Diss. ETH No. 21966
TIK-Schriftenreihe Nr. 144

On the Vulnerability of Network Mechanisms to Sophisticated Attacks

A dissertation submitted to
ETH ZURICH

for the degree of
Doctor of Sciences

presented by

EHUD BEN-PORAT

Master of Science, Tel-Aviv University
in Computer Science
born July 18, 1980
citizen of Israel

accepted on the recommendation of
Prof. Dr. Bernhard Plattner, examiner
Prof. Dr. Anat Bremler-Barr, co-examiner
Prof. Dr. Hanoch Levy, co-examiner
Prof. Dr. Don Towsley, co-examiner

2014

Abstract

The design of computer and communication systems has been based, for decades, on the fundamental assumption that the objective of all users is to *improve their own performance*. In recent years we have witnessed a wave of DDoS attacks threatening the welfare of the internet. These are launched by *malicious* users whose pure incentive is to *degrade the performance of other, innocent, users*. The traditional systems turn out to be quite vulnerable to these attacks.

In this dissertation we address this problem. First we offer a vulnerability metric to address the main reason security is not receiving the attention it deserves (in the context of performance evaluation) - the lack of metric that measures it. Then, we expose vulnerabilities in various systems and discuss them in detail. Finally we offer a high level approach to categorize and understand the causes different types of vulnerabilities in different systems.

In a *first* contribution, we propose a metric that evaluates the vulnerability of a system. We then evaluate a commonly used data structure in network mechanisms – the hash data structure – using our vulnerability metric. We show that a Closed Hash is much more vulnerable to DDoS attacks than an Open Hash, even though the two systems are considered to be equivalent in terms of traditional performance evaluation. We also apply the metric to queueing mechanisms common to computer and communications systems. Lastly we apply it to the practical case of a hash table whose requests are controlled by a queue, showing that even after the attack has ended, the regular users still suffer from performance degradation or even a total denial of service.

In a *second* contribution, we analyze the vulnerability of the Cumulative Distribution Function (CDF) scheduler, designed for modern high speed cellular networks. We provide an effective mechanism for utilizing the channel

data rate for improving throughput performance in wireless data networks by exploiting channel fluctuations. Considering a *single user* we show that no such user can increase his/her channel share by misreporting the channel capacity. In contrast, considering a *group of users*, we present a scheme by which coordination allows them to *gain permanent increase* in both their time slots share and in their throughput on the expense of others, by misreporting their rates. We show that for large populations consisting of regular and coordinated users in equal numbers, the ratio of allocated time slots between a coordinated and a regular user converges to $e - 1 \approx 1.7$. Our scheme targets the very fundamental principle of the CDF scheduler (as opposed to just attacking implementation aspects), which bases its scheduling decisions on the CDF of the channel rates reported by users. Finally, we outline a modified CDF scheduler immune to such attacks.

In a *third* contribution, we examine the Proportional Fairness Scheduler (PFS) - a popular scheduler designed for modern high speed cellular networks. However, this time we focus on the retransmission mechanism of the scheduler. We show that the common straightforward adaptations of PFS to frame losses expose the system to a malicious attack (which can alternatively be caused by malfunctioning user equipment) that can drastically degrade the performance of innocent users. We analyze the factors behind the vulnerability of the system and propose a modification of PFS designed for the frame loss model which is resilient to such malicious attack while maintaining the fairness properties of the original PFS scheduler.

In a *fourth* contribution, we examine DRFQ, a multi-resource scheduler for middleboxes. Modern networks include an increasing number of middleboxes, designed to perform complex tasks on different types of data flows. Such tasks include VPN, IDS, firewalling and so on. Processing a packet may require different amounts of different hardware resources (such as memory, CPU and bandwidth). The challenge we address in this work is how to fairly schedule packets with different requirements in a multi-resource system. A recent work was the first to address this challenge and proposed a scheduler based on Dominant Resource Fairness (DRF). We show that the proposed solution can favor some flows over others in a way that increases neither the efficiency of the system nor its fairness. Moreover, we show that by cooperating, selfish flows can increase their resource share on the expense of other flows. We propose a scheduler that avoids these vulnerabilities by taking into account the past contribution of flows.

In a *fifth* contribution, we analyze and compare the vulnerability of Pea-

cock and Cuckoo hashing schemes – two of the most studied implementations for hardware network systems (such as NIDS, Firewalls, etc.). We evaluate their vulnerability to sophisticated Denial of Service (DoS) attacks. We show that an attacker can use insertion of carefully selected keys to hit the Peacock and Cuckoo hashing schemes at their weakest points. For the Peacock Hashing, we show that after the attacker fills up only a fraction (typically $5\% - 10\%$) of the buckets, the table completely loses its ability to handle collisions, causing the discard rate (of new keys) to increase dramatically ($100 - 1,800$ times higher). For the Cuckoo Hashing, we show an attack that can impose on the system an excessive number of memory accesses and degrade its performance. We analyze the vulnerability of the system as a function of the critical parameters and provide simulations results as well.

Our objective in the *sixth* contribution, is to understand how system performance is affected by malicious behavior and how performance evaluation should account for it. We do so by considering an array of "classical" systems taken from the literature and examining their degree of vulnerability. These can also serve in providing some initial insights into what makes a system design vulnerable or resilient.

Finally, we conclude this work by providing guidelines for dectectng vulneabilities in existing systems, how to avoid them in the first place and discuss future work.

Kurzfassung

Die Entwicklung von Computer- und Kommunikationssystemen basierte während Jahrzehnten auf der zentralen Annahme, dass das Ziel der Nutzer die Verbesserung ihrer eigenen Leistung sei. In den letzten Jahren erlebten wir eine Welle von DDoS-Attacken, welche die Errungenschaften des Internets nachhaltig gefährdeten. Die Angriffe wurden von böswilligen Nutzern durchgeführt, deren einziges Ziel es war, die Leistung von anderen, unschuldigen Internetbenutzern einzuschränken. Traditionelle Systeme stellten sich gegenüber solchen Attacken dabei als verletzlich heraus.

In der vorliegenden Dissertation verfolgen wir das Ziel, einen Weg hin zur Lösung des skizzierten Problems aufzuzeigen. Dafür entwickeln wir als Erstes eine Metrik für die Messung der Verwundbarkeit eines Systems. Die bisherige Inexistenz einer solchen Metrik streicht heraus, dass Sicherheit bis anhin nicht die Aufmerksamkeit erhalten hat, welche sie verdient. In einem weiteren Schritt zeigen wir die Verwundbarkeit vieler Systeme auf und besprechen diese detailliert. Abschliessend präsentieren wir einen konzeptionellen/generischen Ansatz, der es erlaubt, die Gründe für die unterschiedlichen Arten der Verwundbarkeit der verschiedenen Systeme zu kategorisieren und zu erklären.

In einem ersten Beitrag schlagen wir eine Metrik vor, welche die Verwundbarkeit eines Systems evaluiert. Sodann untersuchen wir eine in Netzwerksystemen weit verbreitete Datenstruktur, die Hashtabelle, indem wir unsere Metrik der Verwundbarkeit anwenden. Wir zeigen, dass geschlossenes Hashing wesentlich verwundbarer ist gegenüber DDoS-Attacken als ein offenes Hashing, obwohl bis anhin gemeinhin galt, dass bei einer traditionellen Evaluation ihrer Leistung die beiden Systeme äquivalent seien. Zusätzlich wenden wir unsere Metrik auf Warteschlangen, einen weit verbreiteten Mechanismus in Computer- und Kommunikationssystemen,

an. Schliesslich folgt eine praktische Anwendung: Wir untersuchen, wie sich eine Hashtabelle bei Anfragen verhält, die in einer Warteschlange zwischengespeichert werden. Wir zeigen, dass gewöhnliche Nutzer auch nach Abschluss eines Angriffs noch unter Leistungseinschränkungen oder einem gänzlichen Ausfall des Systems leiden.

In einem zweiten Beitrag untersuchen wir Cumulative Distribution Function (CDF) Scheduler, welcher für die Anwendung in modernen Mobilfunksystemen entwickelt wurde. Dieser Scheduler verwendet einen vermeintlich wirksamen Algorithmus für die Optimierung der Allokation der zur Verfügung stehenden Übertragungskapazität auf die verschiedenen Nutzer, beruhend auf der zeitlichen Variabilität der benötigten Übertragungsraten. Der Algorithmus verwendet Angaben der einzelnen Nutzer über den von ihnen gemessenen Durchsatz, um eine optimale Allokation zu erreichen.

Wir zeigen auf, dass ein *alleine handelnder Nutzer* keinen Vorteil erhalten kann, indem er falsche Angaben über den erhaltenen Durchsatz macht. Hingegen ist es *mehreren, sich absprechenden Nutzern* möglich, einen Vorteil aus Falschangaben zu ziehen. Der daduch erhaltene Vorteil ist nicht nur momentan, sondern verbleibt auch nach dem Abschluss des Angriffs, d.h. dieser hat eine stabile und permanente Wirkung. Wir zeigen auf, dass die Komplizen eines koordinierten Angriffs gegenüber den nicht am Angriff teilnehmenden Nutzern einen um den Faktor 1.7 erhöhten Durchsatz erreichen können.

Unser postulierter Angriff auf den CDF Scheduler setzt bei einem fundamentalen Prinzim des Schedulers an, d.h. ist nicht abhängig von einer allfälligen fehlerhaften Implementation de Schedulers.

Wir schliessen unsere Untersuchung ab, indem wir einen modifizierten Scheduler entwickeln, der auch koordinierten Angriffen widersteht.

In einem dritten Beitrag prüfen wir den Proportional Fairness Scheduler (PFS), der in heutigen 4G-Mobilfunksystemen eingesetzt wird. Wir konzentrieren uns auf den im Scheduler verwendeten Mechanismus, welcher bestimmt, wann fehlerhaft übertragene Daten erneut für die Ubertragung eingeplant werden. Wir zeigen auf, dass die derzeit übliche, “naive” Realisierung des Schedulers böswilligen Attacken ausgesetzt ist, die zu Störungen im Betrieb führen. Diese können eine drastische Verringerung des Durchsatzes für unschuldige Nutzer zur Folge haben. Wir analysieren die verschiedenen Faktoren, die hinter der Verwundbarkeit des Systems stehen und schlagen eine modifizierte Variante von PFS vor, welche fehlerhafte Übertragungen korrekt berücksichtigt und somit derartigen Attacken widerstehen kann, ohne

die Fairness-Eigenschaften von PFS zu verändern.

In einem vierten Beitrag prüfen wir den DRFW, einen Multi-Resource-Scheduler für Middleboxen. Moderne Netzwerke beinhalten eine steigende Anzahl an Middleboxen, welche komplexe Aufgaben auf verschiedene Arten von Datenströmen verrichten. Dazu zählen unter anderem VPN, IDS sowie Firewalls. Diese Aufgaben können abhängig vom verarbeiteten Datenpaket unterschiedliche Hardwareresourcen (wie z.B. Hauptspeicher, Prozessor, Bandbreite) beanspruchen. Wir widmen uns der Frage, wie eine Verarbeitung von Paketen, die verschiedenarige Anforderungen an das Multi-Resource-System stellt, geplant werden kann, so dass die geforderte Fairness nach wie vor garantiert ist. Eine kürzlich erschienene Arbeit hat sich erstmals mit dieser Herausforderung auseinandergesetzt und einen Scheduler vorgeschlagen, welcher auf Dominant Resource Fairness (DRF) basiert. Wir zeigen, dass die vorgeschlagene Lösung gewisse Datenströme gegenüber anderen bevorzugen kann, wobei weder die Effizienz des Systems noch die Fairness gesteigert wird. Zudem zeigen wir, dass durch "eigennützige" Datenströme mittels einer geeigneten Kooperation zusätzliche Ressourcen auf Kosten anderer Datenströme gewinnen können. Wir schlagen deshalb einen Scheduler vor, welcher solche Verwundbarkeiten vermeidet, indem die früheren Beiträge von Datenströmen berücksichtigt werden.

In einem fünften Beitrag analysieren und vergleichen wir die Verwundbarkeit von Peacock- und Cuckoo-Hashing, zwei der meist studierten Hashingmechanismen für Netzwerkhardware, wie z.B. NIDS, Firewalls usw. Wir evaluieren ihre Verwundbarkeit gegenüber Denial of Service (DoS) Attacken, welche auf der inhärenten Komplexität der eingesetzten Algorithmen beruhen. Wir zeigen, dass ein Angreifer sorgfältig gewählte Schlüssel einschleusen kann, welche Peacock- and Cuckoo-Hashing in ihrem verwundbarsten Punkt treffen. Für das Peacock-Hashing zeigen wir, dass, nachdem ein Angreifer nur einen Bruchteil (üblicherweise $5\% - 10\%$) der Plätze besetzt hat, die Hashtabelle vollständig ihre Fähigkeit verliert, mit Kollisionen umzugehen, was eine drastische Erhöhung ($100 - 1,800$ Mal höher) der Ausschlussrate (für neue Schlüssel) zur Folge hat. Für das Cuckoo-Hashing zeigen wir, dass eine Attacke dem System eine übermässige Anzahl an Speicherzugriffe aufdrängen und damit die Kapazität beeinträchtigen kann. Wir analysieren und simulieren die Verwundbarkeit des Systems gegenüber verschiedenen kritischen Parametern.

In einem sechsten Beitrag untersuchen wir, wie das Verhalten eines Systems (und insbesondere dessen Kapazität) durch böswilliges Verhalten

der Nutzer beinflusst werden kann. Wir entwickeln eine Methodik für die Überprüfung komplexer Systeme im Hinblick auf ihre Verwundbarkeit. Wir erläutern unsere Methodik anhand verschiedener klassischer Systeme aus der Literatur und bestimmen den Grad der Verwundbarkeit der betrachteten Systeme. Dies ermöglicht uns, zu verstehen, welche Elemente des Entwurfs ein System verwundbar oder widerstandsfähig machen können.