IEEE distributed systems ONLINE
Expert-authored articles and resources

IEEE DISTRIBUTED SYSTEMS ONLINE 1541-4922 © 2005 Published by the IEEE Computer Society
Vol. 6, No. 10; October 2005

Parallel and Distributed Computing for Cybersecurity

Vipin Kumar, *University of Minnesota*

Parallel and distributed data mining offer great promise for addressing cybersecurity. The Minnesota Intrusion Detection System can detect sophisticated cyberattacks on large-scale networks that are hard to detect using signature-based systems.

This article is based on the author's keynote talk (ppt) (http://www.ieee.org/netstorage/computer_society/dsonline_media/Kumar-PDCS2004/Kumar-KeynoteLecture-PDCS2004.ppt) at the 2004 International Conference on Parallel and Distributed Computing and Systems (PDCS 04).

The phenomenal growth in computing power over much of the past five decades has been motivated by scientific applications requiring massive amounts of computation. But lately a major focus for parallel and high-performance computers has been on data-centric applications in which the application's overall complexity is driven by the data's size and nature. Data mining is one of these data-centric applications that increasingly drives development of parallel and distributed computing technology.

Explosive growth in the availability of various kinds of data in both commercial and scientific domains has resulted in an unprecedented opportunity to develop automated data-driven knowledge discovery techniques. Data mining, an important step in this knowledge-discovery process, consists of methods that discover interesting, nontrivial, useful patterns hidden in the data.[1,2]

The huge size and high dimensionality of available data sets make large-scale data mining applications computationally demanding, so much so that high-performance parallel computing is fast becoming an essential component of the solution. The data tends to be distributed, and issues such as scalability, privacy, and security prohibit bringing the data together. Such cases require distributed data mining.

Into this mix enters the Internet, along with its tremendous benefits and vulnerabilities. The need for cybersecurity and the inadequacy of traditional approaches have piqued interest in applying data mining to intrusion detection. This article focuses on the promise and application of parallel and distributed data mining to cybersecurity.

Need for cybersecurity

Individuals and organizations attack and misuse computer systems, creating new Internet threats daily. The

number of computer attacks has increased exponentially in the past few years,[3] and their severity and sophistication are also growing.[4] For example, when the Slammer/Sapphire Worm began spreading throughout the Internet in early 2003, it doubled in size every 8.5 seconds and infected at least 75,000 hosts.[3] It caused network outages and unforeseen consequences such as cancelled airline flights, interference with elections, and ATM failures.

The conventional approach to securing computer systems is to design mechanisms such as firewalls, authentication tools, and virtual private networks that create a protective shield. However, these mechanisms almost always have vulnerabilities. They can't ward off attacks that are continually being adapted to exploit system weaknesses, which are often caused by careless design and implementation flaws. This has created the need for *intrusion detection*,[5,6] security technology that complements conventional security approaches by monitoring systems and identifying computer attacks.[3]

Traditional intrusion detection methods are based on human experts' extensive knowledge of *attack signatures* (character strings in a message's payload that indicate malicious content). They have several limitations. They can't detect novel attacks, because someone must manually revise the signature database beforehand for each new type of intrusion discovered. And once someone discovers a new attack and develops its signature, deploying that signature is often delayed. These limitations have led to an increasing interest in intrusion detection techniques based on data mining.[5,6]

The Minnesota Intrusion Detection System

The MINDS data-mining-based system (http://www.cs.umn.edu/research/minds) detects unusual network behavior and emerging cyberthreats. It's deployed at the University of Minnesota, where several hundred million network flows are recorded from a network of more than 40,000 computers every day. MINDS is also part of the Interrogator[7] architecture at the US Army Research Lab's Center for Intrusion Monitoring and Protection (ARL-CIMP), where analysts collect and analyze network traffic from dozens of Department of Defense sites.[8] MINDS is enjoying great operational success at both sites, routinely detecting brand new attacks that signature-based systems could not have found. Additionally, it often discovers rogue communication channels and the exfiltration of data that other widely used tools such as Snort (http://www.snort.org) have had difficulty identifying.[8,9]

Figure 1 illustrates the process of analyzing real network traffic data using MINDS . The MINDS suite contains various modules for collecting and analyzing massive amounts of network traffic. Typical analyses include behavioral anomaly detection, summarization, and profiling. Additionally, the system has modules for feature extraction and for filtering out attacks for which good predictive models exist (for example, for scan detection). Independently, each of these modules provides key insights into the network. When combined, which MINDS does automatically, these modules have a multiplicative affect on analysis.

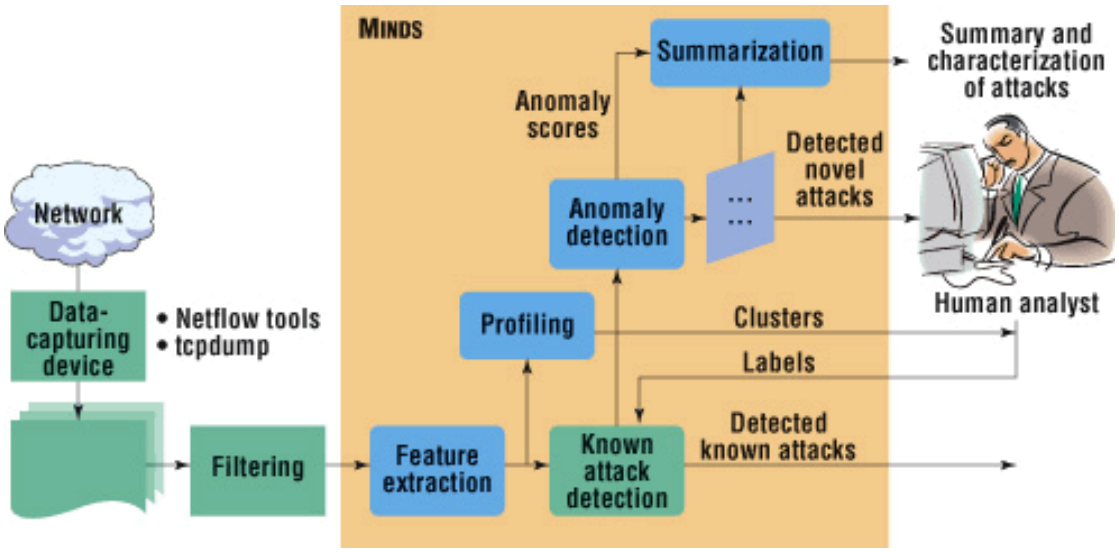


Figure 1. The Minnesota Intrusion Detection System (MINDS).

Anomaly detection

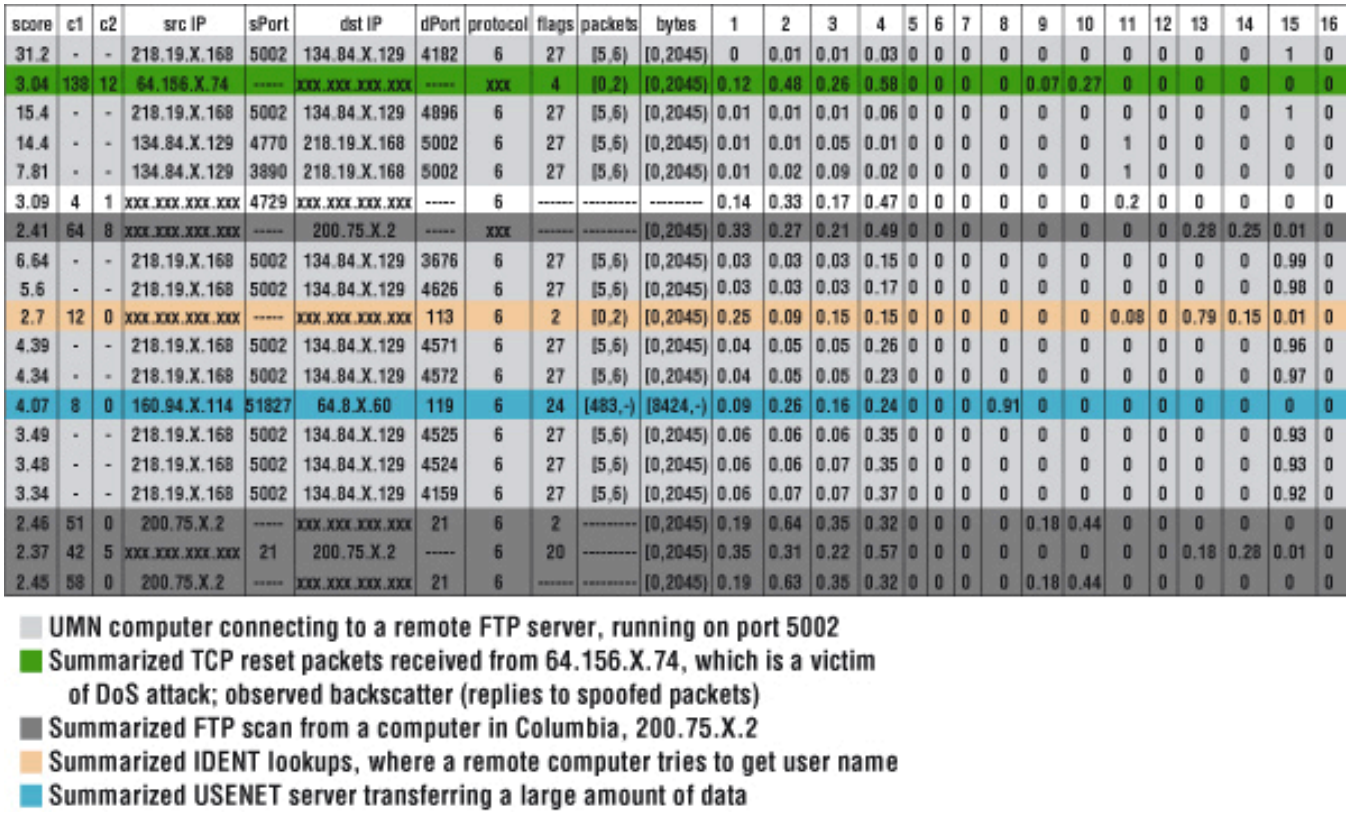
At MINDS ' core is a behavioral-anomaly detection module based on a novel data-driven technique for calculating the distance between points in a high-dimensional space. Notably, this technique enables meaningful calculation of the similarity between records containing a mixture of categorical and numerical attributes (such as network traffic records). Unlike other extensively investigated anomaly detection methods, this new framework doesn't suffer from numerous false alarms. To the best of our knowledge, no other existing anomaly detection technique can find complex behavior anomalies in a real-world environment while maintaining a very low false-alarm rate. A multithreaded parallel formulation of this module allows analysis of network traffic from many sensors in near-real time at the ARL-CIMP.

Summarization

The ability to summarize large amounts of network traffic can be highly valuable for network security analysts who must often deal with large amounts of data. For example, when analysts use the MINDS anomaly detection algorithm to score several million network flows in a typical window of data, several hundred highly ranked flows might require attention. But due to the limited time available, analysts often can look only at the first few pages of results covering the top few dozen most anomalous flows. Because MINDS can summarize many of these flows into a small representation, the analyst can analyze a much larger set of anomalies than is otherwise possible. Our research group has formulated a methodology for summarizing information in a database of transactions with categorical attributes as an optimization problem.[9,10] This methodology uses association pattern analysis originally developed to discover consumer behavior patterns in large sales transaction data sets. These algorithms have helped us better understand the nature of cyberattacks as well as create new signature rules for intrusion detection systems. Specifically, the MINDS summarization component compresses the anomaly detection component's output into a compact representation, so analysts can investigate numerous anomalous activities in a single screenshot.

Figure 2 illustrates a typical MINDS output after anomaly detection and summarization. The system sorts the connections according to the score that the anomaly detection algorithm assigns them. Then, using the patterns

that the association analysis module generates, MINDS summarizes anomalous connections with the highest scores. Each line contains the average anomaly score, the number of connections represented by the line, eight basic connection features, and the relative contribution of each basic and derived anomaly detection feature. For example, the second line in **figure 2** represents 138 anomalous connections. From this summary, analysts can easily infer that this is a backscatter from a denial-of-service attack on a computer that is outside the network being examined. Such inference is hard to make from individual connections even if the anomaly detection module ranks them highly. **Figure 2** shows the analysts' interpretations of several other summaries the system found.



score	c1	c2	src IP	sPort	dst IP	dPort	protocol	flags	packets	bytes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
31.2	-	-	218.19.X.168	5002	134.84.X.129	4182	6	27	[5,6)	[0,2045)	0	0.01	0.01	0.03	0	0	0	0	0	0	0	0	0	0	1	0
3.04	138	12	64.156.X.74	-----	xxx.xxx.xxx.xxx	-----	xxx	4	[0,2)	[0,2045)	0.12	0.48	0.26	0.58	0	0	0	0	0.07	0.27	0	0	0	0	0	0
15.4	-	-	218.19.X.168	5002	134.84.X.129	4896	6	27	[5,6)	[0,2045)	0.01	0.01	0.01	0.06	0	0	0	0	0	0	0	0	0	0	1	0
14.4	-	-	134.84.X.129	4770	218.19.X.168	5002	6	27	[5,6)	[0,2045)	0.01	0.01	0.05	0.01	0	0	0	0	0	0	1	0	0	0	0	0
7.81	-	-	134.84.X.129	3890	218.19.X.168	5002	6	27	[5,6)	[0,2045)	0.01	0.02	0.09	0.02	0	0	0	0	0	0	1	0	0	0	0	0
3.09	4	1	xxx.xxx.xxx.xxx	4729	xxx.xxx.xxx.xxx	-----	6	------	--------	--------	0.14	0.33	0.17	0.47	0	0	0	0	0	0	0.2	0	0	0	0	0
2.41	64	8	xxx.xxx.xxx.xxx	-----	200.75.X.2	-----	xxx	------	--------	[0,2045)	0.33	0.27	0.21	0.49	0	0	0	0	0	0	0	0	0.28	0.25	0.01	0
6.64	-	-	218.19.X.168	5002	134.84.X.129	3676	6	27	[5,6)	[0,2045)	0.03	0.03	0.03	0.15	0	0	0	0	0	0	0	0	0	0	0.99	0
5.6	-	-	218.19.X.168	5002	134.84.X.129	4626	6	27	[5,6)	[0,2045)	0.03	0.03	0.03	0.17	0	0	0	0	0	0	0	0	0	0	0.98	0
2.7	12	0	xxx.xxx.xxx.xxx	-----	xxx.xxx.xxx.xxx	113	6	2	[0,2)	[0,2045)	0.25	0.09	0.15	0.15	0	0	0	0	0	0	0.08	0	0.79	0.15	0.01	0
4.39	-	-	218.19.X.168	5002	134.84.X.129	4571	6	27	[5,6)	[0,2045)	0.04	0.05	0.05	0.26	0	0	0	0	0	0	0	0	0	0	0.96	0
4.34	-	-	218.19.X.168	5002	134.84.X.129	4572	6	27	[5,6)	[0,2045)	0.04	0.05	0.05	0.23	0	0	0	0	0	0	0	0	0	0	0.97	0
4.07	8	0	160.94.X.114	51827	64.8.X.60	119	6	24	[483,-)	[8424,-)	0.09	0.26	0.16	0.24	0	0	0	0.91	0	0	0	0	0	0	0	0
3.49	-	-	218.19.X.168	5002	134.84.X.129	4525	6	27	[5,6)	[0,2045)	0.06	0.06	0.06	0.35	0	0	0	0	0	0	0	0	0	0	0.93	0
3.48	-	-	218.19.X.168	5002	134.84.X.129	4524	6	27	[5,6)	[0,2045)	0.06	0.06	0.07	0.35	0	0	0	0	0	0	0	0	0	0	0.93	0
3.34	-	-	218.19.X.168	5002	134.84.X.129	4159	6	27	[5,6)	[0,2045)	0.06	0.07	0.07	0.37	0	0	0	0	0	0	0	0	0	0	0.92	0
2.46	51	0	200.75.X.2	-----	xxx.xxx.xxx.xxx	21	6	2	--------	[0,2045)	0.19	0.64	0.35	0.32	0	0	0	0	0.18	0.44	0	0	0	0	0	0
2.37	42	5	xxx.xxx.xxx.xxx	21	200.75.X.2	-----	6	20	--------	[0,2045)	0.35	0.31	0.22	0.57	0	0	0	0	0	0	0	0	0.18	0.28	0.01	0
2.45	58	0	200.75.X.2	-----	xxx.xxx.xxx.xxx	21	6	------	--------	[0,2045)	0.19	0.63	0.35	0.32	0	0	0	0	0.18	0.44	0	0	0	0	0	0

Figure 2. Output of the MINDS summarization module. Each line contains an anomaly score, the number of connections that line represents, and several other pieces of information that help the analyst get a quick picture.

Profiling

We can use clustering, a data mining technique for grouping similar items, to find related network connections and thus discover dominant modes of behavior. MINDS uses the Shared Nearest Neighbor clustering algorithm,[11] which works particularly well when data is high-dimensional and noisy (for example, network data). SNN is highly computationally intensive—of the order $O(n^2)$, where n is the number of network connections. So, we need to use parallel computing to scale this algorithm to large data sets. Our group has developed a parallel formulation of the SNN clustering algorithm for behavior modeling, making it feasible to analyze massive amounts of network data.[8]

An experiment we ran on a real network illustrates this approach as well as the computational power required to

run SNN clustering on network data. The data consisted of 850,000 connections collected over one hour. On a 16-CPU cluster, the SNN algorithm took 10 hours to run and required 100 Mbytes of memory at each node to calculate distances between points. The final clustering step required 500 Mbytes of memory at one node. The algorithm produced 3,135 clusters ranging in size from 10 to 500 records. Most large clusters corresponded to normal behavior modes, such as virtual private network traffic. However, several smaller clusters corresponded to minor deviant behavior modes relating to misconfigured computers, insider abuse, and policy violations undetectable by other methods. Such clusters give analysts information they can act on immediately and can help them understand their network traffic behavior. **Figure 3** shows two clusters obtained from this experiment. These clusters represent connections from inside machines to a site called GoToMyPC.com, which allows users (or attackers) to control desktops remotely. This is a policy violation in the organization for which this data was being analyzed.

Start time	Duration	Src IP	Src Port	Dst IP	Dst Port Proto	TTL	Flags	Packets	Bytes
20040407.10:00:10.428036	0:00:00	A	4125	B	8200 tcp	123	***AP*SF	5	248
20040407.10:00:40.685520	0:00:03	A	4127	B	8200 tcp	123	***AP*SF	5	248
20040407.10:00:58.748920	0:00:00	A	4138	B	8200 tcp	123	***AP*SF	5	248
20040407.10:01:44.138057	0:00:00	A	4141	B	8200 tcp	123	***AP*SF	5	248
20040407.10:01:59.267932	0:00:00	A	4143	B	8200 tcp	123	***AP*SF	5	248
20040407.10:02:44.937575	0:00:01	A	4149	B	8200 tcp	123	***AP*SF	5	248
20040407.10:04:00.717395	0:00:00	A	4163	B	8200 tcp	123	***AP*SF	5	248
20040407.10:04:30.976627	0:00:01	A	4172	B	8200 tcp	123	***AP*SF	5	248
20040407.10:04:46.106233	0:00:00	A	4173	B	8200 tcp	123	***AP*SF	5	248
20040407.10:05:46.715539	0:00:00	A	4178	B	8200 tcp	123	***AP*SF	5	248
20040407.10:06:16.975202	0:00:01	A	4180	B	8200 tcp	123	***AP*SF	5	248
20040407.10:06:32.105013	0:00:00	A	4181	B	8200 tcp	123	***AP*SF	5	248

Start time	Duration	Src IP	Src Port	Dst IP	Dst Port Proto	TTL	Flags	Packets	Bytes
20040407.10:00:40.685522	0:00:03	B	8200	A	4127 tcp	123	***AP*SF	4	211
20040407.10:00:58.748922	0:00:00	B	8200	A	4138 tcp	123	***AP*SF	4	211
20040407.10:01:44.138059	0:00:00	B	8200	A	4141 tcp	123	***AP*SF	4	211
20040407.10:02:14.678442	0:00:00	B	8200	A	4145 tcp	123	***AP*SF	4	211
20040407.10:02:44.937577	0:00:01	B	8200	A	4149 tcp	123	***AP*SF	4	211
20040407.10:03:15.308206	0:00:00	B	8200	A	4153 tcp	123	***AP*SF	4	211
20040407.10:04:30.976629	0:00:01	B	8200	A	4172 tcp	123	***AP*SF	4	211
20040407.10:06:16.975204	0:00:01	B	8200	A	4180 tcp	123	***AP*SF	4	211
20040407.10:06:32.105015	0:00:00	B	8200	A	4181 tcp	123	***AP*SF	4	211
20040407.10:06:47.234837	0:00:00	B	8200	A	4182 tcp	123	***AP*SF	4	211
20040407.10:07:02.367471	0:00:00	B	8200	A	4183 tcp	123	***AP*SF	4	211
20040407.10:07:17.494574	0:00:00	B	8200	A	4184 tcp	123	***AP*SF	4	211

Figure 3. Two clusters obtained from network traffic at a US Army base, representing connections with GoToMyPC.com.

Detecting distributed attacks

Interestingly, attacks often arise from multiple locations. In fact, individual attackers often control numerous machines, and they can use different machines to launch different steps of an attack. Moreover, the attack's targets could be distributed across multiple sites. An intrusion detection system (IDS) running at one site might

not have enough information by itself to detect the attack. Rapidly detecting such distributed cyberattacks requires an interconnected system of IDSs that can ingest network traffic data in near real-time, detect anomalous connections, communicate their results to other IDSs, and incorporate the information from other systems to enhance the anomaly scores of such threats. Such a system consists of several autonomous IDSs that share their knowledge bases with each other to swiftly detect malicious, large-scale cyberattacks.

Figure 4 illustrates the distributed aspect of this problem. It shows the two-dimensional global Internet Protocol space such that every IP address allocated in the world is represented in some block. The black region represents unallocated IP space.

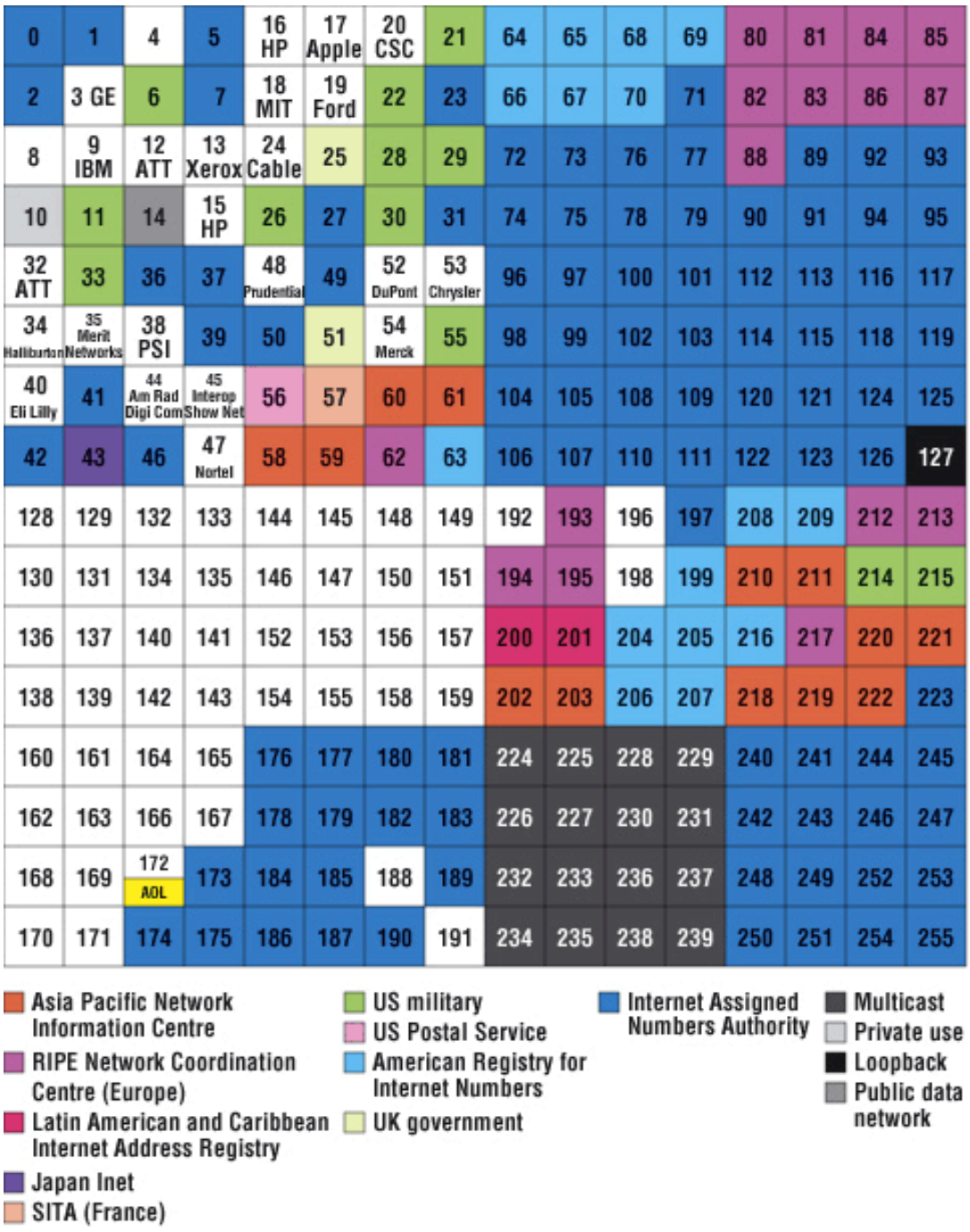


Figure 4. Map of the global IP space.

Figure 5 shows a graphical illustration of suspicious connections originating from the outside (box on the right) to machines inside the University of Minnesota's IP space (box on the left) in a typical time window of 10 minutes. Each red dot in the right-hand box represents a suspicious connection made by a machine to an internal machine on port 80. In this case, it means that the internal machine being contacted doesn't have a Web server running, making the external machines that are trying to connect to port 80 suspected attackers. The right-hand box indicates that most of these potential attackers are clustered in specific Internet address blocks. A close examination shows that most of the dense areas belong to the network blocks of cable and AOL users located in the US or to blocks allocated to Asia and Latin America. There are 999 unique sources on the outside trying to contact 1,126 destinations inside the University of Minnesota IP network space. The total number of involved flows is 1,516, which means that most external sources made just one suspicious connection to the inside. It's hard to tag a source as malicious on the basis of just one connection. If multiple sites running the same analysis across the IP space report the same external source as suspicious, it would make the classification much more accurate.

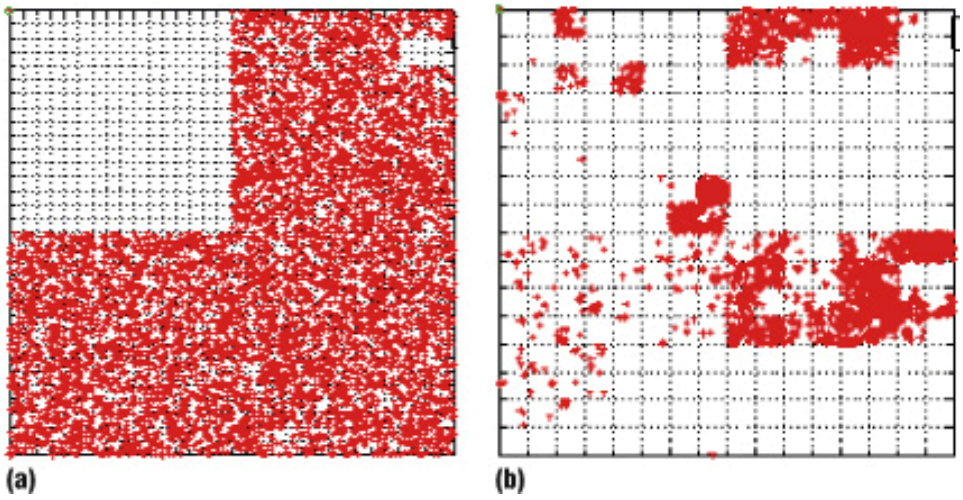


Figure 5. Suspicious traffic on port 80. (a) Destination IP addresses of suspicious connections within the University of Minnesota's three class B networks. (b) Source IPs of suspicious connections in the global IP space.

The ideal scenario for the future would be that we bring the data collected at these different sites to one place and then analyze it. But this isn't feasible because

- the data is naturally distributed and more suited for distributed analysis;
- the cost of merging huge amounts of data and running analysis at one site is very high; and
- privacy, security, and trust issues arise in sharing network data among different organizations.

So, what's really required is a distributed framework in which these different sites can independently analyze their data and then share high-level patterns and results while honoring the individual sites' data privacy. Implementing such a system would require handling distributed data, addressing privacy issues, and using data mining tools, and would be much easier if a middleware provided these functions. The University of Minnesota, University of Florida, and University of Illinois, Chicago, are developing and implementing such a system (see **figure 6**) as part of a US National Science Foundation-funded collaborative project called Data Mining Middleware for the Grid.

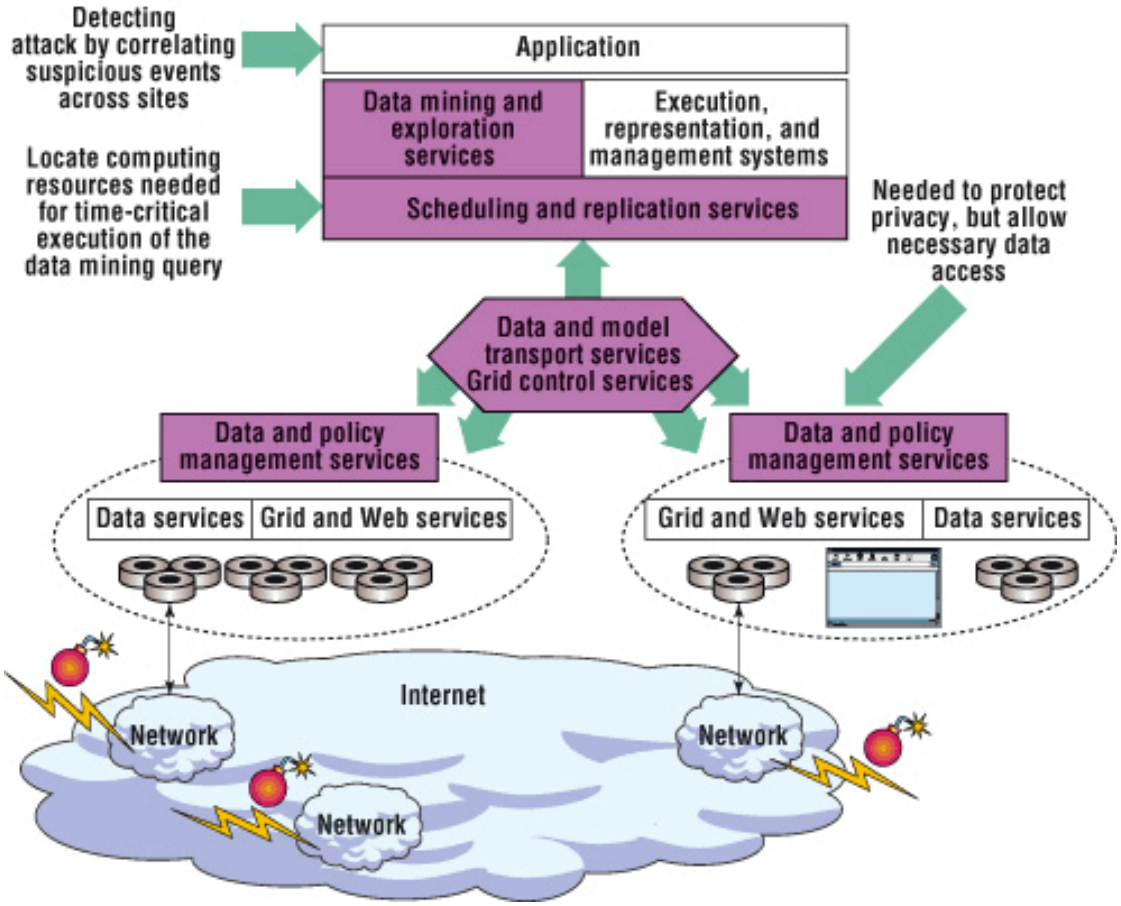


Figure 6. The distributed network intrusion detection system being developed collaboratively by three university teams.

Acknowledgments

This work is supported by ARDA grant AR/F30602-03-C-0243, NSF grants IIS-0308264 and ACI-0325949, and the US Army High Performance Computing Research Center under contract DAAD19-01-2-0014. The research reported in this article was performed in collaboration with Paul Dokas, Eric Eilertson, Levent Ertoz, Aleksandar Lazarevic, Michael Steinbach, George Simon, Mark Shaneck, Haiyang Liu, Jaideep Srivastava, Pang-Ning Tan, Varun Chandola, Yongdae Kim, Zhi-li Zhang, Sanjay Ranka, and Bob Grossman.

I thank Devdatta Kulkarni for his volunteer work on integrating the audio and PowerPoint files.

References

1. U.M. Fayyad et al., *Advances in Knowledge Discovery and Data Mining,* AAAI Press/MIT Press, 1996.
2. P. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining,* Addison-Wesley, 2005.
3. V. Kumar, J. Srivastava, and A. Lazarevic, eds. *Managing Cyber Threats—Issues, Approaches and Challenges,* Springer, 2005.

4. D. Moore et al., The Spread of the Sapphire/Slammer Worm, (http://www.cs.berkeley.edu/~nweaver/sapphire), 2003.
5. D. Barbara and S. Jajodia, eds. *Applications of Data Mining in Computer Security,* Kluwer Academic Publishers, 2002.
6. V. Kumar, J. Srivastava, and A. Lazarevic, "Intrusion Detection—A Survey,"*Managing Cyber Threats —Issues, Approaches, and Challenges,* V. Kumar, J. Srivastava, and A. Lazarevic, eds., Springer, 2005, pp. 19-80.
7. K. Long, "Catching the Cyber Spy: ARL's Interrogator," http://www.asc2004.com/Manuscripts/sessionB/BS-40.pdf, *Proc. 24th Army Science Conf.,* US Army, 2004.
8. E. Eilertson et al., "MINDS: A New Approach to the Information Security Process," http://www.asc2004.com/Manuscripts/sessionB/BP-06.pdf, *Proc. 24th Army Science Conf.,* US Army, 2004.
9. L. Ertoz et al., "MINDS—Minnesota Intrusion Detection System," *Data Mining—Next Generation Challenges and Future Directions,* MIT Press, 2004.
10. V. Chandola and V. Kumar, "Summarization—Compressing Data into an Informative Representation," to be published in *Proc. 5th IEEE Int'l Conf. Data Mining* (ICDM), IEEE CS Press, 2005.
11. L. Ertoz, M. Steinbach, and V. Kumar, "A New Shared Nearest Neighbor Clustering Algorithm and its Applications," *Proc. Workshop on Clustering High Dimensional Data and its Applications, 2nd SIAM Int'l Conf. Data Mining,* Soc. for Industrial and Applied Math., 2002, pp. 105-115.

Vipin Kumar is the William Norris Professor and head of the Computer Science and Engineering Department at the University of Minnesota. His research interests include high-performance computing and data mining. He has coedited or coauthored nine books, including *Introduction to Parallel Computing* (Addison-Wesley, 1994) and *Introduction to Data Mining* (Addison-Wesley, 2005). He received his PhD in computer science from the University of Maryland. He is a fellow of the IEEE, a member of SIAM and the ACM, and a fellow of the Minnesota Supercomputer Institute. Contact him at the Dept. of Computer Science and Eng., 4-192, EE/CSci Bldg., Univ. of Minnesota, Minneapolis, MN 55455; kumar@cs.umn.edu.

Cite this article: Vipin Kumar, "Parallel and Distributed Computing for Cybersecurity," *IEEE Distributed Systems Online*, vol. 6, no. 10, 2005.