Sushil Jajodia
Duminda Wijesekera (Eds.)

Data and Applications Security XIX

LNCS 3654

19th Annual IFIP WG 11.3 Working Conference on
Data and Applications Security
Storrs, CT, USA, August 2005, Proceedings

ifip

Sushil Jajodia Duminda Wijesekera (Eds.)

Data and Applications Security XIX

19th Annual IFIP WG 11.3 Working Conference on
Data and Applications Security
Storrs, CT, USA, August 7-10, 2005
Proceedings

Springer

Volume Editors

Sushil Jajodia
Duminda Wijesekera
George Mason University
Center for Secure Information Systems
Fairfax, VA 22030, USA
E-mail: {jajodia,dwijesek}@gmu.edu

Library of Congress Control Number: 2005929872

CR Subject Classification (1998): E.3, D.4.6, C.2, F.2.1, J.1, K.6.5

ISSN 0302-9743
ISBN-10 3-540-28138-X Springer Berlin Heidelberg New York
ISBN-13 978-3-540-28138-2 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11535706 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3654

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

Preface

The 19th Annual IFIP Working Group 11.3 Working Conference on Data and Applications Security was held August 7–10, 2005 at the University of Connecticut in Storrs, Connecticut. The objectives of the working conference were to discuss in depth the current state of the research and practice in data and application security, enable participants to benefit from personal contact with other researchers and expand their knowledge, support the activities of the Working Group, and disseminate the research results.

This volume contains the 24 papers that were presented at the working conference. These papers, which had been selected from 54 submissions, were rigorously reviewed by the Working Group members. The volume is offered both to document progress and to provide researchers with a broad perspective of recent developments in data and application security.

A special note of thanks goes to the many volunteers whose efforts made the working conference a success. We wish to thank Divesh Srivastava for agreeing to deliver the invited talk, Carl Landwehr and David Spooner for organizing the panel, the authors for their worthy contributions, and the referees for their time and effort in reviewing the papers. We are grateful to T. C. Ting for serving as the General Chair, Steven Demurjian and Charles E. Phillips, Jr. for their hard work as Local Arrangements Chairs, and Pierangela Samarati, Working Group Chair, for managing the IFIP approval process. We would also like to acknowledge Sabrina De Capitani di Vimercati for managing the conference's Web site.

Last but certainly not least, our thanks go to Alfred Hofmann, Executive Editor of Springer, for agreeing to include these proceedings in the Lecture Notes in Computer Science series. This is an exciting development since, in parallel to the printed copy, each volume in this series is simultaneously published in the LNCS digital library (www.springerlink.com). As a result, the papers presented at the Working Conference will be available to many more researchers and may serve as sources of inspiration for their research. The expanded availability of these papers should ensure a bright future for our discipline and the working conference.

August 2005 Sushil Jajodia and Duminda Wijesekera

Organization

General Chair	T. C. Ting (University of Connecticut, USA)
Program Chairs	Sushil Jajodia and Duminda Wijesekera (George Mason University, USA)
Organizing Chairs	Steven Demurjian and Charles E. Phillips, Jr. (University of Connecticut, USA)
IFIP WG11.3 Chair	Pierangela Samarati (Università degli Studi di Milano, Italy)

Program Committee

Gail-Joon Ahn	University of North Carolina at Charlotte, USA
Vijay Atluri	Rutgers University, USA
Sabrina De Capitani di Vimercati	Università degli Studi di Milano, Italy
Steve Demurjian	University of Connecticut, USA
Roberto Di Pietro	University of Rome "La Sapienza", Italy
Csilla Farkas	University of South Carolina, USA
Eduardo Fernandez-Medina	Univ. of Castilla-La Mancha, Spain
Simon N. Foley	University College Cork, Ireland
Ehud Gudes	Ben-Gurion University, Israel
Carl Landwehr	National Science Foundation, USA
Tsau Young Lin	San Jose State University, USA
Peng Liu	Pennsylvania State University, USA
Sharad Mehrotra	University of California, Irvine
Ravi Mukkamala	Old Dominion University, USA
Peng Ning	North Carolina State University, USA
Sylvia Osborn	University of Western Ontario, Canada
Brajendra Panda	University of Arkansas, USA
Joon Park	Syracuse University, USA
Charles Phillips	U.S. Military Academy, USA
Indrakshi Ray	Colorado State University, USA
Indrajit Ray	Colorado State University, USA
Pierangela Samarati	University of Milan, USA
Sujeet Shenoi	University of Tulsa, USA
David Spooner	Rennselaer Polytechnic Institute, USA
Bhavani Thuraisingham	University of Texas at Dalla, and The MITRE Corp., USA
T.C. Ting	University of Connecticut, USA
Ting Yu	North Carolina State University, USA

Sponsoring Institutions

Center for Secure Information Systems, George Mason University
Department of Computer Science and Engineering, University of Connecticut
Dipartimento di Tecnologie dell'Informazione, Università degli Studi di Milano

Lecture Notes in Computer Science

For information about Vols. 1–3514

please contact your bookseller or Springer

Vol. 3654: S. Jajodia, D. Wijesekera (Eds.), Data and Applications Security XIX. X, 353 pages. 2005.

Vol. 3633: C.B. Medeiros, M. Egenhofer, E. Bertino (Eds.), Advances in Spatial and Temporal Databases. XIII, 433 pages. 2005.

Vol. 3632: R. Nieuwenhuis (Ed.), Automated Deduction – CADE-20. XIII, 459 pages. 2005. (Subseries LNAI).

Vol. 3626: B. Ganter, G. Stumme, R. Wille (Eds.), Formal Concept Analysis. X, 349 pages. 2005. (Subseries LNAI).

Vol. 3621: V. Shoup (Ed.), Advances in Cryptology – CRYPTO 2005. XI, 568 pages. 2005.

Vol. 3619: X. Lu, W. Zhao (Eds.), Networking and Mobile Computing. XXIV, 1299 pages. 2005.

Vol. 3615: B. Ludäscher, L. Raschid (Eds.), Data Integration in the Life Sciences. XII, 344 pages. 2005. (Subseries LNBI).

Vol. 3608: F. Dehne, A. López-Ortiz, J.-R. Sack (Eds.), Algorithms and Data Structures. XIV, 446 pages. 2005.

Vol. 3607: J.-D. Zucker, L. Saitta (Eds.), Abstraction, Reformulation and Approximation. XII, 376 pages. 2005. (Subseries LNAI).

Vol. 3602: R. Eigenmann, Z. Li, S.P. Midkiff (Eds.), Languages and Compilers for High Performance Computing. IX, 486 pages. 2005.

Vol. 3598: H. Murakami, H. Nakashima, H. Tokuda, M. Yasumura, Ubiquitous Computing Systems. XIII, 275 pages. 2005.

Vol. 3597: S. Shimojo, S. Ichii, T.W. Ling, K.-H. Song (Eds.), Web and Communication Technologies and Internet-Related Social Issues - HSI 2005. XIX, 368 pages. 2005.

Vol. 3596: F. Dau, M.-L. Mugnier, G. Stumme (Eds.), Conceptual Structures: Common Semantics for Sharing Knowledge. XI, 467 pages. 2005. (Subseries LNAI).

Vol. 3595: L. Wang (Ed.), Computing and Combinatorics. XVI, 995 pages. 2005.

Vol. 3594: J.C. Setubal, S. Verjovski-Almeida (Eds.), Advances in Bioinformatics and Computational Biology. XIV, 258 pages. 2005. (Subseries LNBI).

Vol. 3587: P. Perner, A. Imiya (Eds.), Machine Learning and Data Mining in Pattern Recognition. XVII, 695 pages. 2005. (Subseries LNAI).

Vol. 3586: A.P. Black (Ed.), ECOOP 2005 - Object-Oriented Programming. XVII, 631 pages. 2005.

Vol. 3584: X. Li, S. Wang, Z.Y. Dong (Eds.), Advanced Data Mining and Applications. XIX, 835 pages. 2005. (Subseries LNAI).

Vol. 3583: R.W. H. Lau, Q. Li, R. Cheung, W. Liu (Eds.), Advances in Web-Based Learning – ICWL 2005. XIV, 420 pages. 2005.

Vol. 3582: J. Fitzgerald, I.J. Hayes, A. Tarlecki (Eds.), FM 2005: Formal Methods. XIV, 558 pages. 2005.

Vol. 3581: S. Miksch, J. Hunter, E. Keravnou (Eds.), Artificial Intelligence in Medicine. XVII, 547 pages. 2005. (Subseries LNAI).

Vol. 3580: L. Caires, G.F. Italiano, L. Monteiro, C. Palamidessi, M. Yung (Eds.), Automata, Languages and Programming. XXV, 1477 pages. 2005.

Vol. 3579: D. Lowe, M. Gaedke (Eds.), Web Engineering. XXII, 633 pages. 2005.

Vol. 3578: M. Gallagher, J. Hogan, F. Maire (Eds.), Intelligent Data Engineering and Automated Learning - IDEAL 2005. XVI, 599 pages. 2005.

Vol. 3577: R. Falcone, S. Barber, J. Sabater-Mir, M.P. Singh (Eds.), Trusting Agents for Trusting Electronic Societies. VIII, 235 pages. 2005. (Subseries LNAI).

Vol. 3576: K. Etessami, S.K. Rajamani (Eds.), Computer Aided Verification. XV, 564 pages. 2005.

Vol. 3575: S. Wermter, G. Palm, M. Elshaw (Eds.), Biomimetic Neural Learning for Intelligent Robots. IX, 383 pages. 2005. (Subseries LNAI).

Vol. 3574: C. Boyd, J.M. González Nieto (Eds.), Information Security and Privacy. XIII, 586 pages. 2005.

Vol. 3573: S. Etalle (Ed.), Logic Based Program Synthesis and Transformation. VIII, 279 pages. 2005.

Vol. 3572: C. De Felice, A. Restivo (Eds.), Developments in Language Theory. XI, 409 pages. 2005.

Vol. 3571: L. Godo (Ed.), Symbolic and Quantitative Approaches to Reasoning with Uncertainty. XVI, 1028 pages. 2005. (Subseries LNAI).

Vol. 3570: A. S. Patrick, M. Yung (Eds.), Financial Cryptography and Data Security. XII, 376 pages. 2005.

Vol. 3569: F. Bacchus, T. Walsh (Eds.), Theory and Applications of Satisfiability Testing. XII, 492 pages. 2005.

Vol. 3568: W.-K. Leow, M.S. Lew, T.-S. Chua, W.-Y. Ma, L. Chaisorn, E.M. Bakker (Eds.), Image and Video Retrieval. XVII, 672 pages. 2005.

Vol. 3567: M. Jackson, D. Nelson, S. Stirk (Eds.), Database: Enterprise, Skills and Innovation. XII, 185 pages. 2005.

Vol. 3566: J.-P. Banâtre, P. Fradet, J.-L. Giavitto, O. Michel (Eds.), Unconventional Programming Paradigms. XI, 367 pages. 2005.

Vol. 3565: G.E. Christensen, M. Sonka (Eds.), Information Processing in Medical Imaging. XXI, 777 pages. 2005.

Vol. 3564: N. Eisinger, J. Małuszyński (Eds.), Reasoning Web. IX, 319 pages. 2005.

Vol. 3562: J. Mira, J.R. Álvarez (Eds.), Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach, Part II. XXIV, 636 pages. 2005.

Vol. 3561: J. Mira, J.R. Álvarez (Eds.), Mechanisms, Symbols, and Models Underlying Cognition, Part I. XXIV, 532 pages. 2005.

Vol. 3560: V.K. Prasanna, S. Iyengar, P.G. Spirakis, M. Welsh (Eds.), Distributed Computing in Sensor Systems. XV, 423 pages. 2005.

Vol. 3559: P. Auer, R. Meir (Eds.), Learning Theory. XI, 692 pages. 2005. (Subseries LNAI).

Vol. 3558: V. Torra, Y. Narukawa, S. Miyamoto (Eds.), Modeling Decisions for Artificial Intelligence. XII, 470 pages. 2005. (Subseries LNAI).

Vol. 3557: H. Gilbert, H. Handschuh (Eds.), Fast Software Encryption. XI, 443 pages. 2005.

Vol. 3556: H. Baumeister, M. Marchesi, M. Holcombe (Eds.), Extreme Programming and Agile Processes in Software Engineering. XIV, 332 pages. 2005.

Vol. 3555: T. Vardanega, A.J. Wellings (Eds.), Reliable Software Technology – Ada-Europe 2005. XV, 273 pages. 2005.

Vol. 3554: A. Dey, B. Kokinov, D. Leake, R. Turner (Eds.), Modeling and Using Context. XIV, 572 pages. 2005. (Subseries LNAI).

Vol. 3553: T.D. Hämäläinen, A.D. Pimentel, J. Takala, S. Vassiliadis (Eds.), Embedded Computer Systems: Architectures, Modeling, and Simulation. XV, 476 pages. 2005.

Vol. 3552: H. de Meer, N. Bhatti (Eds.), Quality of Service – IWQoS 2005. XVIII, 400 pages. 2005.

Vol. 3551: T. Härder, W. Lehner (Eds.), Data Management in a Connected World. XIX, 371 pages. 2005.

Vol. 3548: K. Julisch, C. Kruegel (Eds.), Intrusion and Malware Detection and Vulnerability Assessment. X, 241 pages. 2005.

Vol. 3547: F. Bomarius, S. Komi-Sirviö (Eds.), Product Focused Software Process Improvement. XIII, 588 pages. 2005.

Vol. 3546: T. Kanade, A. Jain, N.K. Ratha (Eds.), Audio- and Video-Based Biometric Person Authentication. XX, 1134 pages. 2005.

Vol. 3544: T. Higashino (Ed.), Principles of Distributed Systems. XII, 460 pages. 2005.

Vol. 3543: L. Kutvonen, N. Alonistioti (Eds.), Distributed Applications and Interoperable Systems. XI, 235 pages. 2005.

Vol. 3542: H.H. Hoos, D.G. Mitchell (Eds.), Theory and Applications of Satisfiability Testing. XIII, 393 pages. 2005.

Vol. 3541: N.C. Oza, R. Polikar, J. Kittler, F. Roli (Eds.), Multiple Classifier Systems. XII, 430 pages. 2005.

Vol. 3540: H. Kalviainen, J. Parkkinen, A. Kaarna (Eds.), Image Analysis. XXII, 1270 pages. 2005.

Vol. 3539: K. Morik, J.-F. Boulicaut, A. Siebes (Eds.), Local Pattern Detection. XI, 233 pages. 2005. (Subseries LNAI).

Vol. 3538: L. Ardissono, P. Brna, A. Mitrovic (Eds.), User Modeling 2005. XVI, 533 pages. 2005. (Subseries LNAI).

Vol. 3537: A. Apostolico, M. Crochemore, K. Park (Eds.), Combinatorial Pattern Matching. XI, 444 pages. 2005.

Vol. 3536: G. Ciardo, P. Darondeau (Eds.), Applications and Theory of Petri Nets 2005. XI, 470 pages. 2005.

Vol. 3535: M. Steffen, G. Zavattaro (Eds.), Formal Methods for Open Object-Based Distributed Systems. X, 323 pages. 2005.

Vol. 3534: S. Spaccapietra, E. Zimányi (Eds.), Journal on Data Semantics III. XI, 213 pages. 2005.

Vol. 3533: M. Ali, F. Esposito (Eds.), Innovations in Applied Artificial Intelligence. XX, 858 pages. 2005. (Subseries LNAI).

Vol. 3532: A. Gómez-Pérez, J. Euzenat (Eds.), The Semantic Web: Research and Applications. XV, 728 pages. 2005.

Vol. 3531: J. Ioannidis, A. Keromytis, M. Yung (Eds.), Applied Cryptography and Network Security. XI, 530 pages. 2005.

Vol. 3530: A. Prinz, R. Reed, J. Reed (Eds.), SDL 2005: Model Driven. XI, 361 pages. 2005.

Vol. 3528: P.S. Szczepaniak, J. Kacprzyk, A. Niewiadomski (Eds.), Advances in Web Intelligence. XVII, 513 pages. 2005. (Subseries LNAI).

Vol. 3527: R. Morrison, F. Oquendo (Eds.), Software Architecture. XII, 263 pages. 2005.

Vol. 3526: S. B. Cooper, B. Löwe, L. Torenvliet (Eds.), New Computational Paradigms. XVII, 574 pages. 2005.

Vol. 3525: A.E. Abdallah, C.B. Jones, J.W. Sanders (Eds.), Communicating Sequential Processes. XIV, 321 pages. 2005.

Vol. 3524: R. Barták, M. Milano (Eds.), Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. XI, 320 pages. 2005.

Vol. 3523: J.S. Marques, N. Pérez de la Blanca, P. Pina (Eds.), Pattern Recognition and Image Analysis, Part II. XXVI, 733 pages. 2005.

Vol. 3522: J.S. Marques, N. Pérez de la Blanca, P. Pina (Eds.), Pattern Recognition and Image Analysis, Part I. XXVI, 703 pages. 2005.

Vol. 3521: N. Megiddo, Y. Xu, B. Zhu (Eds.), Algorithmic Applications in Management. XIII, 484 pages. 2005.

Vol. 3520: O. Pastor, J. Falcão e Cunha (Eds.), Advanced Information Systems Engineering. XVI, 584 pages. 2005.

Vol. 3519: H. Li, P. J. Olver, G. Sommer (Eds.), Computer Algebra and Geometric Algebra with Applications. IX, 449 pages. 2005.

Vol. 3518: T.B. Ho, D. Cheung, H. Liu (Eds.), Advances in Knowledge Discovery and Data Mining. XXI, 864 pages. 2005. (Subseries LNAI).

Vol. 3517: H.S. Baird, D.P. Lopresti (Eds.), Human Interactive Proofs. IX, 143 pages. 2005.

Vol. 3516: V.S. Sunderam, G.D.v. Albada, P.M.A. Sloot, J.J. Dongarra (Eds.), Computational Science – ICCS 2005, Part III. LXIII, 1143 pages. 2005.

Vol. 3515: V.S. Sunderam, G.D.v. Albada, P.M.A. Sloot, J.J. Dongarra (Eds.), Computational Science – ICCS 2005, Part II. LXIII, 1101 pages. 2005.

Table of Contents

Streams, Security and Scalability

Theodore Johnson[1], S. Muthukrishnan[2], Oliver Spatscheck[1], and Divesh Srivastava[1]

[1] AT&T Labs–Research
{johnsont, spatsch, divesh}@research.att.com
[2] Rutgers University
muthu@cs.rutgers.edu

Abstract. Network-based attacks, such as DDoS attacks and worms, are threatening the continued utility of the Internet. As the variety and the sophistication of attacks grow, early detection of potential attacks will become crucial in mitigating their impact. We argue that the Gigascope data stream management system has both the functionality and the performance to serve as the foundation for the next generation of network intrusion detection systems.

1 Introduction

The phenomenal success of the Internet has revolutionalized our society, providing us, e.g., the ability to communicate easily with people around the world, and to access and provide a large variety of information-based services. But this success has also enabled hostile agents to use the Internet in many malicious ways (see, e.g., [10,9,36]), and terms like spam, phishing, viruses, worms, DDoS attacks, etc., are now part of the popular lexicon. As network-based attacks increase, the continued utility of the Internet, and of our information infrastructure, critically depends on our ability to rapidly identify these attacks and mitigate their adverse impact.

A variety of tools are now available to help us identify and thwart these attacks, including anti-virus software, firewalls, and network intrusion detection systems (NIDS). Given the difficulty in ensuring that all hosts run the latest version of software, and the limitations of firewalls (e.g., worms have been known to tunnel through firewalls), NIDS are becoming increasingly popular among large enterprises and ISPs. Network intrusion detection systems essentially monitor the traffic entering and/or leaving a protected network, and look for signatures of known types of attacks. In practice, different NIDS use different mechanisms for the flexible specification of attack signatures. Snort [34], e.g., uses open source rules to help detect various attacks (such as port scans) and alert users. Bro [32], e.g., permits a site's security policy to be specified in a high-level language, which is then interpreted by a policy script interpreter.

As the variety and the sophistication of attacks grow, early detection of potential attacks will become crucial in mitigating the subsequent impact of these attacks (see, e.g., [16,23,25,26,29,24,33,38]). Thus, intrusion detection systems would need to become even more sophisticated, in particular for traffic monitored at high speed (Gbit/sec) links, and it becomes imperative for the next generation of NIDS to:

- provide general analysis over headers and contents of elements in network data streams (e.g., IP traffic, BGP update messages) to detect potential attack signatures.

S. Jajodia and D. Wijesekera (Eds.): Data and Applications Security 2005, LNCS 3654, pp. 1–15, 2005.

- provide highly flexible mechanisms for specifying known attack signatures over these network data streams.
- provide efficient (wire-speed) mechanisms for checking these signatures, to identify and mitigate high speed attacks.

In this paper, we explore the utility of a general-purpose data stream management system (see, e.g., [2,1,4,11]), in particular, Gigascope [13,14,15,12,20], for this purpose We argue that Gigascope has both the *functionality* and the *performance* to serve as the foundation for the next generation of network intrusion detection systems.

The rest of this paper is structured as follows. Section 2 presents the main features of Gigascope's query language in an example driven fashion. Section 3 describes a few representative network-based attacks, and illustrates how Gigascope can be used to aid in the detection of these attacks. Finally, Section 4 describes aspects of Gigascope's run-time architecture that enables high performance attack detection.

2 Gigascope

Gigascope is a high-performance data stream management system (DSMS) designed for monitoring of networks with high-speed data streams, which is operationally used within AT&T's IP backbone [13,14,15,12,20]. Gigascope is intended to be adaptable so it can be used as the primary data analysis engine in many settings: traffic analysis, performance monitoring and debugging, protocol analysis and development, router configuration (e.g., BGP monitoring), network attack and intrusion detection, and various ad hoc analyses. In this section, we focus on the query aspects of Gigascope, and defer a discussion of Gigascope's high-performance implementation until Section 4.

Gigascope's query language, GSQL, is a pure stream query language with an SQL-like syntax, i.e., all inputs to a GSQL query are data streams, and the output is a data stream [20,27]. This choice enables the composition of GSQL queries for complex query processing, and simplifies the implementation. Here, we present the main features of GSQL in an example driven fashion. Later, in Section 3, we show how GSQL can be used to detect various network attacks.

2.1 Data Model

Data from an external source arrives in the form of a sequence of data packets at one or more interfaces that Gigascope monitors. These data packets can be IP packets, Netflow packets, BGP updates, etc., and are interpreted by a protocol. The Gigascope run-time system interprets the data packets as a collection of fields using a library of interpretation functions. The schema of a *protocol stream* maps field names to the interpretation functions to invoke [20].

```
PROTOCOL packet {
    uint time get_time (required, increasing);
    ullong timestamp get_timestamp (required, increasing);
    uint caplen get_caplen;
    unit len get_len;
}
```

```
PROTOCOL Ethernet (packet) {
   ullong Eth_src_addr get_eth_src_addr (required);
   ullong Eth_dst_addr get_eth_dst_addr (required);
   ...
}

PROTOCOL IP (Ethernet) {
   uint ipversion get_ip_version;
}

PROTOCOL IPV4 (IP) {
   uint protocol get_ipv4_protocol;
   IP sourceIP get_ipv4_source_ip;
   IP destIP get_ipv4_dest_ip;
   ...
}
```

Network protocols tend to be layered, e.g., an IPV4 packet is delivered via an Ethernet link. As a convenience, the protocol schemas have a mechanism for field inheritance (specified in parentheses). For example, the `Ethernet` protocol contains all the fields of the `packet` protocol, as well as a few others.

2.2 Filters

A *filter* query selects a subset of tuples of its input stream, extracts a set of fields (possibly transforming them), then outputs the transformed tuples in its output stream. The following query extracts a set of fields for detailed analysis from all `TCP` (`protocol` = 6) packets.

```
Q_1^s:  SELECT   time, timestamp, sourceIP, destIP,
                 source_port, dest_port, len
        FROM     TCP
        WHERE    protocol = 6
```

Gigascope supports multiple data types (include IP), and multiple operations on these data types. The following query extracts a few fields from the IPV4 tuples whose `sourceIP` matches `128.209.0.0/24`, and names the resulting data stream as `fq` (this can then be referenced in subsequent GSQL queries).

```
Q_2^s:  DEFINE   { query_name fq; }
        SELECT   time, sourceIP, destIP
        FROM     IPV4
        WHERE    sourceIP & IP_VAL'255.255.255.0' = IP_VAL'128.209.0.0'
```

2.3 User-Defined Functions

While GSQL has a wide variety of built-in operators, there are situations where a user-defined function would be more appropriate. Gigascope permits users to define functions, and reference them in GSQL queries. The following query, for example, uses longest prefix matching on the `sourceIP` address against the local prefix table to extract data about IPV4 packets from local hosts.

```
Q_1^f: SELECT   time/60, sourceIP
       FROM     IPV4
       WHERE    getlpmid(sourceIP, 'localprefix.tbl') > 0
```

2.4 Aggregation

The following *aggregation* query counts the number of `IPV4` packets and the sum of their lengths from each source IP address during 60 second epochs.

```
Q_1^a: SELECT   tb, sourceIP, count(*), sum(len)
       FROM     IPV4
       GROUP BY time/60 as tb, sourceIP
```

Aggregation can be combined with user-defined functions to create sophisticated analyses. The following aggregation query uses a group variable computed using a user-defined function, to count the number of `IPV4` packets and the sum of their lengths from each local host during 60 second epochs.

```
Q_2^a: SELECT   tb, localHost, count(*), sum(len)
       FROM     IPV4
       WHERE    getlpmid(sourceIP, 'localprefix.tbl') > 0
       GROUP BY time/60 as tb,
                getlpmid(sourceIP, 'localprefix.tbl') as localHost
```

2.5 Merges and Joins

A GSQL *merge* query permits the union of streams from multiple sources into a single stream, while preserving the temporal (ordering) properties of one of the (specified) attributes. The input streams must have the same number and types of fields, and the merge fields must be temporal and similarly monotonic (both increasing or both decreasing). For example, the following query can be used to merge data packets from two simplex *physical* (optical) links to obtain a full view of the traffic on a *logical* link. Such merge queries have proven very useful in Gigascope for network data analysis.

```
Q_1^m: DEFINE   { query_name logicalPktsLink; }
       MERGE    O1.timestamp : O2.timestamp
       FROM     opticalPktsLink1 O1, opticalPktsLink2 O2
```

A GSQL *join* query supports the join of two data streams, with a temporal join predicate (possibly along with other predicates), and will emit a tuple for every pair of tuples from its sources that satisfy the predicate in the GSQL `WHERE` clause. The following query, for example, computes the delay between a `tcp_syn` and a `tcp_ack`.

```
Q_1^j: SELECT   S.tb, S.sourceIP, S.destIP, S.source_port,
               S.dest_port, (A.timestamp − S.timestamp)
       FROM     tcp_syn S, tcp_ack A
       WHERE    S.sourceIP = A.destIP and S.destIP = A.sourceIP and
               S.source_port = A.dest_port and S.dest_port = A.source_port
               S.tb = A.tb and S.timestamp <= A.timestamp and
               (S.sequence_number + 1) = A.ack_number
```

Joins can be combined with aggregates for complex GSQL queries.

2.6 User-Defined Aggregation and Sampling

GSQL permits users to define aggregate functions (UDAFs), and reference them in queries, just like regular aggregates [12]. The specification of the UDAF consists of multiple functions: INITIALIZE (which initializes the state of a scratchpad space), ITERATE (which inserts a value to the state of the UDAF), OUTPUT (to support multiple return values from the same UDAF computation), and DESTROY (which releases UDAF resources).[1]

For example, using GSQL's UDAF mechanism, approximate quantile streaming algorithms can be coded, and accessed like in the following query, to compute the median value of `len` for each source IP address and 60 second epoch:

```
Q_1^u: SELECT   tb, sourceIP, count(*), percentile(len,50)
       FROM     IPV4
       GROUP BY time/60 as tb, sourceIP
```

The UDAF mechanism is useful to obtain point values (e.g., median packet length), but it is cumbersome for obtaining set values, such as in returning a sample of the data stream (e.g., a subset-sums or a reservoir sample). Given the utility of sampling to analyze high-speed streams, GSQL supports a sampling operator that can be specialized by users to implement a wide variety of stream sampling algorithms [21]. The key observation employed is that even though there are many differences between various stream sampling algorithms, they follow a common pattern. First, a number of items are collected from the original data stream according to a certain criterion (possibly with aggregation in the case of duplicates); this is the *insert* phase. Then, if a condition on the sample is triggered (e.g., the sample is too large), the size of the sample is reduced according to another criterion; this is the *compress* phase. This alternation of insert and compress phases can be repeated several times in each epoch. At the end of the epoch, the sample is output; this is the *output* phase. For example, the following query will report the 100 most common source IP addresses within a 60 second epoch.

[1] Additional functions are needed to deal with Gigascope's two-level architecture, which we do not discuss further.

Q_2^u: SELECT tb, sourceIP
FROM IPV4
GROUP BY time/60 as tb, sourceIP
CLEANING WHEN local_count(100) = TRUE
CLEANING BY count(*) < current_bucket() − first(current_bucket())

2.7 Query Set

Complex analyses are best expressed as combinations of simpler pieces. By permitting GSQL queries to be named, and re-used in the `FROM` clause of other GSQL queries, a set of inter-related queries, forming a query DAG, can be defined.

3 Attacks

A large variety of network-based attacks have been discussed in the literature, including viruses, worms, DDoS attacks, etc. (see, e.g., [10,9,16,23,25,26,29,24,33,36,38]). Here, we discuss a few representative attacks, and illustrate how Gigascope can be used to aid in the detection of these attacks.

3.1 Denial of Service

A *denial of service* (DoS) attack is characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service [7]. DoS attacks have been among the most common form of Internet attacks. The basic form of a DoS attack is to consume scarce computer and network resources, such as kernel data structures, CPU time, memory and disk space, and network bandwidth.

Email Bombing: An example DoS attack that attempts to consume system and network resources is `Email Bombing`, where attackers send excessively many and large e-mail messages to one or more accounts at a specific victim site [8]. When the attacker makes use of a dispersed set of sources to coordinate such an attack, it is referred to as a distributed DoS (DDoS) attack.

`Email Bombing` can be detected at the victim site if email is sluggish, possibly because the mailer is trying to process too many messages. An alternative way of checking for this possibility is to monitor the `SMTP` traffic entering a protected network using Gigascope, and check for hosts that show significant deviations in expected traffic at port 25/`SMTP`. The following simple GSQL query can track the total `SMTP` traffic for individual destination IP addresses. Deviations can be monitored by comparing recent behavior with more historical trends.

Q_1^{dos}:DEFINE { query_name smtp_perhost; }
SELECT tb, destIP, count(*), sum(len)
FROM TCP
WHERE protocol = 6 and dest_port = 25
GROUP BY time/60 as tb, destIP