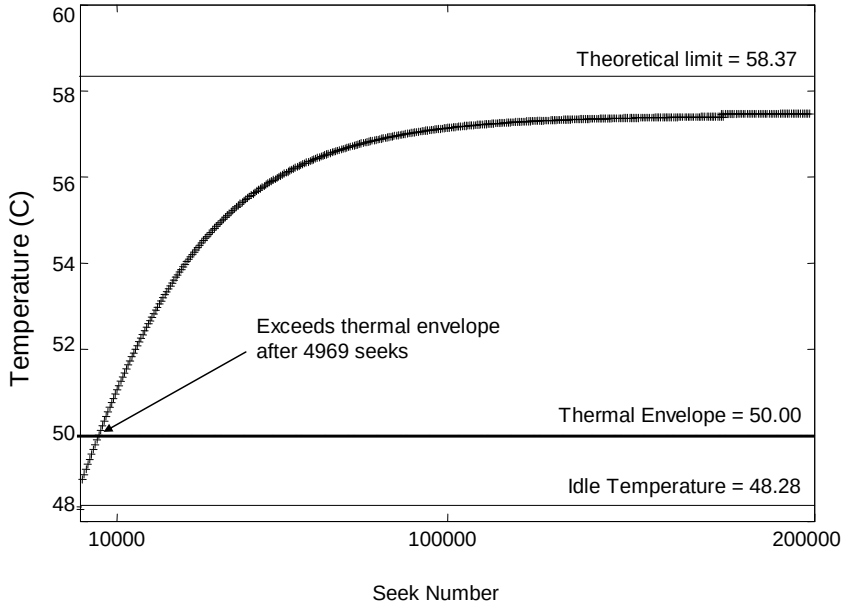

Figure 4. Thermal Attack.

drives past the thermal envelope creating a DoS attack. The worst-case is where all seeks are the hottest possible seek. Combining these worst-case I/O requests we can exceed the thermal envelope.

Our attack will then be caused by the throttling that is designed to protect the drive causing a denial-of-service on the drive as the thermal envelope is repeatedly exceeded in the workload. When a drive throttles, all pending I/O requests to it are delayed until it has sufficiently cooled down. The attacker's goal is to maximize the heat in order to suspend service by causing the drive to throttle. This throttling delays all pending I/O requests leading to a denial-of-service attack.

4 Throttling Attacks

We first consider an attacker who knows all aspects of the disk under attack who is able to control all operations of the machine. The attacker knows the physical characteristics of the drive including the time for a hot seek and the exact physical characteristics of the drive. We also assume the attacker can control any configuration issues in the disk or operating system. This constitutes a powerful attacker, with root-level access to the victim machine. Later, we relax these assumptions and consider attack targets that make it feasible for an attacker with less control. One such target, for example, is a grid network that provides untrusted external users with unprecedented access to the disk by allowing them a more direct way of issuing read and write requests.

4.1 Generating Throttling

The attacker can generate the worst-case I/O by making specific requests that generate the most heat in a disk. The attacker has to the ability to request a specific block from the disk, and they can make the attack more likely to succeed by changing the disk configuration (scheduling algorithm, turn off read-ahead cache). The disk scheduling algorithm can be turned off by a firmware update to the disk. Once the read-ahead cache is turned off and the scheduling algorithms in the OS and the disk are set to FCFS, the attacker can make continuous hot seek requests to throttle the disk. A root-level attacker will be able to do this, but a typical remote attack will need to construct requests that generate nearly worst-case behavior with the given scheduler and competing traffic. We explore how to overcome those difficulties in Section 4.2.

The read-ahead cache can be disabled, but the disk cache may still buffer the specific blocks that are requested. To bypass the cache, an attacker can request different blocks until the set of blocks requested is larger than the disk cache. This obstacle is easily overcome. The attacker just needs to find enough hot seeks that will not entirely fit into the disk cache. For current drives the cache is a maximum of 8-16MB. One characteristic that will make this easier is that a hot seek between two blocks yields the attacker many more hot seeks. Since the read-ahead cache can be disabled, the attacker can request different blocks along the same two tracks that a hot seek was previously found. Other tracks near one of the two blocks in a hot seek can also be used to form a slightly longer or shorter seek. As Figure 3 shows, a seek time that is close to the hot seek still yields a temperature above the throttle temperature.

Figure 4 illustrates the feasibility of the thermal attack under ideal (for the attacker) conditions. Executing a sequence of hot seeks elevates the temperature of the drive over 7°C above its thermal envelope of 50°C. Starting from a normal operating temperature (drive has been warmed up and running), it requires less than 5000

hot seeks to exceed the thermal envelope temperature. By contrast, normal I/O typically stays between the idle temperature and the thermal envelope. Most workloads we tested averaged 49° C.

A disk that uses DTM throttling should be able to mitigate traffic that would raise the temperature above the thermal envelope. Figure 5 depicts an attack on a disk that has DTM enabled. At the throttling temperature (the thermal envelope in this instance), the disk stopped servicing requests for 250 seconds. We chose this delay interval, since this is the lowest amount of time that cools the disk to a normal operating level. A 1 second delay interval cooled the disk to 49.42°C, and a 5 second delay cooled the disk to 49.40°C. The cooling rate decreased as the amount of delay increased.

In this experiment the disk reached the thermal envelope again in just 20 seconds with 1623 requests compared to the first thermal envelope violation of 4969 requests in 62 seconds. Thus the attacker needed much less time to restart an attack.

Some type of protection mechanism is necessary if we want to extend the long-term reliability of this drive. The feasibility of an attack by an attacker with full control and knowledge is possible. Other types of attack where an attacker has less control or may know less information presents more problems that we now discuss in the following.

4.2 Realistic Attacks

The results in Section 4.1 assumed an attacker with absolute control over the victim host. An attacker with such power, of course, can carry out a more damaging attach than just delaying I/O requests. Here, we relax the attacker's control over and knowledge of the victim machine. In this new scenario, the attacker cannot control exactly when the malicious I/O will be serviced. Timing is critical to the success of the attack, and there are several issues that can affect timing: different seek times, larger access sizes, caches, and network latency.

Caching. Caching can be handled in the same manner it was before even if read-ahead is not disabled. Most disks read ahead 64 blocks. The attacker could assume 128 to be safe. Because of the large size of hard drives and the small amount of blocks in a read-ahead, we again have many blocks to choose for the hot seek. Hence, this is not a problem.

Schedulers. The OS and disk schedulers can be handled in a different manner. A scheduler works by reordering pending requests. If the queue of requests has only one pending request, then the scheduler will behave as a

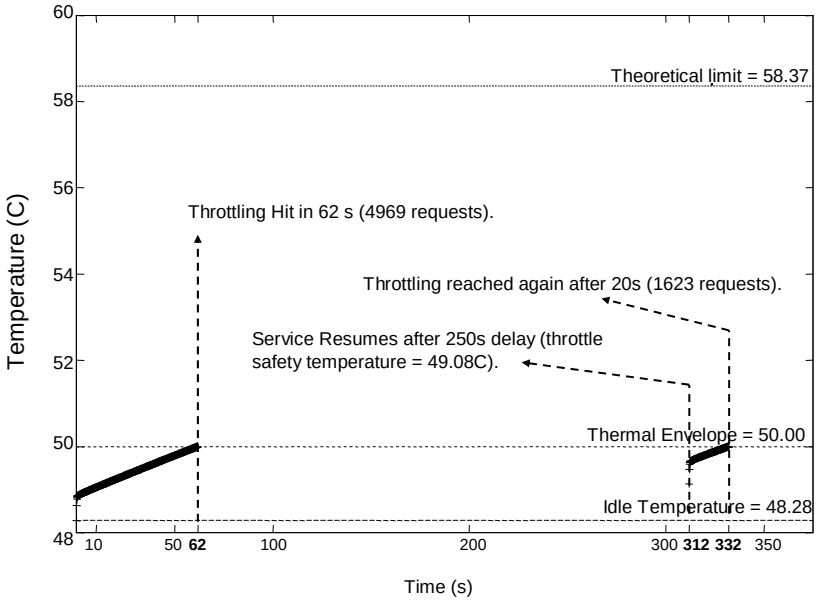

Figure 5. Attack on DTM Throttling. This shows how long it takes to exceed the thermal envelope and throttle (throttling temperature = thermal envelope here). The second throttling happens in just 20 seconds instead of 62 seconds.

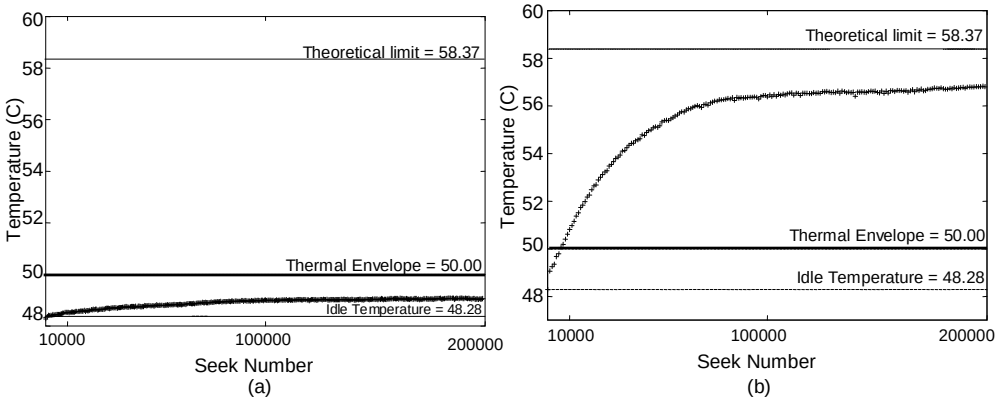

Figure 6. (a) Non-Malicious Traffic (Web Server) [24**]**

(b) Merged Attack Traffic and Non-Malicious Traffic

FCFS scheduler. The attacker will need to submit requests that will force this behavior. If the attacker knows the disk specifications (we assume she does), and she knows specifics about the machine on which the attack traffic is running, then she can submit each malicious request to be serviced before the next request forcing FCFS.

Network Latency. Network latency makes it more difficult to request specific blocks and perform precise timings for an attacker. The attacker's interface to a remote machine will most likely be some type of server (e.g. a web server). The interface will not provide a low-level form of access to the disk. However, some interfaces may provide an SQL interface, and many web servers have a database as a back-end. An SQL query could possibly be constructed that would perform a DoS attack. The query would need to access the disk by issuing many hot seek requests. Another possible interface is an NFS file server, but the attacker may need to exploit a client machine to have access. To bypass timing issues, the SQL query or NFS access could be made in a single request in the client causing many disk accesses on the server.

Rotational Delay. Rotational delay is the amount of time the disk head waits over a specific track for the desired block to rotate to the position where it can be read. This delay gives the disk time to cool. This can be thought of as a seek that is not a hot seek. Although the seeks will not be exact, the temperature can still be elevated past the throttling temperature if the seeks are close enough (see Figure 3). The attacker can also apply known algorithms [23, 25] to find the rotational delay if necessary.

Competing Traffic. Traffic from other processes would most likely cause seeks other than a hot seek to occur. If the throttling point is at the disk's thermal envelope, then the attack is still successful. We ran several experiments with intermingled traffic, and the attack was again successful although the temperature does drop with heavy non-malicious I/O activity.

One such example of competing traffic was a web server trace [24]. By itself, the web server I/O never comes close to the thermal envelope (Figure 6 (a)), but the thermal envelope is violated when combined with an attack trace. The resulting temperature at the end of the trace was a little over one-half degree less than the attack trace by itself (Figure 4). The temperature of the trace is less than the attack trace by itself, but the final temperature is still almost 7°C greater than the thermal envelope. The attack would still be very successful.

Disk Layout. When an attacker cannot request blocks directly via a network interface (e.g. NFS, web server), the number of blocks on a disk available to an attacker would be much less. This can affect requests by forcing the attacker to choose seeks that are not hot seeks and seeks that request more than 1 block. We performed experiments transferring different-sized requests from 1 block to 128 blocks, and the effect was nominal.

5 Eliminating Throttling

Our experiments using the hot seek traces and other normal traffic indicate that an attacker can exploit throttling to deny service to a machine for an extended period of time, but in fact throttling may not be needed at all in practice. Since throttling is such a big performance cost, the drive may be able to endure operating in higher temperatures without failure when it is new. Here, we consider two possible alternatives: just allow the drive to run above the trigger temperature, and employing variable speed disks to perform dynamic thermal management without throttling.

5.1 Exceeding the Thermal Threshold

The simplest countermeasure to the denial-of-service vulnerability throttling introduces is to eliminate throttling and just allow the drive to run hot. Our preliminary experiments indicate that this may be a safe solution for current drives.

We performed an experiment that shows throttling may not even be needed. For our experiment we ran a 500 GB ATA drive in a temperature controlled oven running an intensive I/O workload. Previously reported failure rates by Seagate indicate we should not expect any errors [3].

Current disks have Self-Monitoring, Analysis and Reporting Technology (commonly known as S.M.A.R.T. [22]) information that reports on the drive's status. We can sample this S.M.A.R.T. information while running under a hotter temperature. If the S.M.A.R.T. information reports a problem, then the drive should not be run at that temperature. As the experiment progressed, we recorded the S.M.A.R.T. information to a networked hard drive in case of failure.

The drive itself was sitting inside the oven while the power and IDE cables were attached to the computer itself on the outside. The drive ran in the oven for about 30 hours. During the first 6 hours we gradually increased the temperature of the drive from 25°C (a normal room temperature) to 68°C. After the first 6 hours the drive was running between 80 and 81°C, far hotter than can be achieved by even the ideal attack. The workload bypassed the OS cache using raw I/O, and the disk cache was bypassed by requesting random blocks every other I/O request. Every other third request was a write.

At the end of the experiment the drive showed no errors. Although our experiment only demonstrated that a single drive is not unreliable when operating at high temperatures, it provides support for the argument that operating at high temperature is a long term reliability issue, and that a younger disk can run hotter than the thermal envelope without risking immediate I/O errors. In fact, Seagate figures indicate a drive's reliability under these temperatures would fall to approximately 35,000 hours instead of 200,000 hours [3].

5.2 Alternative Defenses

Instead of using throttling for DTM, one could use a variable-rate RPM disk [2, 19]. Although a variable-rate RPM disk would require an architectural change, this avoids the performance and attack issues with throttling. The RPM effect on power is cubic, so this would alleviate the need for throttling to cool the drive.

If the design is forced to use throttling, there are some other practical defenses. One possibility is to do nothing when the throttling temperature is reached. One option is to use a scheduler that takes heat into account and reorders requests accordingly. This would best be implemented in the disk drive scheduler, since the OS does not have a view of the physical layout of the blocks on the drive. If requests are submitted to force the scheduler to behave as an FCFS scheduler, the scheduler or OS would need to deal with this situation.

6 Conclusion

We have shown that disks using throttling for dynamic thermal management are vulnerable to a thermal denial-of-service attack. On the other hand, operating at high temperature affects the long term reliability of a disk instead of short-term errors. Since workloads can be constructed that cause the disk to throttle (performing a DoS attack), we propose using a variable speed RPM disk instead of disk throttling for DTM. If throttling must be used, we recommend using a better scheduler that takes temperature into account, and throttling should only be used on an older drive to lengthen its long-term reliability. A variable RPM speed disk can avoid the drawbacks of throttling altogether.

References

[1] D. Brooks and M. Martonosi. Dynamic Thermal Management for High-Performance Microprocessors. *In Proceedings of the International Symposium on High-Performance Computer Architecture* (HPCA), pages 172-182, January 2001.

[2] E.V. Carrera, E. Pinheiro, and R. Bianchini. Conserving Disk Energy in Network Servers. In *Proceedings of the International Conference on Supercomputing (ICS)*, June 2003.

[3] Estimating Drive Reliability in Desktop Computers and Consumer Electronics Systems. Seagate Personal Storage Group. TP-338.1.

[4] P. Dadvar and K. Skadron. Potential Thermal Security Risks. *In Proceedings of the IEEE Semiconductor Thermal Measurement, Modeling, and Management Symposium (Semi-Therm 21)*, pp. 229-34, Mar. 2005.

[5] Dell Product Configuration Calculator. http://www1.us.dell.com/content/topics/topic.aspx/global/products/pedge/topics/en/config_calculator.

[6] P. Eibeck and D. Cohen, Modeling Thermal Characteristics of a Fixed Disk Drive, *IEEE Tramsactions on Components, Hybrids, and*

Manufacturing Technology, 11(4):566-570, December, 1988.

[7] G. Ganger, B. Worthington, and Y. Patt. *The DiskSim Simulation Environment Version 2.0*. http://www.pdl.cmu.edu/DiskSim/.

[8] S. Govindavajhala and A. Appel. Using Memory Attacks to Attack a Virtual Machine. In *Proceedings of the IEEE Symposium on Security and Privacy*, May 2003.

[9] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke. DRPM: Dynamic Speed Control for Power Management in Server Class Disks. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, pages 169-179, June 2003.

[10] S. Gurumurthi, A. Sivasubramaniam, V. Natarajan. Disk Drive Roadmap from the Thermal Perspective: A Case for Dynamic Thermal Management. In the /Proceedings of the International Symposium on Computer Architecture/, pages 28-49, June 2005.

[11] J. Hasan, A. Jalote, T.N. Vijaykumar, C. Brodley. Heat Stroke: Power-Density-Based Denial of Service in SMT. In *Proceedings of the International Symposium on High-Performance Computer Architecture (HPCA)*, February 2005.

[12] HDDTemperature.http://hddtemp.com.

[13] G. Herbst. IBM's Drive Temperature Indicator Processor (Drive-TIP) Helps Ensure High Drive Reliability. *In IBM Whitepaper*, October 1997.

[14] Hitachi Global Storage Technologies – HDD Technology Overview Charts. http://www.hitachigst.com/hdd/technolo/overview/storagetechchart.html.

[15] H. Ho. Fast Servo Bang-Bang Seek Control. *IEEE Transactions on Magnetics*, 33(6):4522–4527, November 1997.

[16] Y. Kim, S. Gurumurthi, and A. Sivasubramaniam. Understanding the Performance-Temperature Interactions in Disk I/O of Server Workloads. *Penn State Tech Report CSE-05-007*, November 2005.

[17] Y. Kim, S. Gurumurthi, A. Sivasubramaniam. Understanding the Performance-Temperature Interactions in Disk I/O of Server Workloads. *International Symposium on High Performance Computer Architecture* (To Appear), February 2006.

[18] Maxtor Maxline Pro 500, http://www.maxtor.com/_files/maxtor/en_us/documentation/data_sheets/maxline_pro_500_data_sheet_en.pdf.

[19] E. Pinheiro and R. Bianchini. Energy Conservation Techniques for Disk Array-Based Servers. In *Proceedings of the International Conference on Supercomputing*, pages 68-78, June 2004.

[20] P. Schmid. Hitachi's 500GB DeskStar Monster. http://www.tomshardware.com/2005/07/14/hitachi/index.html.

[21] K. Skadron, M.R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-Aware Microarchitecture. In *Proceedings of the International Symposium on Computer Architecture* (ISCA), pages 1-13, June 2003.

[22] Self-Monitoring, Analysis and Reporting Technology (S.M.A.R.T.) specification, SFF-8035i. April 1996.

[23] N. Talagala, R. Arpaci-Dusseau, and David Patterson. *Microbenchmark-based Extraction of Local and Global Disk Characteristics*. CSD-99-1063. University of California, Berkeley, 1999.

[24] UMass Trace Repository. http://traces.cs.umass.edu.

[25] B. Worthington, G. Ganger, Y. Patt, and J. Wilkes. On-line Extraction of SCSI Disk Drive Parameters. *In Proceedings of the 1995 ACM SIGMETRICS Conference*, pages 146-156, May 1995.

[26] Q. Zhu, F.M. David, C. Devraj, Z. Li, Y. Zhou, and P. Cao. Reducing Energy Consumption of Disk Storage Using Power-Aware Cache Management. *In Proceedings of the International Symposium on High-Performance Computer Architecture (HPCA)*, Febuary 2004.

[27] Q. Zhu, A. Shankar, and Y. Zhou. PB-LRU: A Self-Tuning Power Aware Storage Cache Replacement Algorithm for Conserving Disk Energy. In *Proceedings of the International Conference on Supercomputing (ICS)*, June 2004.