Securing Ultra-High-Bandwidth Science DMZ Networks with Coordinated Situational Awareness

Vasudevan Nagendra
Stony Brook University
vnagendra@cs.stonybrook.edu

Vinod Yegneswaran
SRI International
vinod@csl.sri.com

Phillip Porras
SRI International
porras@csl.sri.com

ABSTRACT

The Science DMZ (SDMZ) is a special purpose network infrastructure that is engineered to cater to the ultra-high bandwidth needs of the scientific and high performance computing (HPC) communities. These networks are isolated from stateful security devices such as firewalls and deep packet inspection (DPI) engines to allow HPC data transfer nodes (DTNs) to efficiently transfer petabytes of data without associated bandwidth and performance bottlenecks. This paper presents our ongoing effort toward the development of more fine-grained data flow access control policies to manage SDMZ networks that service large-scale experiments with varying data sensitivity levels and privacy constraints.

We present a novel system, called `CoordiNetZ` (`CNZ`), that provides coordinated security monitoring and policy enforcement for sites participating in SDMZ projects by using an *intent-based policy framework* for effectively capturing the high-level policy intents of non-admin SDMZ project users (e.g., scientists, researchers, students). Central to our solution is the notion of *coordinated situational awareness* that is extracted from the synthesis of context derived from SDMZ host DTN applications and the network substrate. To realize this vision, we present a specialized process-monitoring system and flow-monitoring tool that facilitate context-aware data-flow intervention and policy enforcement in ultra-high-speed data transfer environments. We evaluate our prototype implementation using case studies that highlight the utility of our framework and demonstrate how security policy could be effectively specified and implemented within and across SDMZ networks.

1 INTRODUCTION

Today researchers struggle to exchange humongous volumes of science data, on the order of petabytes per month, across widely dispersed research institutes that span multiple countries. Some of the noteworthy challenges include: (*i*) DTNs that cannot sustain or scale to line-rate data transfers, (*ii*) stateful firewalls and DPI devices that drastically degrade the performance of the science data transfers [22] and (*iii*) lossy wide area network (WAN) channels having unpredictable transport performance impacting overall data transfer rates [25, 26]. These challenges are effectively addressed by the U.S. Department of Energy (DoE) through the SDMZ network. SDMZs are deployed at more than 40 core research sites, which conduct high-speed transfers (i.e., petabytes of data per month) using the following elements:

1. Customized host DTNs [1, 20, 27] that transfer data at 10 to 100 Gbps,
2. SDMZ network perimeter architecture that bypasses stateful DPI devices for their science data transfers (as shown in Figure 1) and
3. Dedicated SDMZ core network with capacity to carry more than 100 Gbps of science data-flow rates without loss[1].

Currently, more than 140 campus networks are peered to this SDMZ Core with direct 10/40/100 Gbps uplinks, collectively exchanging more than 50 petabytes of data each month [10]. Implementing security policy for effectively managing such ultra-high-volume data transfers without sacrificing underlying transport performance and throughput requirements is a formidable challenge. The SDMZ network primarily relies on coarse-grained traffic management mechanisms using router-based access control lists (ACLs) and aggressive filtering, to secure its infrastructure [6]. To address the latency, packet loss, and bandwidth constraints each SDMZ installation and its DTNs are isolated from enterprise local area networks (LAN) and configured to bypass stateful deep-packet-inspecting middleboxes (e.g., firewalls, intrusion prevention systems (IPSs)) [6, 22, 23].

This architectural decision leaves these networks open to adversarial threats both from insiders and external networks [9, 22]. The SDMZ network administrators, like large-scale Internet service providers (ISPs), need to rely on offline detection mechanisms and switch/router ACLs to block malicious traffic flows at such ultra-high line rates [4], as other approaches such as traffic shunting and scrubbing require significant human resources.

HotNets-XVI, November 30–December 1, 2017, Palo Alto, CA, USA

ACM ISBN 978-1-4503-5569-8/17/11…$15.00
https://doi.org/10.1145/3152434.3152460

[1]Considering the growing bandwidth requirements of SDMZ applications this core network is soon expected to get upgraded to 400 Gbps [21]

We make the case that security mechanisms currently implemented in SDMZ networks fall short along multiple dimensions. First, security mechanisms implemented in contemporary SDMZ networks are too coarse-grained (IP, port-level ACLs) for managing high-performance science applications that exchange potentially very sensitive, proprietary, or personal-private information across interconnected multi-institutional networks [6, 7, 9]. Second, the lack of application awareness, DPI capabilities and contextual information leaves wide gaps in the SDMZ security architecture, limiting effective security analysis of network packets and flows [22]. Notably, SDMZ networks exchange large volumes of opaque traffic (i.e., traffic that is either encrypted or compressed). This prevents the network-monitoring plane (e.g., NIDS) from making dynamic and fine-grained filtering decisions for security policy enforcement based on operational context information (i.e., *who*, *what*, *where*, *when* and *how* the resource is being accessed or requested).

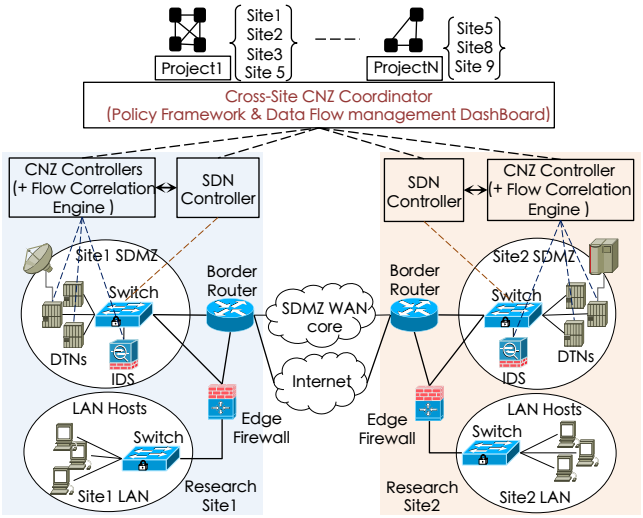


Figure 1: Proposed SDMZ Security Architecture

We introduce a new system, called `CoordiNetZ`, (shown in Figure 1) which enables an SDMZ project user to express fine-grained data-flow, user, and application policies. It works by providing application-level context to network nodes that enforce security, enabling them to filter traffic or route it for DPI. The necessary contextual information, built at the host DTNs, is shared across the network-control and monitoring plane for fine-grained policy enforcement. `CoordiNetZ` effectively integrates an intent-based policy management framework with context-based monitoring and enforcement mechanisms. This enables SDMZ project users to express the expected experiment interactions such that we can arbitrate conflicts in how applications and network data flows interact with respect to project- and site-specific policies.

The SDMZ network infrastructure is primarily used by non-admin users (such as professors, scientists, researchers and students) for collaborating on science projects across multiple sites in accordance with their project and experimental policies. Considering the non-admin science user's role in an SDMZ project's policy specification, its policy framework must be simple, intuitive and provide the ability to directly capture the user's intent by decoupling *what* policies are to be enforced from *how* to enforce them. Hence, `CoordiNetZ` incorporates an intent-based policy framework (as illustrated in Section 2.2) that abstracts the intricate low-level details of the network infrastructure.

We present a prototype implementation that demonstrates how our `CNZ Controller` and `CNZ Coordinator` interact with a new host monitoring system called `SciMon` and flow summarization tool called `SciFlow`. `SciMon` communicates the SDMZ network's host application context to the network monitoring plane and effectively allows it to enforce security policy by dynamically blocking or steering traffic to appropriate IDS instances using SDN techniques. Finally, we present three operational use-cases that highlight the utility of the system.

2 MOTIVATION

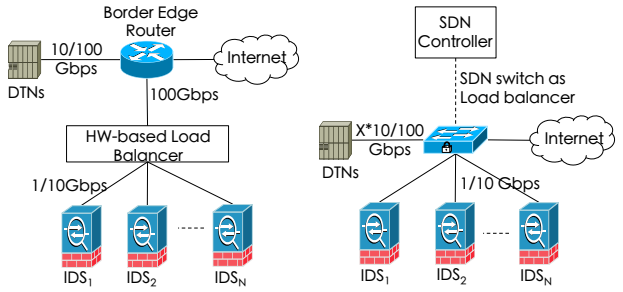
The SDMZ network infrastructure is emerging as a vital platform for storing and transporting petabytes of scientific data across research testbeds and data repositories in the US and Europe. The SDMZ network infrastructure differs from enterprise LAN networks in the following respects: (*i*) individual SDMZ applications typically transfer terabytes of data per-month over elephant flows, (*ii*) unlike traditional desktops and servers, host DTNs are customized to handle high performance TCP with a limited set of "trusted" data transfer applications [20] and (*iii*) security and protection for science data flows is provided using simple router-based ACLs that isolate stateful inspection devices along the data path [22].

2.1 Data Flow Tracking

SDMZ sites participate in project experiments that exchange data with varying sensitivity levels (and privacy constraints). However, the cross-site and multi-tenant nature of experiments introduce potential vulnerabilities, such as leaks, resource misuse, or integrity threats that may arise from malicious (or accidental) accesses occurring during an SDMZ experiment. Hence, it is important to expose greater visibility into how the data is accessed at each of the sites, how it is being transformed and to where it is being transported. Understanding and following data flow and transformation across the project resources is vital to ensuring data security.

2.2 Unified Policy Framework

Currently, security policy requirements for a specific project are exchanged offline (i.e., via documents or email) between multiple site admins participating in that project. These policy rules are therefore manually inserted (and often statically configured) into routers and monitoring devices. Existing SDN based policy frameworks (such as PGA [19], Kinetic [13],



(a) Berkeley DoE National Laboratory IDS Model [3]. **(b) Indiana Global Network Operation IDS Model [23].**

Figure 2: Current Clustered / Parallel IDS Deployments

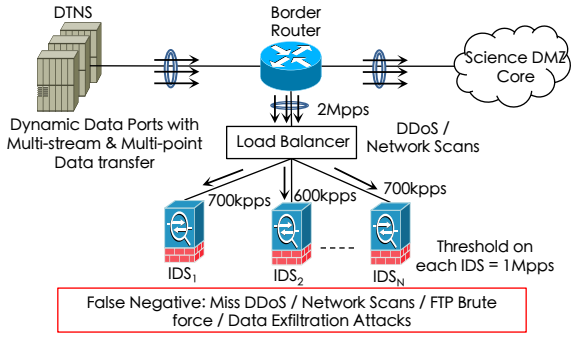


Figure 3: Lack of Context: Missed Network-Level Attacks

Pyretic [15]) lack mechanisms to handle the dynamic data security policy requirements that arise in SDMZ deployments, as we describe below.

To address the limitations of existing policy specification and deployment mechanisms, we make the case for a new intent-based policy specification language with efficient composition techniques to automatically arbitrate conflicts among the policies specified by different projects and users. This would enable the development of a unified policy framework that could effectively reconcile policy intents across multiple sites owned by different administrative authorities. The framework should address the following considerations:

Policy 1. *Data Flow Policies.* Policies that limit access and transfer of sensitive data.

`Example`: Sensitive data derived from project P1 experiment2 in site S1 must only be shared among nodes running P1 experiments. If projects P1 and P2 are co-resident in an SDMZ application node, P2 users or applications may not exfiltrate P1 data to other nodes.

`Example`: Application binaries that are not white-listed are not allowed to access sensitive files or send packet of size greater than X bytes using protocols such as DNS and NTP to specific geo-locations.

Policy 2. *Temporal and spatial policies.* Policies that limit access based on time, network address space or geography.

`Examples`: (1) Sensitive science data produced by project P1 is not allowed to be accessed or transmitted after 6 PM and before 9 AM (e.g., in the absence of administrators, to prevent malicious data access). (2) Export-controlled scientific data derived from project P2 is not to be transmitted to IP addresses that are geo-located within ITAR restricted countries.

Policy 3. *Network security policies.* Policies that deal with dynamic security state of the network.

`Example`: Notify admin and quarantine hosts to prevent any sensitive data transfers outside DTN if there is evidence of a successful brute-force attack.

2.3 Contextual Awareness

The performance requirements of ultra-high-bandwidth networks like the SDMZs preclude the use of in-line stateful security devices (e.g., firewalls, IPS, application-layer proxies) for securing data flows. As even offline security solutions (such as Bro IDS) cannot effectively handle the data flow rates (of 10-100 Gbps), clustered detection techniques, shown in Figure 2, have been proposed with SDN-based dynamic traffic steering for efficient DPI. However, these techniques fall short in effectively handling elephant flows that are common in SDMZ experiments. Consider for example the case of SDMZ fast data-transfer applications (e.g., GridFTP, bbftp, bbcp, globus [11]) which are basically opaque multi-point and multi-stream applications, where a single data flow can be transferred in parallel as multiple data streams on to multiple data nodes.

Consolidating or correlating the distributed parallel TCP streams is difficult as the TCP port numbers used in the data transfer are dynamically negotiated as part of the GridFTP secure control messages (using the FTP `PORT` command). As shown in Figure 3, various attacks such as application-layer DDoS and brute-force attacks could go undetected with a clustered IDS solution. We posit that the inability of the traditional or SDN-based security solutions to detect these attacks can be effectively addressed with associated context from the SDMZ applications.

3 SYSTEM DESIGN

We present the design of `CoordiNetZ` SDMZ security architecture that is a tangible step toward addressing the limitations discussed above. An overview of the `CoordiNetZ` system architecture is shown in Figure 1 with site specific details illustrated in Figure 4. Following set of capabilities are provided by `CoordiNetZ` to enhance the security of SDMZ networks: (*i*) coordinated situational awareness, (*ii*) centralized data flow tracking and (*iii*) intent-based policy framework.

The prototype components of DTN, `CNZ Controller` and `CNZ Coordinator` were built using Java and Python and tested on an Ubuntu 16.04.2 LTS system. As a proof of concept, we have developed `CoordiNetZ` with the following functional capabilities. The components of `CoordiNetZ` include: (a) `host DTN`: a host customized for high data transfer rates (b) `SciMon`: a host-based (**Sci**ence DMZ **Mon**itor) (c) `SciFlow` (**Sci**ence

Flow): a context-aware flow summarization tool (d) `CNZ Controller`: mediator between `SciMon` and `CNZ Coordinator` and (e) `CNZ Coordinator` (Cross-Site Data Flow & Policy Management framework).

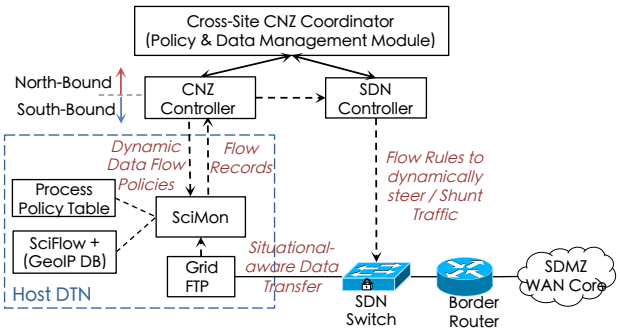


Figure 4: `CoordiNetZ` - Site-Level System Architecture

3.1 Data Transfer Node (DTN)

The DTN host is customized with the ability to provide the context necessary for the system to effectively secure the SDMZ infrastructure. The main focus of the host customization is to provide all the necessary information pertaining to a file I/O and its associated network operation to the `CNZ Controller`, which could further be shared with other network monitoring nodes (such as BroIDS) and SDN controllers. In certain cases, applications may have to be instrumented (e.g., GridFTP [24] in our case) on the host DTN to expose the application-specific information that is exchanged via secured control messaging. In other cases, such context may be simply derived from the application logs.

3.2 SciMon

The `SciMon` module performs the following specific tasks: (*i*) build contextual information at process level required for effectively filtering the flows inside the network, (*ii*) collect the data tracking details required for building the data-flow graphs at `CNZ Coordinator` and (*iii*) provide infrastructure support for enforcing host and process-specific data policies on to the DTN host. The `SciMon` module continuously monitors the host for process instantiations and associated file system access and network IO events to build host-specific data flow contextual information. The network flow related contextual information is gained with the `SciFlow` module (discussed in Section 3.3).

User: <USER Name>
Application Binary: <Application Binary Name>
Process: <Process name or PID>
Time: <Temporal details>
From location: <city/latitude/longitude/Country of Origin>
Action: <Block Operation / Notify Admin>
Network Source: <Black Listed Countries / IPs / domain-names>
Network destination: <Black Listed Countries / IPs / domain-names>

Figure 5: Sample Process Flow Table Entry

[SciFlow]: srcIP, srcPort, dstIP, dstPort, start, end, duration, protocol, state, srcZeropaks, srcDatapaks, srcAvgpak, srcBytecnt, srcPakcnt, dstZeropaks, dstDatapaks, vlan, dstAvgpak, dstBytecnt, dstPakcnt, updateTime, updateSrcBytecnt, pdateSrcPakcnt, srcPrefix, dstPrefix, updateDstBytecnt, updateDstPakcnt, icmpPakcnt, srcDomain, dstDomain, srcCountry, srcCity, dstCountry, dstCity, srcLatitude, userID, srcLongitude, dstLatitude, dstLongitude, IPScore

#[SciMon]: username, hostname, processID, appname, execpath, execArguments, execCredential, openFileList, integrity, pProcessID, pAppname, pExecPath, sensorID, sensorVer

Figure 6: DTN Flow Record Field (`Flow Record = Timestamp + SciFlow Record + SciMon Record`).

`SciMon` keeps track of the files accessed with each process instantiation and imposes file access and network data flow restrictions in accordance with the policies configured in the Process Policy Table (shown in Figure 5). The policies specified at the `CNZ Coordinator` module are decomposed into a set of Process Flow Rules to be configured onto each DTN's Process Policy Table as shown in Figure 5. Each policy dictates the access control rules pertaining to the application binary, process, user, and their access restrictions on sensitive science data and the data transport over network.

3.3 SciFlow

`NetFlow`[17] is a popular network flow monitoring tool that generates flow-level traffic summaries (packet counts, byte counts etc.) in the form of NetFlow records. While highly scalable (over 100K flows per second), the limited set of flow attributes provided by the NetFlow limits it from being a comprehensive solution for security monitoring. `SciFlow` seeks to addresses this limitation by introducing additional flow specific security context such as DNS transaction summaries, unfinished SYN handshakes, unsolicited ACKs, ACK timeouts, IP address reputation [2], and geography information (domain, country, city, latitude and longitude)[12].

The `SciFlow` module runs as a daemon to monitor for flows generated from a specific interface inside the host (or can run independently on any network device) and triggers `SciMon` to gather user and process info, file I/O, and application binary information associated with this network flow. The flow records gathered by the `SciMon` and `SciFlow` modules at the host are sent to the `CNZ Controller` for further processing. A snapshot of the flow records generated by the `SciMon` and `SciFlow` modules is shown in Figure 6. The fields that are extracted from the host and network flows are customized per `CNZ Controller`'s policies.

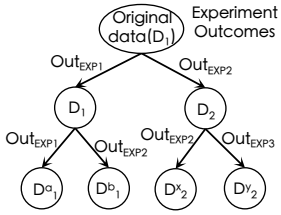
3.4 CNZ Controller

`CNZ Controller` acts as a mediator between the DTN hosts and the `CNZ` coordinator, and provides three key functional capabilities: (*i*) collects host and process-layer contextual information provided by multiple hosts as flow records (shown in Figure 6) and consolidates these attributes into meta-data for forwarding to the `CNZ Coordinator`, to

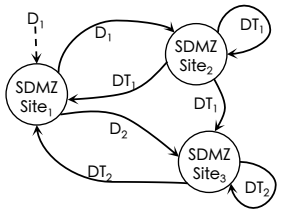
build data-flow graphs, (*ii*) reconciles project-specific policies pushed down from the `CNZ Coordinator` with site-specific policies to generate host-specific rules for policy enforcement and (*iii*) triggers the SDN Controller to insert flow rules for filtering the malicious traffic.

3.5 CNZ Coordinator

The `CNZ Coordinator` acts as a centralized manager for specifying project policies across various SDMZ sites and implements two essential functional components: (*i*) `DashBoard` for data tracking and (*ii*) an intent-based framework for policy specification, composition, and decomposition.



(a) Tree-based representation of data transformation with-in a SDMZ site.



(b) Graph-based representation of cross-site data-flow tracking.

Figure 7: Dynamic Data-Flow Management

Data Flow Tracking DashBoard. As discussed in Section 2, the current SDMZ infrastructure does not provide any capabilities for enforcing cross-site data flow policies. While prior work discussed data flow tracking within a host and across hosts[14, 16, 18, 28], these frameworks are heavyweight and do not address the performance requirements of the SDMZ. Hence, we implement a lightweight forensic tracker and use the `CNZ DashBoard` to support two key data-tracking capabilities: (*i*) ability to capture all readand write operations carried out on the data within a host (shown in Figure 7a) and (*ii*) ability to effectively capture the flow of data across hosts and associated data flow and data transformation restrictions (shown in Figure 7b).

Therefore, we define the following properties and capabilities to effectively track the data across SDMZ sites: (*i*) a unique data identifier across sites within a project and (*ii*) a mechanism to capture the relation between "original" and "transformed" data. The unique data identifier is required to identify the data across multiple sites and capture its transformation in the future. This also allows SDMZ project users to effectively query the data flow and data transformation details. The relation between the original and the transformed data is captured at each of the SDMZ project hosts and shared with the `CNZ Coordinator` for building data flow graphs. Data flow tracking allows the users to monitor for data-flow violations and specify the new policies to restrict violating data flows using a high-level policy specification language discussed below.

SDMZ Security Policy Framework. The `CNZ Coordinator` simplifies cross-site policy specification by providing a platform for SDMZ users to specify

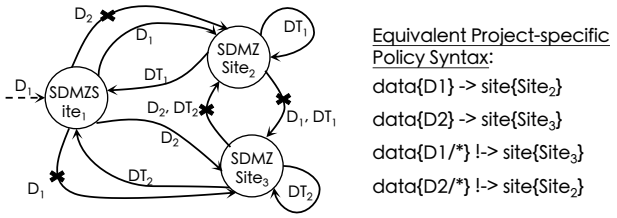


Figure 8: Graph & Syntax-based Policy Specification

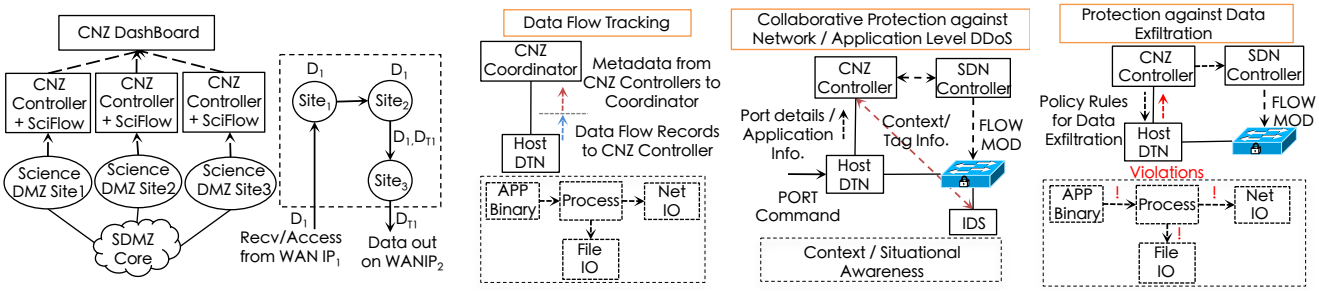
policies through a high-level policy specification language and a graph-based policy specification mechanism (shown in Figure 8). The `CNZ Coordinator` allows policies to be specified by multiple site administrators or users independently. These policies are effectively composed together for resolving conflicts. Subsequently, the conflict-free policies are then decomposed into site-specific policies by `CNZ Coordinator`, which are further translated into device-specific rules by the `CNZ Controller`.

The SDMZ currently supports the following three types of policies using the `CNZ Coordinator` framework: (*i*) data flow policies, (*ii*) host and process-specific ACLs and (*iii*) network security ACLs. For example, consider a typical science data flow using multi-point and multi-streaming opens tens of TCP ports to multiple host DTNs to send the data, which complicates the policy specification mechanism for the administrator. The `CNZ Coordinator` abstracts away the complexities and represents the policies as data flow policies (shown in Figure 8), which will be automatically decomposed and enforced on to SDMZ network as host- and network-specific ACLs. This allows the user to specify the policies either by using host or network ACLs or by using abstracted data-flow graphs. Policies with scope limited to a single site can be specified at the `CNZ Controller` module, while cross-site policies are specified through the `CNZ Coordinator`.

4 `COORDINETZ` SECURITY USE-CASES

As part of science experiments, SDMZ users bring in a wide variety of data with varying sensitivity levels into the SDMZ network infrastructure (e.g., patient's private data, classified government data, proprietary data, public data). Protecting data belonging to different administrative domains, by providing them with necessary security infrastructure for supporting their experimental need is a key requirement. To illustrate the benefits of `CoordiNetZ`, we discuss the following three use-cases: (*i*) providing necessary capability to SDMZ users for effectively tracking their data flow and access patterns across SDMZ sites, (*ii*) protecting sensitive data from malicious exfiltration and (*iii*) improving detection fidelity with context-awareness provided by `CoordiNetZ` .

1) Data Exfiltration. An important security requirement of SDMZ environments is protection of data from unauthorized access and exfiltration to unauthorized locations. To illustrate the capability of `CoordiNetZ` to provide the protection against data exfiltration, we consider the following scenarios.



(a) `CNZ DashBoard` - Data Flow Tracking **(b) Data Flow Tracking** **(c) Collaborative Protection** **(d) Data Exfiltration**

Figure 9: `CoordiNetZ` SDMZ Use-Cases. Context: *who*(user/app./process), *what*(file/network I/O), *how*(remote login), *when*(timestamp), *where*(country, city, IP)

An attacker exploits the GridFTP application's vulnerability (CVE-2012-3292 [5]) or uses a brute-force password cracking scheme to gain SDMZ user privileges on the DTN host. Upon gaining privileges the attacker could readily access the files and exfiltrate the data outside the SDMZ host. But, the underlying host-level protection mechanism (i.e., `SciMon`) monitors the host system based on the data exfiltration policies configured on the host and raises violations that occur during the process, network and file I/O interactions (see Figure 9c) which are forwarded to the `CNZ Controller`. `SciMon` enforces policies restricting access based on the following attributes: (*a*) usernames, (*b*) application binaries, (*c*) ability to access sensitive files, (*d*) ability to send data out of host (protocol level restrictions such as packet size, protocol etc.), (*e*) situational information (such as time, location, geolocation etc.). The `CNZ Controller` could configure network devices with a block rule (e.g., using an OpenFlow FlowMod rule) to thwart the data exfiltration.

2) Data Tracking. To illustrate the benefits of `CNZ DashBoard's` data tracking capability, we generate traffic that traverses sequentially between three hosts, each representing an SDMZ site (Site 1, Site 2 and Site 3), as shown in Figure 9a. The data is allowed to traverse multiple sites and is potentially subject to multiple transformations. `SciMon` which continuously monitors for process-related activities (i.e., process arguments, file I/O, network I/O) captures the data flow activity within and across hosts among different SDMZ sites. As shown in Figure 9b, the captured flow records are pushed onto the `CNZ Controller` for initial processing. These flow records are translated to metadata, which can be easily consumed by `CNZ Coordinator` for building the data-flow graphs (as discussed in Section 3.5). Examples of data-flow tracking graphs captured within a host and across sites is shown in Figures 7a and 7b respectively.

3) Network/Application-Level Attack Detection. Clustered monitoring (see Section 2), prevents the IDS instances from effectively detecting attacks (such as DDoS and reconnaissance scans) using threshold-based filters as shown in Figure 3. The fact that SDMZ data-transfer applications (e.g., GridFTP, ddftp), rely on encryption and parallel data streaming techniques, using dynamically generated ports, further complicates network intrusion detection.

`CoordiNetZ` addresses this problem by providing contextual information from the host DTN to BroIDS, which effectively allows the traffic to be aggregated or categorized for filtering. In Figure 9d, the host DTN node adds flow-based tags[8] to the traffic that need to be collectively inspected by the same IDS instance and also adds necessary rules in the SDN switch to steer the traffic in accordance with flow-based tags to the respective IDS entity.

5 CONCLUSION

The `CoordiNetZ` system addresses the problem of securely managing SDMZ applications and experiment data across a broad range of projects, sites, and user communities. `CoordiNetZ` helps bridge a critical gap between applied-security research and science experiments on real near-production infrastructure at scale, maximizing the benefits of SDN. This is effectively achieved in `CoordiNetZ` by extracting the necessary contextual information from the host systems at the granularity of process-specific details pertaining to its file and network IO and distributing it to the network through SDN and `CNZ Controller` entities. `CoordiNetZ` also provides a platform that allows science users to gain visibility on their data through its data-flow tracking `DashBoard` and data-specific policies. We believe that this framework will spawn collaborative research on new security mechanisms and provide a foundation for studying the cybersecurity challenges in this vital infrastructure.

ACKNOWLEDGMENTS

We greatly appreciate Jennifer Rexford (our shepherd) and the anonymous reviewers for their insightful feedback. This work was supported in part by the National Science Foundation (NSF) award no. 1642150.

REFERENCES

[1] 100G DTN. 2017. https://fasterdata.es.net/science-dmz/DTN/100g-dtn/.
[2] AlienVault: IP reputation DB. 2017. https://reputation.alienvault.com/reputation.data.

[3] Berkeley Lab 100G Intrusion Detection System. 2017. https://goo.gl/xc61Zv.

[4] Best Practices for Science DMZ Security. 2017. https://goo.gl/hvDEr3.

[5] CVS GridFTP Vulnerability for attackers to gain privileges. 2017. http://www.cvedetails.com/cve/CVE-2012-3292/.

[6] E. Dart, L. Rotman, B. Tierney, M. Hester, and J. Zurawski. The Science DMZ: A Network Design Pattern for Data-intensive Science. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '13, pages 85:1–85:10, New York, NY, USA, 2013. ACM.

[7] ESnet's Science DMZ Breaks Down Barriers, Speeds up Science. 2015. https://cs.lbl.gov/news-media/news/2015/esnet-science-dmz/.

[8] S. K. Fayazbakhsh, L. Chiang, V. Sekar, M. Yu, and J. C. Mogul. Enforcing Network-Wide Policies in the Presence of Dynamic Middlebox Actions using FlowTags. In *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, pages 543–546, Seattle, WA, 2014. USENIX Association.

[9] Firewall TCP Performance with Science DMZ. 2017. https://fasterdata.es.net/assets/fasterdata/Firewall-tcptrace.pdf.

[10] How the World's Fastest Science Network Was Built. 2017. https://esnetupdates.wordpress.com/2016/08/12/how-the-worlds-fastest-science-network-was-built/.

[11] How to transfer large amounts of data via network. 2017. http://moo.nac.uci.edu/~hjm/HOWTO_move_data.html.

[12] IP Geo-location DB. 2017. https://dev.maxmind.com/geoip/legacy/geolite/.

[13] H. Kim, J. Reich, A. Gupta, M. Shahbaz, N. Feamster, and R. Clark. Kinetic: Verifiable dynamic network control. In *Proceedings of the 12th USENIX Conference on Networked Systems Design and Implementation*, NSDI'15, pages 59–72, Berkeley, CA, USA, 2015. USENIX Association.

[14] T. Malik, L. Nistor, and A. Gehani. Tracking and sketching distributed data provenance. In *Proceedings of the 2010 IEEE Sixth International Conference on e-Science*, ESCIENCE '10, pages 190–197, Washington, DC, USA, 2010. IEEE Computer Society.

[15] C. Monsanto, J. Reich, N. Foster, J. Rexford, and D. Walker. Composing Software-defined Networks. In *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation*, NSDI'13, pages 1–14, Berkeley, CA, USA, 2013. USENIX Association.

[16] D. Muthukumaran, D. O'Keeffe, C. Priebe, D. Eyers, B. Shand, and P. Pietzuch. Flowwatcher: Defending against data disclosure vulnerabilities in web applications. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, pages 603–615, New York, NY, USA, 2015. ACM.

[17] NetFlow. 2017. https://en.wikipedia.org/wiki/NetFlow.

[18] V. Pappas, V. P. Kemerlis, A. Zavou, M. Polychronakis, and A. D. Keromytis. Cloudfence: Data flow tracking as a cloud service. In *Proceedings of the 16th International Symposium on Research in Attacks, Intrusions, and Defenses - Volume 8145*, RAID 2013, pages 411–431, New York, NY, USA, 2013. Springer-Verlag New York, Inc.

[19] C. Prakash, J. Lee, Y. Turner, J.-M. Kang, A. Akella, S. Banerjee, C. Clark, Y. Ma, P. Sharma, and Y. Zhang. PGA: Using graphs to express and automatically reconcile network policies. *ACM SIGCOMM Computer Communication Review*, 45(4):29–42, 2015.

[20] Science DMZ: Data Transfer Nodes. 2017. https://fasterdata.es.net/science-dmz/DTN/.

[21] Science DMZ ECAR - WG Technology Spotlight. 2017. https://library.educause.edu/~/media/files/library/2015/11/erb1511.pdf.

[22] Science DMZ Security - Firewalls vs. Router ACLs. 2017. https://fasterdata.es.net/science-dmz/science-dmz-security/.

[23] SciPass: IDS Load Balancer & Science DMZ. 2017. https://globalnoc.iu.edu/sdn/scipass.html.

[24] SGT 6.0 GridFTP. 2017. http://toolkit.globus.org/toolkit/docs/latest-stable/gridftp/.

[25] TCP Issues Explained. 2017. https://fasterdata.es.net/network-tuning/tcp-issues-explained/.

[26] TCP Issues Explained: Packet Loss. 2017. https://fasterdata.es.net/network-tuning/tcp-issues-explained/packet-loss/.

[27] Web10G Kernel Patch and API Packages. 2017. https://www.web10g.org/index.php/software.

[28] A. Zavou, G. Portokalidis, and A. D. Keromytis. Taint-exchange: A generic system for cross-process and cross-host taint tracking. In *Proceedings of the 6th International Conference on Advances in Information and Computer Security*, IWSEC'11, pages 113–128, Berlin, Heidelberg, 2011. Springer-Verlag.