Towards Reducing the Data Exfiltration Surface for the Insider Threat

Bob G. Schlicher, Lawrence P. MacIntyre, and Robert K. Abercrombie
Computational Sciences and Engineering Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831-6085
schlicherbg@ornl.gov, macintyrelp@ornl.gov, abercrombier@ornl.gov

Abstract

Unauthorized data exfiltrations from both insiders and outsiders are costly and damaging. Network communication resources can be used for transporting data illicitly out of the enterprise or cloud. Combined with built-in malware copying utilities, we define this set of tools as comprising the Data Exfiltration Surface (DXS). For securing valuable data, it is desirable to reduce the DXS and maintain controls on the egress points. Our approach is to host the data in a protected enclave that includes novel Software Data Diode (SDD) installed on a secured, border gateway. The SDD allows copying data into the enclave systems but denies data from being copied out. Simultaneously, it permits remote access with remote desktop and console applications. Our tests demonstrate that we are able to effectively reduce the DXS and we are able to protect data from being exfiltrated through the use of the SDD.

1. Introduction

Data exfiltration, is an unauthorized transport of data from within an organization to an external recipient or destination [1]. The term breach is similarly used to describe a security event that results in the disclosure or potential disclosure of data. Data disclosure is more concisely defined as a breach with a confirmation that data was obtained by an unauthorized party [2]. For all the definitions, the common element is that an account on the inside has access to some data and has the privilege and means to copy that data out. An inside account used for performing these unauthorized copies can be either a trusted insider or a malicious intruder who has gained control of that account. Accidental data leakage is caused by inside users that commit mistakes against security policies and practices. This may include naively sending sensitive documents by email, uploading files to online services, and copying files to personal devices [3]. However, the Insider Attack involves one or more malicious individuals who are entrusted with authorized access and has knowledge of the system architecture and contents. The severity of an insider for data exfiltration depends on their knowledge regarding the sensitivity of the data and the active security on the system. For example, a worker in an Information Technology (IT) department typically has more knowledge than an employee in other departments. Insider threats can be anyone - the insider could be a disgruntled employee, an employee leaving for a competitor to the current organization, or a contractor with other motives [4]. In 2014, for the trusted insider, 55% of security incidents involved an abuse of privileges that may have involved a motive for obtaining financial gain from corporate espionage, convenience for processing data, and acting on a grudge. Trusted insiders comprised just over 17% of all confirmed breaches for 2014. Most of the breaches originated from external threats with over 80% of them in 2014 [2]. For cloud computing, the data leakage threat impacts the system integrity and confidentiality giving it a highest priority rating and a medium to high likelihood for its occurrence [5].

With the high potential for being breached, the Gartner Group estimates in the next three years that 40% of large organizations will have a formal plan to respond to attacks. This is an increase that will continue to rise [6] and indicates the change in priorities for dealing with the ongoing breach issues. The underlying premise is that despite all these measures, sensitive data will most likely be exfiltrated and the plan should include a contingency for assessing the damage. Further guidance is provided for minimizing the storage of data with a common sense

This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy (DOE). The United States Government (USG) retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for USG purposes. The DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-publicaccess-plan).

rule of thumb *if your organization does not have the data, it cannot lose it*. Organizations are advised to only store the minimum data required to conduct their business. The guidance advocates monitoring ingress and egress data traffic using Data Leak or Loss Prevention (DLP) technologies [7] that perform outbound data content and flow monitoring [8] with data analytics [3].

Our requirements are driven by the need to protect a big data collection of commercial, proprietary data in support of the Foresight and Understanding from Scientific Exposition (FUSE) program [9]. This big data is comprised of published artifacts including scientific articles, journals, and patents and stored in data repositories in the FUSE network (FUSEnet) system. FUSEnet is a secured cloud that enables remote researchers to run codes that use these repositories. The primary challenge for FUSEnet is that users operate the codes within the system but are prevented from copying the data out of FUSEnet. Even for the slightest convenience, a single artifact is not allowed to be copied or exfiltrated. This exfiltration requirement is fulfilled while simultaneously providing acceptable responsiveness for interactive use including remote graphical user interface (GUI) and console interactions. Users belong to different teams or organizations and hence are required to have their own separate computing systems but use the same data repositories [10].

Consider common sense advice for egress points: *If your organization does not have them, then data cannot be exfiltrated*. However, for most organizations, this is not feasible since most systems are required to communicate over the Internet. In this paper, we focus on specifying a Network Enclave as the basis for establishing a minimum number of access and egress points for a group of computers [11]. We refer to the Data Exfiltration Surface (DXS) as the collection of egress points. We enumerated 26 tests based on egress utilities (i.e., tools) to measure the DXS and effectiveness of the SDD. FUSEnet is configured to separate its computing assets into such enclaves with groups of VMs (virtual machines) in a sub-network that can only communicate with each and are only accessible through a gateway. The purpose behind this is to fulfill the requirement that a FUSEnet team assigned to an enclave is logically and securely separated from other teams. To keep the minimum DXS, the gateway controls the incoming and outgoing traffic with a Software Data Diode (SDD) technology. This SDD is designed to disallow and deny attempts from exfiltrating any data to the outside. The SDD is not a traditional hardware data diode [12]. Rather, it is software designed for data counter-exfiltration operated in the Secured Shell (SSH) layer, integrated with a custom limited shell in a TCP/IP based environment.

Apart from FUSEnet, confidential and sensitive data for your organization is advised to be stored in an enclave system [11]. An enclave is segmented from other parts of the enterprise and consists of systems and network devices with a common security policy [13]. The minimum requirements include Network Intrusion Detection Systems (NIDS), anomaly detection, perimeter protection with the 'deny by default' setting selected including routers, firewalls, and proxies, network demilitarized zone (DMZ), and security compliant applications, services, and operating systems [11].

On the perimeter for the enclave, the router is used for access control for the network with restrictions on the inbound, outbound, and internal connections. The restrictions are for source and destination IP addresses including service protocols such as HTTP, FTP, SSH, Telnet, and DNS. The application proxies are used to provide information from outside the enclave (e.g. web servers) for inside usage. The DMZ is comprised of one or more servers that provide data to users outside the enclave. The DMZ is isolated from the protected enclave. The firewall is prescribed to have the most restrictive rules with the Deny by Default policy, where authorized network services are allowed and everything else is denied [11]. Different types of firewalls include packet filtering, stateful, deep packet inspection, application-aware, and application-proxy firewalls [14].

Based on a set of 200 case studies for data breaches in 24 different countries, the most common techniques for data exfiltration are through Microsoft Windows Network Shares and Native Remote Access. Other techniques include using the native file transfer services such as FTP and capabilities built into various malware [15]. We identify the entire collection of these techniques, benign and malicious, as the DXS. Despite the available requirements and guidance from DISA combined with the knowledge of the DXS, breaches and mass data exfiltrations continue [2]. For all these measures, an unauthorized copy is still possible because the capability to perform an outbound copy is not explicitly denied. For example, SCP is a native application for most Linux distributions. We address this as one of the main requirements for employing the SDD.

To operate within the ORNL Data Center, the FUSEnet system is required to have a security plan that is in compliance with the National Institute of Standards and Technology (NIST) Special Publication 800-53r4 “Security and Privacy Controls for Federal Information Systems and Organizations” [16]. To support this security plan, we assembled a test environment and executed a battery of the exfiltration techniques to challenge our enclave settings for copying data out of the system. With the SDD installed in the enclave gateway, we executed specific tests for ingress and egress operations and verified that the SDD performed as specified. It permitted interactive network traffic for GUI and console operations and it denied all attempts to copy data out of the enclave.

The main contributions of this paper are:

1. Specifying a networked system with a minimal or reduced data exfiltration surface (DXS),
2. Devising and deploying a counter-exfiltration SDD, and
3. Evaluating and verifying the counter exfiltration operations.

The structure of this paper is organized with Section 2 identifying related work in the area of data leak and loss protection including data exfiltration detection. Section 3 describes the security requirements for the FUSEnet system and introduces the features of the data diode software. Section 4 presents the environment for testing and verifying the data exfiltration surface. In Section 5, we review the test results for the counter exfiltration operations. Finally, in Section 6 we provide our conclusions and describe future work.

2. Related works

Network Enclaves are designed for protecting network assets [13] and their requirements [11] are specified for minimizing the DXS in addition to preventing attack penetration. The DXS is enumerated by an example list of file copy utilities combined with a cursory mention of malware that performs data exfiltration [15, 17, 18]. This establishes an initial foundation for tests that can be performed to verify the installation and continued maintenance of an enterprise [18] and a secured enclave. However, the challenge is that full protection from all data exfiltration techniques is considered unfeasible. This is due to the complexity and broad spectrum across applications, protocols, and services. For example, data can be piggybacked undetected within legitimate traffic such as email and web traffic [19]. Analogous to building a repository of malware signatures, we attempt to organize the exfiltration techniques and advocate this is useful for identifying, testing, and eventually reducing the DXS within an enterprise, cloud, and enclave.

For organizing this enumeration, an attempt at categorizing exfiltration techniques is based on a taxonomy with three primary categories of 'Network', 'Physical', and 'Cognitive' (NPC taxonomy). Under the 'Network' category, two classifications are identified as 'Usually Benign', which contains an enumeration of conventional utilities such as HTTP, SCP, and 'Known Malicious', which is an enumeration of exploits such as rootkits, botnets, and DNS poisoning [20]. For the DXS for our paper, the 'Physical' and 'Cognitive' categories are not required.

In contrast to the NPC taxonomy, exfiltration can be organized by communication channel mechanisms for 'overt', 'tunneled', and 'covert' where various techniques or exploits are assigned accordingly. We have called this the 'Channel taxonomy'. Overt communications is the authorized communications and is open, observable, and identifiable. Privacy is included through encryption (e.g. HTTPS, SCP, SSH) to preserve the contents from unauthorized users in normal operations. Tunneled communications is unauthorized and would normally be blocked. The data is communicated within (tunneled) an authorized overt channel and is intentionally cloaked to masquerade as legitimate. Covert communications is embedded in parts of the network or application protocol and encoded in the payload of an overt channel using steganography techniques to hide the sensitive data being exfiltrated [21]. Another useful taxonomy is used to describe a cyber-attack scenario in a decomposition consisting of defining the incident, attack, and the event. Seven components of this Computer and Network Security Incident taxonomy or just Incident Taxonomy include attacker, tool, vulnerability, action, target, unauthorized result, and objectives [22]. The enclave and SDD tests were formulated based on the Channel and Incident taxonomies while incorporating entries from the NPC.

Blocking and filtering legitimate application and utility communication channels with traditional approaches is considered not feasible. Such filtering could block normal traffic and disrupt productive operations. Applying deep-packet inspection may not be effective because an attacker can encrypt the data before it is exfiltrated [19]. Hence, monitoring outbound metadata, flow, and the payload contents combined with analytics form the basis for DLP technologies [17] to counter data exfiltration. Some DLP methods include endpoint document scanning, blocking copies to the clipboard in windows user interfaces, disabling USB drivers, blocking data channels in familiar applications such as Skype and Yahoo Messenger, and inspecting emails and attachments. DLP is advised to be installed within the entire system to prevent leaks and measures should be taken to ensure all devices in the system are involved [3].

The Sensitive Information Dissemination Detection (SIDD) framework is a DLP that detects data exfiltration through traffic identification, content detection, and covert communications detection. SIDD operates at the network boundary and learns the metadata about the application and the data traffic characteristics that includes packet size, timing and structure. In SIDD, the protected network is prescribed to have a controller egress at a gateway where sensitive traffic is discovered and an alert is generated [21]. The SIDD approach was for limited application and traffic types and did not cover a broad range of exfiltration techniques. Also, SIDD is able to determine hidden information in covert communications but not able to fully recover or recognize the contents.

In another approach to prevent exfiltration, systems are modeled based on their computer parameters and behavior using machine learning algorithms. Estimates are made for normal behavior patterns and then anomalies are discovered by comparing against the norm and selected as candidates for unauthorized exfiltration attempts. Using a correlation coefficient method, these candidates are further refined for analysis to reduce mistaken identification [23]. The shortcomings for this are that the technique does not correct for identifying false-positives and false-negatives, which inevitably can flood the alert system at the reporting or data center.

Virtual machine introspection (VMI) is used to obtain activity information of an installed VM through the virtualization layer [24]. The actions of users are observed from outside the VM, presumably unbeknownst to the user or attacker, and include modification, deletion, disabling, moving, copying, pasting, installation, bypassing, and printing. Multiple actions may constitute the malicious activity. For data exfiltration, actions that are specifically observed include the installation and use of malicious tools. In general, the VMI technique [24] uses similar analysis to other methods [23, 25] that involves establishing a statistical normal pattern and then comparing the VMI observations against that norm. The underlying assumption concerning outbound data traffic is that non-threatening usage has a predictable pattern with historical analysis [25].

In the aforementioned techniques, unauthorized data exfiltration is still possible and can bypass the DLP technologies. The underlying assumption is that normal traffic fits a well-defined pattern and anything outside that pattern is potentially an unauthorized activity. This leads to false-positive alerts and can also produce false-negatives, where an actual data theft is undetected. DLP has some limitations that impact its effectiveness for counter exfiltration including difficulty with inspecting encrypted data, processing unknown file formats, recognizing protocols beyond well-known ones such as HTTP, below 100% detection performance, using proper detection algorithms for different situations. The DLP system can be compromised and used to reveal the positions of the sensitive data [3].

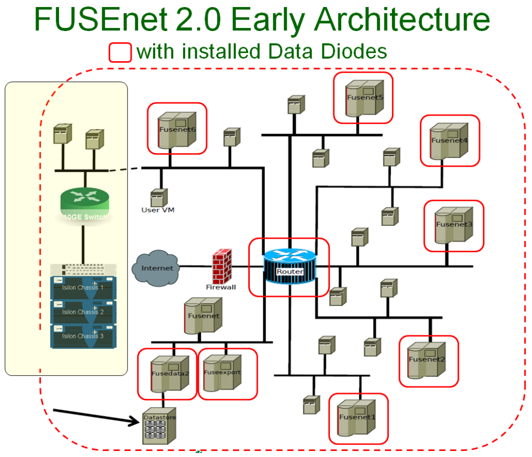


Figure 1. FUSEnet architecture with SDD gateways where each SDD is a gateway for a separate enclave.

To increase the effectiveness against data exfiltration, we introduce the SDD in the enclave gateway. The FUSEnet SDD only monitors outbound traffic to log and report the activity for further processing and for forensic data collections. As a Deny by Default policy, the SDD blocks all outbound copy traffic but permits user interaction protocols for GUIs and console applications. The SDD has the capability to open channels for application traffic such as transaction processing; however, this is out of scope for this paper.

3. FUSEnet system and security requirements

FUSEnet is currently a government system hosted by ORNL at the ORNL Data Center. FUSEnet is a cloud computing environment that stores unclassified, copyright-protected scientific information and provides remote access for approved users to process and analyze the stored data to satisfy the research objectives of the IARPA FUSE Program. The most challenging requirement for the security of the FUSEnet system is to protect the commercial, proprietary data from being leaked to unauthorized parties. This data is supplied by licenses uniquely obtained from multiple commercial data vendors. The data content or repository comprises their collection of published technical articles, patents, and metadata and is prepared in each of the vendor's own native, file formats. Each data repository is an important and core asset for the vendor. Leaking this data could have a serious financial impact on their commercial business.

These vendors include Thomson Reuters, Lexis-Nexis, Elsevier, Institute of Electrical and Electronics Engineers (IEEE), Nature Publishing Group, PubMed Central, and others.

An important tenet for FUSEnet is that data integrity, availability, and protection are maintained [10]. This is accomplished by adhering to enclave designs [16] and integrating additional security measures with an ORNL developed SDD embedded within each FUSEnet gateway. Each gateway allows access to protected data, but with the SDD, data removal by all users is prevented. As necessary, a mechanism for approved data export is built into the system architecture. Also by design, the activities and work products of individual user teams are segregated from each other in the cloud computing virtual environment [10].

Table 1. Incident taxonomy specific to FUSEnet requirements.

Incident Component	Description	Applied to FUSEnet Requirements
Attacker	An adversary that attempts to attack a system to meet an illicit objective. For data exfiltration, this could include an insider with the intention of moving sensitive data out of the system.	Unsuspected insider with data access privileges.
Tool	The means and methods to perform the attack or action. This can involve using native copy utilities, built-in malware copy features, and exploiting a vulnerability to copy data out of the system.	Native utilities available including moving data internally among VMs with SCP and SFTP. Copy data out with SCP, SFTP, FTP, uploads via HTTP.
Vulnerability	A weakness or flaw in the design, implementation, or configuration of a system known only to the attackers. Exfiltration examples could include having FTP installed, a user with unnecessary elevated privileges, and an unmonitored egress channel.	VMs can be installed for any OS with any utilities and configurations. VMs are user-specific and user-managed. The gateway with the SDD is an access for users that may be able to install malware on the gateway.
Action	An act taken by the attacker to perform the attack possibly with the tool(s) to achieve the objective. For exfiltration, an attacker using SCP to copy files to a remote system.	An authorized insider uses one or more of the tools to copy sensitive data out of the system.
Target	The component of the system that is the aim of the attack and presumably is vulnerable. A target can be a server with sensitive data stored in it.	An authorized insider obtains data from the data server and uses native utilities to copy it to an outside server.
Unauthorized Result	An unauthorized consequence of an event. This would involve data leakage in mass quantity.	An authorized insider uses native utilities to copy a large volume of sensitive data to an outside server.
Objectives	The results expected by the attacker. For data exfiltration, the objective is to obtain sensitive data in mass quantity out of the system.	An authorized insider has copied a large volume of sensitive data to a server outside of FUSEnet.

A summary of the FUSEnet benefits and capabilities relevant to this paper includes:

- A data repository of over 100 million published scientific and patent documents;
- Operation of the system with 24/7 and 99.8% availability within domain-specific expertise;
- Support for many types of VMs; and
- Immediate data protection for the repository and custom end-user data.

The primary use case for FUSEnet is that the data remains securely in the enclave. This data is processed by applications and utilities resident in VMs in the enclave. With these restrictions, there is an impact on usability. Users can engage the system through a terminal window accessed by SSH connections and through a remote graphical user interface supporting X11 protocols and Windows VMs. Users can copy and store data into the enclave. Further, user can create documents, write programs, and generate data within the enclave. However, bringing those out requires a controlled procedure described in Section 4. By design, as part of reducing the DXS, conventional means for copying data out are not allowed, which also prohibits useful utilities such as Internet browsers and searches from operating. These are accommodated by other Windows sessions that are not connected the enclave. Hence, users are not required to operate exclusively within an enclave, which provides a balance between usability and security (i.e., prevention of exfiltration).

In accordance with Federal Information Processing Standards Publication 199, “Standards for Security Categorization of Federal Information and Information Systems” [26], FUSEnet systems as a part of ORNL computing are rated at the Low impact level for confidentiality, availability, and integrity because of its specific commercial security needs. FUSEnet resides in the ORNL Open Research Protection Zone, apart from its core zone, which allows for it various and unique cyber security requirements including:

1. Remote access to system,
2. Single-factor authentication with support for two-factor authentication,
3. Self-identification for account establishment for identity management,
4. Protection of the data repositories from data exfiltration and unauthorized copies out,
5. Copy data from outside to inside,
6. User operated and managed applications,
7. Authorization of new accounts, and
8. Creation of new, approved accounts.

3.1. FUSEnet SDD

Observing requirements 1, 4, and 5, the FUSEnet system allows graphical and console interactions with application protocols (e.g. NxClient) that require messages into and out of FUSEnet. File copies from the outside into FUSEnet are permitted. However, through the gateway, enclave design, and as a 'deny by default' policy, file copies are denied from going inside FUSEnet to outside systems. Hence, the major challenge for FUSEnet is fulfilling requirements 1, 4, and 5 simultaneously, while effectively providing the end user with the services they expect to operate remotely in the enclave. To meet this technical challenge, an enclave design is implemented in a VM environment combined with a gateway VM that has a unique capability we call the SDD.

FUSEnet enables remote researchers (outside of ORNL's network) to run codes using the data repositories contained within the FUSEnet system. Their interaction requires a typical two-way data communication. The SDD is designed to prevent any exfiltration of the commercial data while simultaneously providing acceptable responsiveness for interactive use including remote graphical user interface interactions. FUSEnet is configured to separate its computing assets into security enclaves – groups of VMs machines in a sub-network that can only communicate with each and are accessible through a gateway. The purpose behind this is to fulfill the requirement that a performer team assigned to an enclave is logically and securely separated from other teams. The gateway node in each enclave contains the FUSEnet SDD.

The SDD is not a traditional hardware data diode [18]. Rather, it is software for data counter-exfiltration operated in an SSH layer, with a custom limited shell, in a TCP/IP environment. Commercial data diodes work in a receive-only mode, and since they lack the physical capability to transmit, they are provably secure. They are typically deployed for transferring data from the "low" side security to the "high" side for military and intelligence networks. Data does not flow, due to physical constraints, in the other direction. Attractive as this type of data diode may be, it did not meet the FUSEnet security requirements because of the interactive (e.g. GUI) requirements with remote access.

The FUSEnet users are divided into six different teams. Each team is assigned to an enclave where they have their own subnet and their own area on the storage server as shown in Figure 1. A routing filter prevents connectivity between the subnets in order to prevent the teams from accessing the other teams' VMs and data. The routing filter also prevents systems within an enclave from accessing systems outside its enclave including popular systems on the Internet such as Google Drive and Dropbox storage.

All teams have read-only access to the data repositories. Each team remotely accesses FUSEnet through their assigned gateway across the Internet. Each gateway machine may be accessed using SSH or the NX protocol, which has proven to demonstrate the best performance for interactive responses over a wide-area network. In order to ensure that the access is secured, the NX protocol is tunneled over SSH.

Once access is granted through their gateway, each team creates and manages their own VMs and runs their codes within their enclave. FUSEnet is designed so that VMs within an enclave have no direct connections with any system via the Internet, and therefore are not capable of copying data outside FUSEnet. Accessing FUSEnet is a single authentication but involves a two-step process. Users of FUSEnet are required to first login to the gateway where the SDD resides using SSH or the NX protocol. From the gateway, users access their VMs using SSH, NX, or the remote desktop protocol (Windows).

Each gateway machine is configured with the SSD supporting SSH, a modified shell severely restricts the commands available to the user remote terminal shells.

3.2. Authorized data exporting

The SDD prevents direct copying of data from inside to outside of a secured FUSEnet enclave. However, there are occasions when users need to export files for their own purposes to their own remote site. To accommodate this need, FUSEnet provides a two-step process for such approved data exporting or exfiltration. Users are expected to copy the files they wish to export to a designated data share, identified by the FUSEnet staff. The requestor then registers a request for export. For FUSEnet, the Redmine ticket tracking system [27] is employed on the publicly accessible FUSEwiki. An authority from the FUSEnet team manually reviews the request, and if necessary scans the contents to protect the licensed intellectual property. If acceptable, the authority provides approval for export. After approved, a staff person from the FUSEnet Data Center copies the data to another special area on the team's gateway machine where the users can access the data from an outside network. Once the export is complete, the requester is expected to update the Redmine ticket indicating that the export has been completed. The archival file is deleted from the shared FUSEnet VM directories, and closes out the ticket.

4. Testing and verifying DXS and the SDD

Compliance with NIST 800-53r4 [16] calls for organizations to execute due diligence with regards to the breadth and depth of information security and risk management. Further constraints on the FUSEnet system led to the implementation of the aforementioned SDD that enables remote users to engage the system with typical interactive usage but with data copying capabilities disabled. The Incident Taxonomy provides seven components for describing the elements involved in security attack incidents [22]. Table 1 provides a summary for applying this taxonomy for unauthorized data exfiltration and applying that specifically to FUSEnet requirements.

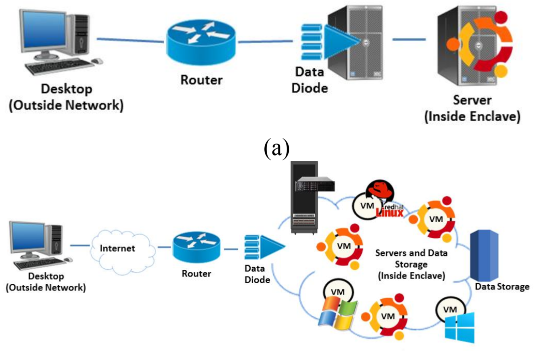
To support FIPS 800-53r4 compliance and using the Incident taxonomy we constructed for FUSEnet, two networked scenario environments were established for testing and verifying the SDD operations and enclave. These environments are illustrated in Figure 2 as the unit test (Figure 2a) and virtual enclave configuration (Figure 2b). Similar security tests using VMs were used to verify the operations for DLP technologies [3, 17], while other tests focused on network security testing through simulated data exfiltration techniques with an embedded testing agent [28]. In one set of tests, authentication, session management, and access controls were challenged to identify vulnerabilities for breaching the system [3]. Using a testing agent within a network, messages that contain simulated data, for example credit card and personal information, were transmitted through application protocols that natively copied data and transmitted with non-standard protocols to discover exfiltration channels. These tests were semi-automated. Messages received on the outside, if that happened, were compared with the original contents on the inside [28] to determine the success of the exfiltration technique. Considering the Channel taxonomy [21] the aforementioned test processes, and other sources [15, 17, 29], we compiled a list of data exfiltration techniques as shown in Table 2.

Table 2. Data exfiltration techniques for testing the DXS.

Data Exfiltration Techniques	**Channel Taxonomy and Capability**
Native Remote Access Applications	Overt: SSH, SCP, SFTP, FTP, HTTP, HTTP upload, HTTPS, IRC, windows copy-paste
Native Remote Networking	Tunneled: ICMP, VPN, SSH, SMTP, DNS
Remote Access Applications	Overt: Instant Messaging, Microsoft Windows Network Shares
Malware Embedded Capability	Overt, Tunneled: FTP, IRC, SMTP Covert: text in media files, files archived inside pictures, pictures in other media
Vulnerabilities	Overt: SQL Injection Covert: Encrypted backdoor

The "Tunneled" native remote networking involves data embedded in the protocol that is ignored by the protocol handlers but extracted by the destination. For example, DNS can be exploited for data exfiltration based on tunneling and forming a covert channel for transferring files, usually in segments, that are eventually assembled at the receiving node [29]. The most common techniques for data exfiltration are through MS Windows Network Shares and Native Remote Access Applications with a combined 56% of usage for a set of recent data breaches [15]. This data shows that attackers are using the natively installed utilities and the sensitive data is copied out undetected. Regardless of their popularity, all of these methods are effective to some extent and must be denied from getting data to the outside of the secured enclave. Native Remote Access Applications also includes file copy utilities such as a remote desktop (e.g. NX Client, X2Go, Windows Remote Desktop) file-copy and drag-and-drop, in addition to the remote copy utilities such as secure copy (scp). For all the tests for FUSEnet, the simulated attacker, originating from the outside network desktop computer as shown in Figure 2, had a legitimate account on one or more systems within the enclave and also had the permissions to launch their own VMs within the enclave. Penetrating the network was out of scope for this testing and therefore was not necessary and not performed. The simulated attacker ran tests using some of the data exfiltration techniques listed in Table 2. A description of the testing is as follows:

1. Testing was performed using some of the techniques identified in Table 2.
2. Testing was performed within the bounds of FUSEnet system requirements. Selected requirements that constrained this testing included:
   a. The SDD was deployed in 100% blocking mode as 'Deny be Default' – ad hoc tunneling was disabled (this is by default) but interactive user protocols between the outside and inside were active.
   b. Computational processes that access and process the data repository(s) were operated exclusively inside the protected enclave.



(a)

(b)

Figure 2. SDD Testing in (a) unit test configuration and (b) virtual enclave test configuration.

Table 3. SDD Test Matrix.

Source Location	Test Id	Tool	Purpose	SDD Response / Metric	Verification Status
Outside Network	O-1	MS Windows Network	Connect to diode	Allow	Verified
	O-2	MS Windows Network	Copy file from diode	Deny	Verified
	O-3	MS Windows Network	Copy file from inside VM	Deny	Verified
	O-4	MS Windows Network	Copy file to diode	Allow	Verified
	O-5	Native FTP	Copy file to/from diode	Deny	Verified[1]
	O-6	Native FTP	Copy file to/from inside VM (FTP server VM)	Deny	Verified[2]
		Native remote access with ssh, scp:			
	O-7	ssh <diode>	Connect to diode	Allow	Verified
	O-8	scp <diode>: file .	Copy file from diode	Deny	Verified
	O-9	scp file <diode>:	Copy file to diode	Allow	Verified
Diode	D-1	Malware FTP[4], Malware IRC2, Native FTP, Encrypted backdoor, HTTP upload	Copy file to outside	Deny	Verified[4]
	D-2	ssh <outside>	Connect to outside node	Deny	Verified
	D-3	scp <inside>	Connect to inside node	Allow	Verified
	D-4	scp <outside>: file	Copy file from outside node	Deny	Verified
	D-5	scp file <outside>:	Copy file to outside node	Deny	Verified
	D-6	scp <inside>:file	Copy file from inside enclave	Deny	Verified
	D-7	scp file <inside>:	Copy file to enclave	Allow	Verified
Inside Protected Enclave	I-1	Malware FTP[2], Malware IRC[2], Encrypted backdoor, HTTP upload	Copy file to diode	Deny	Verified[4]
	I-2	Malware FTP[2], Malware IRC[2], Encrypted backdoor, HTTP upload	Copy file to outside	Deny	Verified[5]
	I-3	MS Windows Network	Connect to diode, Copy file from diode, Copy file to diode	Deny	Verified[6]
	I-4	MS Windows Network	Copy file to outside	Deny	Verified[3]
	I-5	ssh <diode>	Connect to diode	Deny	Verified[4]
	I-6	scp <diode>:file .	Copy file from diode into enclave	Allow	Verified
	I-7	scp file <diode>:	Copy file to diode	Deny	Verified[4]
	I-8	Native FTP	Copy file to/from diode	Deny	Verified[4]
	I-9	Native FTP	Copy file to outside	Deny	Verified[3]
	I-10	SQL Injection	Copy data to the outside	Deny	Verified[7]

Node names are enclosed in angle brackets <>

1. FTP is not installed and is prevented from being installed by the restricted SDD shell.
2. Systems outside the enclave are denied from directly accessing any ports and hence, servers on enclave VMs because there is no network route to the enclave VM. All user work is performed through the SDD.
3. Malware was simulated as a Java program with embedded FTP and IRC protocols. No Malware was used on the ORNL network for this testing.
4. FTP, HTTP client and server, and any other native and installable software are prevented from being installed and used because of the restricted SDD Shell. However, a root exploit on the SDD OS (e.g. Ubuntu) could facilitate installation and operation with sophisticated hacking – eventually files could be copied from the SDD. Nevertheless, the possibility of a root exploit on a router (e.g. Cisco) is comparable to that of the SDD Gateway.
5. All VMs within a secured enclave do not have access to systems or networks outside of the enclave.
6. All VMs within a secured enclave do not have access to the SDD VM.
7. Direct SQL calls from the outside to a server running on an enclave VM are not permitted, since there is no network route to any enclave VM. SQL injection is possible from within the enclave; however, the data remains confined within the enclave due to the DD and the other described security measures.

c. All GUI operations that executed from a remote client interacted with the VMs inside the protected enclave through a FUSEnet gateway that contained the SDD. By network settings, VMs inside a protected enclave were not permitted to interact with VMs in other enclaves.

d. Copying into the FUSEnet system was unconstrained.

FUSEnet exfiltration testing originates from an outside desktop system with an authorized account for the FUSEnet system. In Figure 2, three source locations are identified for targets of an exfiltration attack including Outside Network, in the SDD system, and Inside the Protected Enclave. In the unit test configuration in Figure 2a, the components included a remote desktop connected locally through a router that connected a gateway machine that hosted the SDD. The desktop system was either Windows or Linux that possessed the capability to login, access remote systems, use remote graphical software, and perform file transfers. The gateway machine's operating system was Ubuntu Linux. Through the SDD, an authorized user gained access to the server, which was an Ubuntu machine running an App Server and Data storage service.

By proper network settings, this server system was restricted from directly communicating with any other machine in this configuration. In the virtual enclave test configuration in Figure 2b, the components included the same desktop outside the network. However, in this setup it was connected over the

Internet to the gateway router that connected the SDD gateway deployed as a VM. This gateway was part of a virtual network where the virtualization was implemented with VMware. Through network configuration, all VMs within this network were not permitted to communicate with machines outside of the network or with the SDD. However, through the SDD, an authorized user had access to the VMs and the data storage for pushing data in and operating in the network. The enclave and SDD configuration were setup to prevent data from getting out.

Following the FUSEnet access procedure, the desktop user securely accessed the SDD, which served as the gateway to its corresponding enclave. The user performed the access with independent tests using a Linux-type terminal window and with an NX Client Tool (e.g. OpenNX) for a graphical user interface that engaged both Windows and Linux based VMs. After successfully gaining access to the gateway, the user then logged into one or more of the VMs in the enclave. From a terminal window, the login was performed with SSH. From the NX Client Tool, the login was performed by one of three applications: 1) Remote Desktop for connections with Windows VMs, 2) QTNX for connecting to a Linux VM running an NX server, and 3) the gateway shell called ORNL Shell.

The SDD Text Matrix, provided in Table 3, organizes each test by source location, then by the command or software utility, purpose, the expected SDD response, and an additional verification status. The verification status is used to indicate the result of the tests that were performed. These tests were performed with the configurations described in Figures 2a and 2b, with the data exfiltration techniques within each of the systems as described with the aforementioned procedure.

No malware was executed in these environments and therefore was not installed or operated within the ORNL computing environment. Instead, this was simulated using a custom written, Java-based program with embedded FTP and IRC protocols with no malicious code. The program was executed from an approved account on the VM or VMs within the secured enclave.

5. Experimental results

All tests were executed individually. Table 3 results indicate that the counter exfiltration operations were verified, access protocols were permitted, and interactive user protocols, operated successfully. The SDD and the secured enclave were successful at denying data exfiltration of the DXS techniques from Table 2. In Table 3, test results indicate a positive verification with 'Verified" in the Verification Status column. These entries are 'Verified' and require no additional explanation for the test having been verified. Some entries require an explanation or a caveat for the verification and are provided in the referenced notes. If there were any unsuccessful test results, the Verification Status would indicate 'Not Verified' with a footnote that offered an explanation or further details for the error.

6. Conclusions and future work

Recent data and reports indicate that data breaches are increasing. Organizations are encouraged to have formal plans for a data leakage incident that could involve sensitive data. In this paper, we have provided an approach for protecting this sensitive data in a cloud operation that is exposed to remote, Internet users. Our approach is to host the data in repositories protected in a secured enclave with access control exclusively through a gateway with a new capability called a SDD. With this approach, the Data Exfiltration Surface (DXS), (i.e., the means to copy or stream data out of the system) is minimized. The SDD has been installed and used for the FUSEnet System at the ORNL Data Center to protect an estimated 100 million commercial documents. The SDD is software designed for counter-exfiltration to disallow and deny attempts from leaking any data outside of the protected enclave while simultaneously permitting user interactions through remote graphical user interfaces and consoles. We applied the Channel taxonomy and the *Computer and Network Security Incident* taxonomy to assist with our development of a test environment for the FUSEnet enclaves and to enumerate utilities and malware capabilities that comprise a DXS for the enclave. Our tests and results demonstrate that our approach effectively reduces the DXS for an enclave, users can engage the FUSEnet system and control their own VMs, and the SDD successfully denies file copies from inside the enclave to the outside. Future work includes enabling channels in the SDD for specific data traffic, traversing through the SDD, and integrating DLP technologies for processing the data in motion through those channels, and red-teaming, perhaps involving system administrators.

7. Acknowledgments

This research was supported by the Intelligence Advanced Research Projects Activity (IARPA) via the Department of Energy (DOE). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOE, ORNL, or the U.S. Government.

8. References

[1] "When Breaches Happen: Top Five Questions to Prepare For," in *SANS Institute InfoSec Reading Room*, ed: SANS, 2012.

[2] "2015 Data Breach Investigations Report," Verizon Enterprise Solutions 2015.

[3] T. Torsteinbø, "Data Loss Prevention Systems and Their Weaknesses," Masters Thesis, Department of Information Technology, University of Agder, 2012.

[4] E. Kowalski, D. Cappelli, B. J. Willke, and A. P. Moore, "Insider threat study: Illicit cyber activity in the government sector," US DHS, US Secret Service, CERT, and SERI-CMU, Tech. Rep 2008.

[5] A. U. Khan, M. Oriol, M. Kiran, J. Ming, and K. Djemame, "Security risks and their management in cloud computing," in *IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom)*, 2012, pp. 121-128.

[6] D. Woodburn, "Gartner: Firewalls and AV no longer enough," ed: CRN 2015.

[7] "Data Protection & Breach Readiness Guide. Online Trust Alliance," Online Trust Alliance (OTA) 2014.

[8] G. Lawton, "New Technology Prevents Data Leakage," *Computer,* vol. 41, pp. 14-17, 2008.

[9] D. A. Murdick. (2011, January 9). *Foresight and Understanding from Scientific Exposition (FUSE)*. Available: http://www.iarpa.gov/index.php/research-programs/fuse

[10] B. G. Schlicher, J. J. Kulesz, R. K. Abercrombie, and K. L. Kruse, "A Computing Environment to Support Repeatable Scientific Big Data Experimentation of World-Wide Scientific Literature," in *15th International Conference on Scientometrics and Informetrics*, Istanbul, Turkey, 2015.

[11] "Enclave Security Technical Implementation Guide (Enclave STIG), V4R5," DISA Field Security Operations, 2014.

[12] M. Stevens and M. Pope, "Data Diodes," Electronics and Surveillance Research Laboratory, DSTO-TR-0209, July 1995.

[13] (2009). *TrustCC -Network Enclaves – Enhanced Internal Network Segmentation*. Available: https://trustcc.wordpress.com/2009/08/13/network-enclaves-%E2%80%93-enhanced-internal-network-segmentation/

[14] (2015, June 12). *What Is Firewall Security? Dell SecureWorks*.Available: http://www.secureworks.com/resources/articles/other_articles/firewall-security/

[15] N. J. Percoco. (2010), Data Exfiltration: How Data Gets Out. *CSO Online*. Available: http://www.csoonline.com/article/2135266/network-security/data-exfiltration--how-data-gets-out.html

[16] "Security and Privacy Controls for Federal Information Systems and Organizations," National Institute of Standards and Technology (NIST) Gaithersburg, MD NIST Special Publication 800-53 Revision 4, 2013.

[17] A. Shabtai, Y. Elovici, and L. Rokach, "Data Leakage Detection/Prevention Solutions," in *A Survey of Data Leakage Detection and Prevention Solutions* A. Shabtai, *et al.*, Eds., ed New York: Springer, 2012, pp. 17-37.

[18] C. A. Nilsen, "Method for Transferring Data from an Unsecured Computer to a Secured Computer," USA Patent 5,703,562, 1997.

[19] K. Brancik and G. Ghinita, "The Optimization of Situational Awareness for Insider Threat Detection," in *First ACM Conference on Data and Application Security and Privacy - CODASPY'11*, 2011.

[20] A. Giani, V. H. Berk, and G. V. Cybenko, "Data Exfiltration and Covert Channels," in *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense V*, 2006.

[21] Y. Liu, C. Corbett, K. Chiang, R. Archibald, B. Mukherjee, and D. Ghosal, "SIDD: A Framework for Detecting Sensitive Data Exfiltration by an Insider Attack," in *42nd Hawaii International Conference on System Sciences*, 2009, pp. 1-10.

[22] J. D. Howard and T. A. Longstaff, "A Common Language for Computer Security Incidents," Sandia National Laboratories, Albuquerque, NM and Livermore, CA, Technical Report SAND98-8667, 1998.

[23] R. Ramachandran, S. Neelakantan, and A. S. Bidyarthy, "Behavior model for detecting data exfiltration in network environment," in *2011 IEEE 5th International Conference on Internet Multimedia Systems Architecture and Application (IMSAA)*, 2011, pp. 1-5.

[24] M. Crawford and G. Peterson, "Insider Threat Detection Using Virtual Machine Introspection," in *46th Hawaii International Conference on System Sciences (HICSS)*, 2013, pp. 1821-1830.

[25] N. R. Suresh, N. Malhotra, R. Kumar, and B. Thanudas, "An integrated data exfiltration monitoring tool for a large organization with highly confidential data source," in *Computer Science and Electronic Engineering Conference (CEEC), 2012 4th*, 2012, pp. 149-153.

[26] "Standards for Security Categorization of Federal Information and Information Systems," FIPS PUB 199, 2004.

[27] (2015). *Redmine*. Available: http://www.redmine.org/projects/redmine

[28] T. T. Hawthorn, N. Miller, and J. LoSapio, "Data exfiltration attack simulation technology," USA Patent US 8,782,796 B2, 2014.

[29] K. Born, "Browser-Based Covert Data Exfiltration," in *9th Annual Security Conference*, Las Vegas, NV, 2010.