Using Splunk to Detect DNS Tunneling

Steve Jaworski
Abstract

DNS tunneling is a method to bypass security controls and exfiltrate data from a targeted organization. Choose any endpoint on your organization’s network, using nslookup, perform an A record lookup for www.sans.org. If it resolves with the site’s IP address, that endpoint is susceptible to DNS Tunneling. Logging DNS transactions from different sources such as network taps and the DNS servers themselves can generate large volumes of data to investigate. Using Splunk can help ingest the large volume of log data and mine the information to determine what malicious actors may be using DNS tunneling techniques on the target organizations network. This paper will guide the reader in building a lab network to test and understand different DNS tunneling tools. Then use Splunk and Splunk Stream to collect the data and detect the DNS tunneling techniques. The reader will be able apply to what they learn to any enterprise network.
1. Introduction

Domain Name System (DNS) is described as the Internet phone book. (Gonyea, 2010) DNS maps a host and domain name such as www.sans.org to an IP address 66.35.59.202. In this case, the host is www and the domain is sans.org. DNS permits the Internet user to access websites using names instead of IP addresses. The Internet can operate without DNS. However, users would need to know the IP Address of the website, email server or some other service they want to access. IPv4 supports over four billion IP addresses and IPv6 supports over three hundred and forty undecillion. That is an impossible amount of IP addresses for someone to memorize. Due to the sheer size of the Internet, there needs to be an effective method for users to navigate the Internet. DNS provides this service.

As organizations continue to secure their networks and assets by implementing Defense in Depth Strategies, malicious actors still find ways to circumvent the controls. (National Security Agency, n.d.) DNS is often overlooked for security because no one considered using the protocol for data transmission. It was determined as early as 1998 that transferring data over the DNS protocol was possible. (Farnham, 2013) DNS tunneling software has been developed and available to the public since that time. Organization’s internal DNS servers are often dependent on upstream DNS servers from their Internet Service Providers or companies that provide DNS services. If the DNS provider is not monitoring their DNS servers for malicious domains, the malicious domain can then be resolved using the organization’s DNS server. It is up to the organization to secure and monitor their DNS services. Without monitoring of an organization’s DNS services, a malicious actor could tunnel any data in and out of the network undetected.

Preventing all DNS tunneling is not possible, creating a high-risk of successful data exfiltration, but it can be limited. (Nadkarni, 2014) If a malicious actor chooses to exfiltrate data using a few DNS packets every so often over time, it is very hard to detect. Data that can be leaked using a DNS tunnel could be intellectual property, trade secrets, customer records and employee data. A DNS tunnel requires software on the victim machine to work. The malicious actor is able to bypass all of the organization’s security controls and successfully establish a persistent backdoor with a DNS tunnel.

Steve Jaworski, jaworski.steve@gmail.com
Since inhibiting all DNS tunneling is not likely, it is important to monitor and log all the DNS services on the network. DNS events and logs are available from multiple sources such as DNS servers, Intrusion Detection Systems, proxies, hosts on the network, and firewalls. To detect malicious DNS activity effectively, all the event and log data should be sent to a central system for analysis. The data can be analyzed using custom scripts. This approach will take time. If the analyst wants to visualize the data, another tool will have to installed and configured. The analyst will have to write their own statistical function application or find an existing tool to meet their needs. With Splunk, the analyst can easily ingest data from multiple DNS related sources, perform statistical analysis on the data, visualize the data, share the results with other analysts, and create alerts. Splunk can also scale with the size of the organization. Depending on the size of the organization, DNS events, and logs can be in the millions if not billions per minute, hour or day. The organization will require a tool that scales to a large volume of data and quickly find notable DNS security events within the data.

2. DNS

To understand how a DNS tunnel can be used to bypass the network’s security controls, it is important to understand how DNS works. When a user wants to access www.sans.org, their computer will first query its local DNS cache. If there is no result found, it will then query its configured upstream DNS server. The user’s ISP, their company, or another public DNS service may operate the upstream DNS server. The upstream DNS server will check its local cache for the answer. If it does not have the answer, it will query the root DNS servers or another upstream DNS server if configured. The root DNS servers will then direct the querying DNS server to the appropriate top level domain (TLD) DNS server, in this case the TLD server for .org. The .org TLD DNS server will then instruct the querying DNS server to the sans.org authoritative DNS server. The sans.org DNS server will resolve the IP address for www.sans.org. To improve the response time of resolving the query for www.sans.org, both the client and requesting DNS servers will cache the result based on the time to live (TTL) configured by the sans.org domain administrator. (Gonyea, 2010)
2.1. Record Types

DNS uses record types to determine the requested service. There are eighty-three record types registered with the Internet Assigned Numbers Authority. (IANA, 2016) The different record types help the Internet user find web pages, mail servers, DNS servers and a variety of other services.

2.1.1. Common Record Type

Some of the common record types used in DNS are the A, PTR, MX, CNAME, TXT, NS, and SOA records. (Faudle, 2015) When analyzing DNS logs and packets, the analyst will see these records the most often. The A and PTR record are required to perform a forward and reverse lookup. The A record maps a host and domain name to the IP address, for the forward lookup. The PTR record provides the IP address to host and domain name, for the reverse lookup. The MX record provides the host and domain mapping for mail servers. The CNAME (Canonical Name) record is used as an alias to other A or CNAME records. The NS (Name Server) record is used to tell other DNS servers and clients who the authoritative server is for a particular domain. The record type SOA (Start of Authority) provides information such as the current version of the domain’s records. The TXT (Text) record stores any text string. The most popular use of a TXT record is to store IP address and domains of valid email senders for a particular domain. The txt record type is also known as the Sender Policy Framework (SPF) record.

2.1.2. Uncommon Record Types

The seven common record types can still be used for DNS tunneling. The analyst will have to spend more time evaluating the common record types to find tunnels because there will be a larger amount of data to search. However, the remaining seventy-six record types can be identified more quickly as red flags on the organization’s network. It is important for the analyst not to assume the uncommon records are always malicious. Uncommon records that may appear are AAAA, AXFR, DNSKEY, but they are valid. The AAAA record resolves domain names for the 128-bit IPv6 IP address. The AXFR record indicates a zone transfer. A zone transfer could be an entirely different security issue for the organization. Unless the organization explicitly allows zone transfers for

Steve Jaworski, jaworski.steve@gmail.com
specific hosts, this is a red flag someone may be performing active footprinting of the organization’s network. DNS zone transfers should be limited and restricted to prevent someone from easily being able to identify hosts and mapping the organization’s network. (Lau, 2003) The DNSKEY record is for Domain Name System Security Extension (DNSSEC) identification. DNSSEC is the signing of domain names and records to validate their authenticity against any modification by a third party. (ICANN, 2014) This record type could make the common list someday, but not every organization in the world has adopted DNSSEC. As the world continues to adopt DNSSEC, this record type will become more common.

3. DNS Tunneling

One purpose of DNS tunneling is to bypass hotspot security controls at airports or hotels to acquire free Internet access. (Farnham, 2013) A more malicious reason for DNS tunneling is to exfiltrate data from an organizations network. Data exfiltration has more of a negative impact to an organization than stealing bandwidth. Once it is discovered that data has been exfiltrated from the network, the organization will incur the cost of incident response services, compliance fines, and public media management. Even worse is intellectual property loss or customer data that negatively affects the business. (Cruz, 2013) DNS tunneling techniques still works well because DNS is not monitored as well as other applications or systems on the network because DNS is blindly trusted. (Branscombe, 2015)

For tunneling to work, a client-server model is used. The client is typically behind the organization’s security controls and the server is located somewhere on the Internet. The DNS communications between the client and server occur over the organization’s own DNS infrastructure and any other public DNS servers. Since this is a client-server model, any type of traffic can be sent over the tunnel. Some tunnel applications even provide encryption.
3.1. Tunneling tools

Two different tunneling tools will be analyzed, Iodine and Dnscat2. Erik Ekman and Bjorn Andersson maintain the Iodine application. Iodine is similar to a client-server application. There is the server executable “iodined” and the client executable “iodine”. Iodine creates tunnel interfaces on the client and server. Any traffic can be sent over the tunnel and can be initiated from the client or server. (Ekman & Andersson, 2014)

Dnscat2 is also a client-server application. The difference is the Dnscat2 software operates similar to command and control software. Dnscat2 also encrypts traffic verses Iodine’s encoded traffic. The Dnscat2 application is not designed to bypass restricted access on hotel or coffee shop networks to gain free Internet access. Ron Bowes actively develops and maintains Dnscat2. (Bowes, 2015) Both tunnel applications can bypass upstream DNS servers for data transfer if the organization’s perimeter allows unrestricted IP address access outbound. By not having to send requests thru the DNS infrastructure, data transfer rates are even faster.

3.2. Tunneling Example

An organization has implemented egress filtering on their perimeter making it more challenging to exfiltrate data from the network. (Brenton, 2006) The egress filtering is so restrictive the internal hosts cannot directly access the Internet. An authenticated proxy is required to access the Internet. The malicious actor manages to compromise an internal host by social engineering a user to install the DNS tunnel software. In Figure 1, the DNS Tunnel Client is installed on the compromised machine and is configured to use the organization’s internal DNS server (See the arrow labeled with 1). The internal DNS server forwards non-cached requests an upstream/public DNS server. The firewall only allows TCP/UDP on port 53 from the Internal DNS server to the upstream/public DNS server (See the arrow labeled with 2). Since the attacker has a registered domain name for their attack, all the DNS requests are forwarded to the DNS...
Using Splunk to Detect DNS Tunneling

Tunnel Server (See the arrow labeled with 3). If the upstream/public DNS server does not have in its cache the attacker’s domain name, it will perform the required steps to check with its configured upstream forwarder or root servers to resolve the domain name.

Taking a deeper look into DNS Tunneling with software Iodine, the software creates tunnel interfaces on the client and server. The Iodine software follows the exact DNS path described in the previous paragraph and figure. The malicious actor is then able to send data back and forth between the client and server. The malicious actor starts Iodined on their DNS tunnel server in Figure 2. The -f keeps the software in the foreground, -P is the tunnel password, the IP address is the tunnel interface, and t1.security.local is the attacker’s domain.

Steve Jaworski, jaworski.steve@gmail.com
On the client, the malicious actor starts Iodine. The -f is used to keep the software running in the foreground, same as the server, the -P for the tunnel password is specified, and the required destination domain. Figure 3 shows its local tunnel interface is 10.10.10.3 and the tunnel server is 10.10.10.1. The maximum transmission unit (MTU) size in use is 1130 bytes due to EDNS0 extension available for use by the organization’s DNS server. To put the MTU size of 1130 bytes in perspective, Ethernet’s standard MTU is 1500 bytes. Being able to add more data into a single DNS request can aid the malicious actor in operating undetected.

To see the tunnel interfaces created, execute the command ip or ifconfig. Notice the subnet for the tunnel is a /27 or 255.255.255.224 in Figure 4 and
Figure 5. While this subnet size supports thirty hosts, Iodine supports sixteen clients per tunnel server.

With the tunnel up, the attacker can transfer any data between the client and server. The Iodine client sends a keep-alive every four seconds. Looking at this Splunk graph in Figure 6, it shows low connectivity and then a sudden spike in traffic.
The spike in traffic is due to the malicious actor transferring a file from the client to the server over the tunnel. With a busy DNS server, this spike may not be as obvious. The illustration displays how many DNS packets are required to transfer data. In Figure 7, is the file copied over the DNS tunnel using SCP.

Reviewing the file on the malicious actor’s DNS server, the file transferred was successful. Refer to Figure 8, running md5sum shows no modification to the file occurred in transit.

Digging a little deeper into what a DNS tunnel request looks like, in Figure 9 the first record is a keep-alive and the second record is the data transfer. The keep-alive

Steve Jaworski, jaworski.steve@gmail.com
requires a very short DNS name while the data transfer request uses the maximum length of a DNS record.

**Figure 9**

![DNS Tunneling Example]

---

Steve Jaworski, jaworski.steve@gmail.com

© 2016 The SANS Institute Author retains full rights.
4. Splunk

DNS Tunnels will generate thousands upon thousands of requests to a specific domain, use uncommon recorded types, send keep-alives, or have very long host names. A tool such as Splunk can help capture and analyze all the DNS data generated by an organization. Splunk is commercial software used to consume large datasets and provide keyword searching capabilities, dashboarding, reporting, and statistical analysis. Splunk’s search speed is based on MapReduce developed at Google in 2004. (Sorkin, 2011) Splunk can consume almost any type of data. Splunk has many built-in field extractions for common data such as Windows event logs and Apache web logs. A field extraction is simply a way of normalizing data into common fields, making it easier to analyze. Example field extractions are time, hostname, IP address, destination, etc. If a prebuilt field extraction does not exist, the Splunk administrator can write their own. Field extraction is important because it provides context for an event. One of the most important field extractions is time. The organization needs to find when an event occurred. Another important field extraction is the IP address. With Splunk, the administrator can query the index for X IP addressed during Y timeframe. Field extractions also make it easier for the analyst to perform statistical analysis on the data.

Splunk offers a Free Enterprise version with a 500-megabyte data limit every 24 hours. Some limits of the free license version are no login credentials and real-time alerts. There are no restrictions on collecting different types of data. Splunk Enterprise has no operational restrictions and is licensed by how much data is collected in a 24-hour period. The license size can be as small as one gigabyte and as large as multiple terabytes. For the purpose of this lab, the Free Enterprise version is more than sufficient.

Splunk by itself is an extremely powerful platform, and by using Splunk apps, Splunk can be even more powerful. In Splunk, the name app is short for application. A Splunk app is a prebuilt package for specific functions or a defined data set. For example, a firewall vendor develops a Splunk App for their firewall platform. The app may contain prebuilt field extractions, dashboards, reports, lookup tables, and alerts. (Splunk, n.d.) The Splunk analyst saves time by not having to create the vendor specific elements themselves. Apps also allow analysts who are not Splunk experts to start extracting value from the data.

Steve Jaworski, jaworski.steve@gmail.com
Sometimes there is not a way to retrieve data from an endpoint and send to Splunk. It could be due to a technical, political, or security issue. Splunk has developed an app called Splunk Stream. The app collects data directly off the network wire and decodes it. In the case of analyzing DNS packets, Splunk stream can use a mirror port and collect all DNS transactions off the network wire. The analyst can then query the data looking for specific events and then alert or report on them. Stream installs on a Splunk server and universal forwarder. The universal forwarder is installed as a data collection agent on servers and does more than just run the Stream app. The universal forwarder uses the stream technology add-on app, which collects and forwards the data to the Splunk server. Think of using the universal forward agent as sensors around the network. The Stream app for the Splunk server contains a collector and dashboards displaying network metrics.

5. Lab Layout

Building a lab to experiment with DNS tunnels provides a safe environment to learn how they work. The appendix contains instructions on how to build the lab and duplicate the results discussed in this paper. The lab consists of a single PC running open source and free to use commercial software. The core operating system is hypervisor VMware ESXi 6.0. The guest operating systems are Debian 8.X and Ubuntu 14.04 LTS for emulating the Bind DNS server and DNS tunneling servers. The client workstation is Debian 8.X and runs the DNS tunneling clients. A virtual pfSense firewall separates every server and client into a dedicated subnet. All log traffic is being sent to the free version of Splunk also running on the server as a virtual machine. The virtual switch contains a port group for each firewall segment. The individual port groups support promiscuous mode to allow the Splunk Stream app to collect data from the network segments.

The lab supports two different DNS tunnel application simultaneously. Iodine and Dnscat2 were chosen for the lab because they are easy to configure and operate. Domain t1.security.local is for the Iodine tunnel and t2.security.local is for Dnscat2. Each tunnel application also provides a wide range of options on how to traverse DNS. Choosing different options can reduce the likelihood of tunnel detection. Splunk analyzes the
tunnels using logs from the pfSense firewall, Bind DNS server, and Splunk Stream App. See Figure 10 for the lab topology.

Figure 10

6. Detection

There are multiple detection techniques to find DNS tunnels. Greg Farnham’s paper “Detecting DNS Tunneling” written for the Global Information Assurance Certification (GIAC) Certified Intrusion Analyst (GCIA) outlined several ways to detect DNS tunneling within an organization’s network. Farnham described how to detect tunnels using a pseudocode approach so as not to reveal the commercial system used to perform the detection. The two main detection techniques outlined were payload analysis and traffic analysis. Payload analysis comprises of various techniques such as the size of a DNS request and response, the entropy of the Fully Qualified Domain Name (FQDN), statistical analysis, infrequent record types, and unauthorized DNS servers. (Farnham, 2013) Traffic analysis encompasses analyzing volumes of DNS requests by IP address, domain, or hostname. Other traffic analysis techniques include geographic locations of
DNS servers, non-existent domain responses also known as NXDomain, and orphaned requests. (Farnham, 2013)

Splunk can perform all of the described detection techniques. Ryan Kovar and Steve Brant of Splunk have presented and written on how to use Splunk to detect DNS tunnels. In addition to their research, this approach provides the reader with a way to experiment with DNS tunnels and provides a more thorough understanding of how Splunk looks at the data in a lab environment.

6.1. Payload Analysis

6.1.1. Payload Analysis, Unauthorized DNS Servers

One of easiest ways to detect DNS tunneling is to determine which systems are valid DNS servers and block any other DNS service. The organization’s security policy should dictate what DNS servers are accessible to the hosts on the local network. Forcing all clients to use a restricted set of DNS servers helps narrow where DNS traffic is inspected and analyzed. The lab’s configuration emulates a production network. The perimeter firewall is pfSense, which is segmenting all the internal networks. Figure 11 shows the firewall rule set for Lan 4, which is Subnet 172.31.40.0/24, and VMware port group Firewall Leg 3. This subnet contains the DNS Tunnel Client 172.31.40.100, or also known as the victim machine. The first rule is configured to block access to the DNS tunnel servers on LAN3, subnet 172.31.30.0/24, and VM port group Firewall Leg 2. Rule 2 only permits DNS traffic to the Bind Server 172.31.20.60 on LAN 2, subnet 172.31.20.60/24, and VM port group Firewall Leg 1. Rule 3 blocks all other DNS traffic. Rule 4 is a permit all rule to allow the client access to the Internet. Notice the blue icon with an (i) to the left of the each rule. The blue icon indicates the rule has logging enabled. The firewall’s configuration sends syslog to a Splunk listener on UDP port 516.

Steve Jaworski, jaworski.steve@gmail.com
To emulate DNS requests in the lab, a tool named DNS Grind 1.0 from pentestmonkey.com and a list of the top one million domain names from Alexa will generate the required traffic to trigger the firewall rules. Alexa, an Amazon Company, tracks the top one million domain names and publishes the list as a free download. The DNS Grind tool is a Perl script that enumerates host names for a given domain. It is also a great tool to perform specific record type queries fulfilling the need to generate DNS traffic in the lab. Before using the top one million domain name list, it has to be downloaded, extracted, ranking numbers removed and then split into lists. Refer to the appendix on how to reduce the domain name list.

First, a firewall needs to trigger a block for unauthorized DNS servers. See Figure 12. The dns-grind.pl script calls the chosen domain list using switch -f. Then a specific DNS server is used with switch -n, and switch -m is set to limit the number or process to five. Next, switch -v is called for verbose output, and finally the script is instructed to find name server records for the given domain list.

Second, the firewall DNS permit rule needs to be triggered. The same DNS Grind script executes with two differences. See Figure 13. A different domain list is used to...
provide a variety of DNS names in the logs and the client 172.31.40.100 is configured to query the Bind 9 DNS server 172.31.20.60.

Figure 13

| Trigger a firewall permit. |
|:--------------------------:
| `/dns-grind.pl -f ..l1mldomains/xab -m 5 -v NS` |

Output

<table>
<thead>
<tr>
<th>Resolver Processes</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Records file</td>
<td>...l1mldomains/xab</td>
</tr>
<tr>
<td>Query timeout</td>
<td>5 secs</td>
</tr>
<tr>
<td>Recursive queries</td>
<td>On</td>
</tr>
<tr>
<td>DNS Server</td>
<td>Determine by OS</td>
</tr>
</tbody>
</table>

Next, the security analyst uses Splunk to find the unauthorized DNS servers. The query in Figure 14 narrows the result to pfSense logs only by defining the sourcetype and only destination port 53. Since Splunk is monitoring all segments of the lab, the Bind 9 server is also excluded as a source from the search. Otherwise, duplicate firewall log events will appear. The analyst only wants to find clients making DNS requests, not the DNS servers making requests. Use the stats command to count by source IP address, destination IP address and transport.

Figure 14

```
sourcetype=pfsense* dest_port=53 src_ip!=172.31.20.60
| stats count by action src_ip dest_ip transport
```

Figure 15

<table>
<thead>
<tr>
<th>action</th>
<th>src_ip</th>
<th>dest_ip</th>
<th>transport</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>allowed</td>
<td>172.31.30.100</td>
<td>172.31.20.60</td>
<td>udp</td>
<td>2</td>
</tr>
<tr>
<td>allowed</td>
<td>172.31.40.100</td>
<td>172.31.20.60</td>
<td>udp</td>
<td>7</td>
</tr>
<tr>
<td>blocked</td>
<td>172.31.40.100</td>
<td>208.67.220.220</td>
<td>udp</td>
<td>10</td>
</tr>
<tr>
<td>blocked</td>
<td>172.31.40.100</td>
<td>208.67.222.222</td>
<td>udp</td>
<td>15</td>
</tr>
<tr>
<td>blocked</td>
<td>172.31.40.100</td>
<td>8.8.8.4</td>
<td>udp</td>
<td>30</td>
</tr>
<tr>
<td>blocked</td>
<td>172.31.40.100</td>
<td>8.8.8.8</td>
<td>udp</td>
<td>6</td>
</tr>
</tbody>
</table>

As expected, the firewall rule configuration for the DNS tunnel client prohibits host 172.31.40.100 from accessing all DNS servers except for Bind server 172.31.20.60. See Figure 15 above. The result of found unauthorized DNS servers could be an indication

Steve Jaworski, jaworski.steve@gmail.com
of an infected or misconfigured host. This search and report will instruct the analyst to review the configuration of the affected host 172.31.40.100.

6.1.2. Payload Analysis, Hostname Entropy

Entropy describes the randomness of a string. In the case of DNS names, Domain Generating Algorithms (DGAs) create random hostnames such as asdlfkjasdflwerjak.t1.security.local. The more randomness in the string creates a higher the entropy. The less randomness, such as www.sans.org, the lower the entropy score. There are different formulas for entropy. The most common entropy formula for this use case is related to computer science and was developed by Claude Shannon. (Kovar, 2015) Splunk does not calculate the entropy of a hostname by default. A free app named “URL Toolbox” is available at the Splunk App Store created by Cedric Le Roux. This URL Toolbox app supports two functions to help find DNS tunnels. First, URL Toolbox extracts the hostname from the FQDN, second it includes the entropy function to detect the randomness of the hostname.

Splunk can be used to identify the example DNS tunnel described in Section 3.2. The tunnel is passing through the lab Bind DNS server. See Figure 16. The search will focus on the logs generated by Bind. The goal is to find hostnames with a high entropy score.

Figure 16

The query in

Figure 17 is restricted to just the logs from Bind DNS. The EVAL command instructs Splunk to create the field value pair utlist=custom. Custom represents the top level domain (TLD) list that is used by the ut_shannon command. The custom TLD file was edited to add .local and .lan, which are the TLD’s chosen in the lab. Refer to the

Steve Jaworski, jaworski.steve@gmail.com
appendix on how to configure URL Toolbox. Then the command ut_shannon is called on the event field query. The event field, query, contains all FQDN’s indexed by Splunk. The next step in the search is to look for an entropy score less than 2.5. In the last portion of the search, the formatted output is a table with time, DNS name and entropy score.

**Figure 17**

```plaintext
sourcetype="isc:bind:query" | eval utlist = "custom" | `ut_shannon(query)` | search ut_shannon < 2.5 | table _time query ut_shannon | sort -ut_shannon
```

The search was restricted to an entropy score less than 2.5 to show domain names that may not be malicious or using DNS tunneling. Here are the results in Figure 18.

These are valid domain names from the Alexa top one million domain names list.

**Figure 18**

<table>
<thead>
<tr>
<th>time</th>
<th>query</th>
<th>ut_shannon</th>
</tr>
</thead>
<tbody>
<tr>
<td>2016-03-21 05:15:45.614</td>
<td>emoment.net</td>
<td>2.481714572986073</td>
</tr>
<tr>
<td>2016-03-21 05:16:01.705</td>
<td>use-us.ru</td>
<td>2.4193819456463714</td>
</tr>
<tr>
<td>2016-03-21 05:15:57.999</td>
<td>btmee.net</td>
<td>2.4193819456463714</td>
</tr>
<tr>
<td>2016-03-21 05:15:40.137</td>
<td>nawara.ru</td>
<td>2.4193819456463714</td>
</tr>
<tr>
<td>2016-03-21 05:15:39.165</td>
<td>eonon.com</td>
<td>2.4193819456463714</td>
</tr>
<tr>
<td>2016-03-21 05:15:38.971</td>
<td>msa.ac.za</td>
<td>2.4193819456463714</td>
</tr>
<tr>
<td>2016-03-21 05:15:07.578</td>
<td>eonon.com</td>
<td>2.4193819456463714</td>
</tr>
<tr>
<td>2016-03-21 05:15:06.721</td>
<td>msa.ac.za</td>
<td>2.4193819456463714</td>
</tr>
</tbody>
</table>

Take the same query and reverse the entropy score to greater than > 2.5 in Figure 19.

**Figure 19**

```plaintext
sourcetype="isc:bind:query" | eval utlist = "custom" | `ut_shannon(query)` | search ut_shannon > 2.5 | table _time query ut_shannon | sort -ut_shannon
```

The results immediately show a high entropy score and very random domain names. Due to the randomness of the hostname, this could indicate a DNS tunnel is in use. Detecting entropy of DNS names indicates the use of domain generating algorithms. DGA’s are not only an indication of DNS tunnels but also malware and web exploits. (Kovar, 2015) See

**Figure 20.**

Steve Jaworski, jaworski.steve@gmail.com
Using Splunk to Detect DNS Tunneling

Steve Jaworski, jaworski.steve@gmail.com

Figure 20

<table>
<thead>
<tr>
<th>time</th>
<th>query</th>
<th>ut_shannon</th>
</tr>
</thead>
<tbody>
<tr>
<td>2016-03-26 10:47:14.090</td>
<td>z4g2aAbBcCdDeEfGhHiJkJkKlMnNoOpPqQrRsSTtVuVvWwXYyZyZ.1.security.local</td>
<td>5.6821621492957926</td>
</tr>
<tr>
<td>2016-03-26 12:37:08.740</td>
<td>0eeai82\190w\238sJ\249aabacuqe1c\206eabgbi\250bG\206Oekfc\190E\239Cawr7lgg7\225Hgcam\196JW\aat\192\219c\203\206W\233i\189\191\192\243\208\252\190h\244\190bD\200\233nKG.N\171\2034\206pfL\244v\193Ew\243\233nq\234Ki\232oW\239d\226A\239OUHE\248HG\233F\217a\230sLo\250\1966qYPE\235\196.11.security.local</td>
<td>4.655122932524963</td>
</tr>
<tr>
<td>2016-03-26 12:36:59.378</td>
<td>0qhak82\190w\238sJ\249aabacuqe1b\2500abgDJG\206Oekfc\190E\239CawrTb\2127\225G\190Gcam\196i\212a.aat\192\219c\203\206W\233i\189\191\192\243\208\252\190h\244\190bD\200\233nKG.N\171\2034K\206pfL\244v\193Ew\243\233nq\234Ki\232oW\239d\226A\239OUHE\248HG\233F\217a\230sLo\250\1966qYPE\235\196.11.security.local</td>
<td>4.645879120898666</td>
</tr>
</tbody>
</table>

6.1.3. Payload Analysis, Statistical Analysis

A company offering services on the Internet wants to make it easy for consumers to access their services. It makes sense that companies will use a DNS name that is easy to remember and as short as possible. Analyzing the length of the FQDN can help determine which domains are malicious.

There could be millions if not billions of DNS requests to analyze on an organization’s network. Simply relying on entropy and query counts only provides one
point of view. Using the statistics engine in Splunk, DNS tunnels are detected by calculating the standard deviation. (Brant & Kovar, 2015) The standard deviation formula can show what DNS requests are not normal on the organization’s network. Executing the search in Figure 21 will compute the length, average and standard deviation for each DNS query. It will then display only the results where the length of the DNS request is greater than three times the standard deviation. An outlier is the request length being more than three times the average length of all the DNS requests. The value three times the standard deviation is not always correct. Depending on the organization and data set the value can be anywhere from two to five times the standard deviation. (Kovar, 2015) The analyst has to determine which value fits their data set.

**Figure 21**

```
tag=dns | eval qlen=len(query)  | eventstats avg(qlen) as avg stdev(qlen) as stdev |
| where qlen>(stdev*3) | stats count by qlen stdev avg sourcetype query
```

Notice the difference with the search in Figure 21 above. It starts with tag=dns and will pull DNS events from multiple sourcetypes in Splunk. The lab is configured to send multiple data sources to Splunk. They are pfSense logs, Bind logs, and Splunk Stream DNS logs. In previous queries, the sourcetype was part of the search. The sourcetype helps Splunk know the difference between data types. The tag is a way to search similar events without specifying each sourcetype. Users, apps and Splunk defaults can create Splunk search tags. The tag=dns ties to sourcetypes isc:bind:query from Bind logs and stream:dns from the Stream App. With this search, the tunnel was detected in both the Bind logs and off the wire by Splunk Stream. Another item to note is that the Bind logs show backslashes (\) in the query and the events from the Stream App do not. See Figure 22. It appears the Splunk Stream App cannot decode DNS names if there are backslashes in the request. The tunnel application Iodine uses backslashes in the query while Dnscat2 uses alphanumeric characters. It is important to collect DNS event information from as many sources as possible. Each source will provide the analyst a different point of view. It is an important reminder that there is not a single solution to stop or detect all DNS tunneling.

Steve Jaworski, jaworski.steve@gmail.com
The most common record types for DNS are A, PTR, MX, CNAME, TXT, NS, and SOA records. Infrequent record types are AAAA, AXFR, and DNSKEY. The Sender Policy Framework (SPF) relies on TXT record types to reduce email spam. TXT record requests should only be coming from network hosts that require this type of lookup. If TXT records are increasing and not coming from a valid source such as a mail gateway, this is a red flag. The analyst should investigate the event further. Using Splunk, the security analyst performs simple counts of DNS record types over a particular time range. The Splunk Stream app collected the DNS events off the wire between the client and its upstream DNS server. Notice there is a new element in the search following the sourcetype event field called (source="stream:Test_DNS_Tunnel_Detection"). See Figure 23. By default, the Splunk Stream app collects specific DNS events and summarizes the data. Configuring the additional source collects and decodes all DNS traffic.

Steve Jaworski, jaworski.steve@gmail.com
The results show a high amount of NULL, SRV, and TXT queries. See Figure 24. The significant amount of rare record types should prompt the analyst to investigate the events further.

<table>
<thead>
<tr>
<th>query_type</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULL</td>
<td>4479</td>
</tr>
<tr>
<td>SRV</td>
<td>1105</td>
</tr>
<tr>
<td>TXT</td>
<td>437</td>
</tr>
<tr>
<td>AAAA</td>
<td>128</td>
</tr>
<tr>
<td>A</td>
<td>45</td>
</tr>
<tr>
<td>NS</td>
<td>13</td>
</tr>
<tr>
<td>PTR</td>
<td>11</td>
</tr>
<tr>
<td>*</td>
<td>6</td>
</tr>
<tr>
<td>DS</td>
<td>6</td>
</tr>
<tr>
<td>DNSKEY</td>
<td>5</td>
</tr>
<tr>
<td>65399</td>
<td>2</td>
</tr>
<tr>
<td>CNAME</td>
<td>2</td>
</tr>
<tr>
<td>MX</td>
<td>2</td>
</tr>
</tbody>
</table>

The power of Splunk allows the analyst to drilldown into the Null record type events revealing the requested queries. The security analyst can see the possibility of a DNS tunnel because of the random hostname queries. See Figure 25.
The security analyst continues to drill down into an individual event to review more details. In Figure 26 the analyst learns the client making the request is 172.31.40.100 and it is relaying through the organization’s DNS server of 172.31.20.60 to the final DNS server destination 172.31.30.100. Because the Splunk Stream app captures the entire DNS query from the network, it is able to provide additional information about the request to the analyst.

6.2. Traffic Analysis

6.2.1. Traffic Analysis, Volume of DNS Requests

Similar to finding rare record types such as Null and SRV, just performing a simple count of top domains can detect a tunnel. Parsing out the hostname and subdomains from
the FQDN allows the analyst to perform additional metrics. The newly parsed fields could be the hostname, subdomain name, the domain name, or the top level domain. Since a tunnel will create a tremendous amount of DNS requests when transferring a file, one can assume a simple count may result in outliers.

In this search, the goal is to build a time chart for DNS requests over the last hour. See Figure 27. This search calls the URL parsing function of URL Toolbox. It will break apart the domain name into a TLD, domain, hostname, and subdomains if any exist. In this case, the search is only for the domain and TLD. In Figure 28, there are three spikes for domain security.local, which is a good indicator there is a large amount of traffic for this domain.

Figure 27

```plaintext
sourcetype="isc:bind:query" | eval list="custom" | `ut_parse(query,list)` | timechart
span=1m useother=f usenull=f count(ut_domain) by ut_domain
```

Figure 28

The next step is to learn a little more about this domain. Are there any subdomains in use? Using a similar query as before, the search below concatenates the extracted subdomain event field generated by the ut_parse command to the ut_domain creating a new event field subanddomain. See Figure 29.

Figure 29

```plaintext
sourcetype="isc:bind:query" | eval list="custom" | `ut_parse(query,list)` | eval
subanddomain=ut_subdomain_level_1+"."+ut_domain | timechart span=1m
useother=f usenull=f count(subanddomain) by subanddomain
```

Steve Jaworski, jaworski.steve@gmail.com
The blue peaks in Figure 30 are very close to the red peaks in Figure 28 above. The only thing different is the blue peaks are for t1.security.local. Now there is a somewhat flat but elevated green line for t2.security.local. There are two tunnels in the domain security.local. Tunnel t1.security.local is transferring more data, while t2.security.local is not.

![Figure 30](image)

In a real world scenario, the attacker may not use a subdomain because they need as much length as possible in the DNS name to increase the speed of exfiltrating large amounts of data. (Ekman & Andersson, 2014) The point of the subdomains is to display the difference in tunneling tools in the lab. For the most part, tunneling will be somelonghostname.domain.tld, not somelonghostname.subdomain.domain.tld.

**6.2.2. Traffic Analysis, Geographic Location**

Another way to detect unwarranted DNS requests is to see where in the world the requests are forwarding. The analyst should investigate DNS requests resolving to DNS servers outside the organization’s geographic area. If the organization does not conduct business in X country, does it make sense for DNS requests to be going there? Splunk has built in IP location services to provide the analyst an idea of where the remote DNS server may be located. Refer to the search from Section 6.1.1. Payload Analysis, Unauthorized DNS Servers. Appending an additional command to the search named iplocation. Excluding the internal DNS servers from the search removes duplicate destination IP addresses. See Figure 31.

Steve Jaworski, jaworski.steve@gmail.com
Figure 31

Original Query from 6.1.1
sourcetype=pfsense* dest_port=53 src_ip!=172.31.20.60
| stats count by action src_ip dest_ip transport

New Query with IP location
sourcetype=pfsense* dest_port=53 dest_ip!=172.31.0.0/16
| stats count by action src_ip dest_ip transport
| iplocation dest_ip

The results show what city, country, region, latitude and longitude of where the destination DNS server may be located. Geolocation services are not 100% accurate. The geolocation provider primarily relies on the accuracy of the Regional Internet Registries databases. The service only provides a general geographical location. See Figure 32.

Figure 32

<table>
<thead>
<tr>
<th>dest_ip</th>
<th>transport</th>
<th>count</th>
<th>City</th>
<th>Country</th>
<th>Region</th>
<th>lat</th>
<th>lon</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.21.11.162</td>
<td>udp</td>
<td>1</td>
<td>Tokyo</td>
<td>Japan</td>
<td>Tokyo</td>
<td>35.6850</td>
<td>139.75140</td>
</tr>
<tr>
<td>1.21.170</td>
<td>udp</td>
<td>1</td>
<td>Tokyo</td>
<td>Japan</td>
<td>Tokyo</td>
<td>35.6850</td>
<td>139.75140</td>
</tr>
<tr>
<td>1.8.241.1</td>
<td>tcp</td>
<td>2</td>
<td>Beijing</td>
<td>China</td>
<td>Beijing Sh</td>
<td>39.92890</td>
<td>116.38830</td>
</tr>
<tr>
<td>1.8.241.1</td>
<td>udp</td>
<td>1</td>
<td>Beijing</td>
<td>China</td>
<td>Beijing Sh</td>
<td>39.92890</td>
<td>116.38830</td>
</tr>
<tr>
<td>1.8.241.1</td>
<td>tcp</td>
<td>1</td>
<td>Beijing</td>
<td>China</td>
<td>Beijing Sh</td>
<td>39.92890</td>
<td>116.38830</td>
</tr>
<tr>
<td>1.8.242.1</td>
<td>udp</td>
<td>3</td>
<td>Beijing</td>
<td>China</td>
<td>Beijing Sh</td>
<td>39.92890</td>
<td>116.38830</td>
</tr>
<tr>
<td>1.8.243.1</td>
<td>tcp</td>
<td>1</td>
<td>Beijing</td>
<td>China</td>
<td>Beijing Sh</td>
<td>39.92890</td>
<td>116.38830</td>
</tr>
<tr>
<td>1.8.243.1</td>
<td>udp</td>
<td>1</td>
<td>Beijing</td>
<td>China</td>
<td>Beijing Sh</td>
<td>39.92890</td>
<td>116.38830</td>
</tr>
<tr>
<td>100.42.61.118</td>
<td>udp</td>
<td>1</td>
<td>Santa Rosa</td>
<td>United States</td>
<td>California</td>
<td>38.43800</td>
<td>-122.67530</td>
</tr>
<tr>
<td>100.42.62.228</td>
<td>udp</td>
<td>2</td>
<td>Santa Rosa</td>
<td>United States</td>
<td>California</td>
<td>38.43800</td>
<td>-122.67530</td>
</tr>
</tbody>
</table>

Taking the search a step further, some analysts or managers may prefer a geographical map. Append an additional command to the search called geostats. See Figure 33.

Figure 33

sourcetype=pfsense* dest_port=53 dest_ip!=172.31.0.0/16 | stats count by action src_ip dest_ip transport | iplocation dest_ip | geostats latfield=lat longfield=lon count by dest_ip

Steve Jaworski, jaworski.steve@gmail.com
The analyst can use the drilldown map to determine where the DNS servers might be geographically located. See Figure 34.

**Figure 34**

![Drilldown Map](image)

### 7. Conclusion

Neglecting to monitor DNS traffic is high risk to any organization. Due to the nature of how DNS functions, it is not conceivable to block every possible DNS tunnel scenario. DNS tunnels bypass security controls to exfiltrate data, gain free Internet access or execute malware functions.

Iodine and DNSCAT2 are not the only DNS tunnel tools available. Other tunnel applications are OzymanDNS, Dns2tcp, Heyoka, DNSCat, NSTX, DNSscapy, MagicTunnel, and VPN over DNS. (Mazerik, 2014) It is important for the analyst to learn what type of events these other DNS tools generate. Adding the results of the other tunnel applications to Splunk will increase the organization’s detection capability.

Learning to use Splunk and taking the time to build this lab will provide the analyst a better understanding of how tunnels function. The analyst will also improve their detection techniques and start to think like a malicious actor. By modifying the configuration, the analyst can search in Splunk to determine how the events are different.

Steve Jaworski, jaworski.steve@gmail.com
Tunnels can be detected using a variety of payload and traffic analysis techniques. Analyzing the length and entropy of the DNS requests helps the analyst determine what DNS traffic is valid. Just by looking at what and how many record types are being used, provide valuable insight into what users are doing on the network. Determining where DNS requests are possibly being sent geographically help limit exposure to malicious domains.

With the appropriate DNS configuration and monitoring, DNS tunnels can at least be detected before extensive damage affects the organization.
8. References


Steve Jaworski, jaworski.steve@gmail.com


Appendix 1

9. Lab Introduction

This appendix provides the reader with instructions to rebuild the laboratory used to develop this paper. The idea was to provide enough information for all skill levels to follow and learn.

Based on the installer’s technical skill, they may or may not need to review all the steps. The only recommendation is to install and configure the system in the same order outlined below. The lab build can take anywhere from 2 to 8 hours to build.

There are hyperlinks in some sections the author used help configure the lab. Notes from the author are also included. If the installation is not working, refer to the links. Google is a great friend for troubleshooting.

Warning: The instructions do not include how to secure any of the systems. Defaults certificates are used and hardening of systems is not included in the steps. Expect this system to be insecure, because it is.

You can email the author with questions. He will answer when time permits.

10. How to Build the Lab

The lab can be built using a single PC. The PC specs are listed below that was used to build the lab. With virtualization, the more memory the better. The CPU has to be 64 Bit and have at least two cores. The lab specs can be exceeded.

In addition to having a PC dedicated to running the virtualization hypervisor, an management workstation is required to download tools and software, access the SPLUNK UI, SSH to virtual machines, and manage the hypervisor.

PC Virtual Server Specs
- 1 CPU Intel i5-3470 3.20 GHZ
- 8 Gigabytes of memory
- Hard disk
  - 140 GB Drive for the VMware Hypervisor and ISO image storage
  - 2 TB for guest images
  - 2 TB for Splunk guest image
- Single Gigabyte Network Adapter 82574L
Workstation Specs
- Preferably run Windows 7 x64
- Microsoft .Net (vSphere client will download required version if not already installed)
- 5 GB of free disk space
- 4 GB of RAM
- 1 GHz or faster processor

Operating Systems
- VMware ESXi 6.x, Free Hypervisor
- Debian 8.x, Net Install amd64
- Ubuntu 14.04.4 LTS Network Install x86_64

Applications
- Various Linux Tools and Applications
  - tcpdump
  - sudo
  - scp
  - ping
  - nslookup
  - dig
  - vi
  - md5sum
  - ifconfig
  - sed
  - split
  - wget
  - git
  - make
- Splunk 6.3.x, Free Version
- Splunk Apps
  - Splunk Stream 6.4.2
  - Splunk CIM 4.3.1
  - Technology Add-on for pfSense 2.0.6
  - Splunk Add-on for ISC Bind 1.0.0
  - URL Toolbox 1.5
- pfSense 2.2.6-RELEASE (amd64)
- Bind9
- SSH Client
  - Putty
  - Any client that support SSHv2
- WinSCP
- DNS Tunneling Tools
  - Iodine
  - DNSCAT2
Ruby Development Tools

- DNS Grind 1.0
- VMware
  - vSphere Client 6.0
  - VMware vCenter Converter Standalone Client 6.1 or Higher (Optional)
  - VMware Workstation Pro 12 (Optional)

Note: A user account is required to download the VMware Hypervisor from vmware.com. Registration is free. Splunk also requires registration and it is free.

11. Plan the Network Layout

The lab requires four subnets and two virtual switches.

This is what the topology will look like when completed.

It is assumed there is a DHCP server on the local network outside the hypervisor to supply IP addresses to the management workstation and other systems on that subnet.

- First Virtual Switch
  - vSwitch0 (default switch created when installing VMware)
  - No VLAN configured
  - No port groups are configured
  - The outside/external/WAN pfSense firewall interface is connected to this virtual network.
Using Splunk to Detect DNS Tunneling

- The management interface of the Splunk Server is connected to this virtual network
- The physical interface of the PC is connected to this virtual switch
- The subnet for this virtual switch is 172.31.10.0/24

- Second Virtual Switch
  - 3 port groups will be created, each port group will be connected to a pfSense interface
  - The pfSense firewall is the bridge between the first and second virtual switch
  - Port Group 1
    - Assigned VLAN 100
    - Network Label = Firewall Leg 1
    - Promiscuous Mode is enabled
    - Connected to pfSense interface LAN 2
    - The subnet for this port group is 172.31.20.0/24
  - Port Group 2
    - Assigned VLAN 200
    - Network Label = Firewall Leg 2
    - Promiscuous Mode is enabled
    - Connected to pfSense interface LAN 3
    - The subnet for this port group is 172.31.30.0/24
  - Port Group 3
    - Assigned VLAN 300
    - Network Label = Firewall Leg 3
    - Promiscuous Mode is NOT enabled
    - Connected to pfSense Interface LAN 4
    - The subnet for this port group is 172.31.40.0/24

Note: The IP subnets can be any private subnet documented in RFC 1918. The network labels can be named anything. Be sure to document the chosen network labels and IP subnets. Adjust any required settings that fit best. Use the mapping below to help keep track of the network configuration. The second virtual switch will have the next number available. The writer of this lab has other virtual switches for other projects. The lab shows the second virtual switch as number 4. The virtual switch number is arbitrary.
When everything is configured, the secondary virtual switch will look similar to the figure below:

<table>
<thead>
<tr>
<th>vSwitch</th>
<th>Port Group</th>
<th>VLAN</th>
<th>Promiscuous Mode</th>
<th>Virtual Server</th>
<th>Subnet/IP</th>
<th>pfSense Interface</th>
<th>pfSense IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>None</td>
<td>None</td>
<td>Disable</td>
<td>Splunk Server Mgmt</td>
<td>172.31.10.0/24</td>
<td>WAN/External</td>
</tr>
<tr>
<td>#</td>
<td>Firewall Leg 1</td>
<td>100</td>
<td>Enabled</td>
<td>Bind 9 Server</td>
<td>172.31.20.0/24</td>
<td>LAN 2</td>
<td>172.31.20.1</td>
</tr>
<tr>
<td>#</td>
<td>Firewall Leg 2</td>
<td>200</td>
<td>Enabled</td>
<td>Iodine Tunnel Srv</td>
<td>172.31.30.0/24</td>
<td>LAN 3</td>
<td>172.31.30.1</td>
</tr>
<tr>
<td>#</td>
<td>Firewall Leg 3</td>
<td>300</td>
<td>Disable</td>
<td>DNS Tunnel Client</td>
<td>172.31.40.0/24</td>
<td>LAN 4</td>
<td>172.31.30.1</td>
</tr>
</tbody>
</table>

When everything is configured, the secondary virtual switch will look similar to the figure below:

12. Start the Downloads

VMware ESXi 6.x Free Hypervisor

Steve Jaworski, jaworski.steve@gmail.com
https://my.vmware.com/en/web/vmware/info/slug/datacenter_cloud_infrastructure/vmware_vsphere_hypervisor_esxi/6_0

Note: The hypervisor has a trial period for all the options. Be sure to register for a free license key before the trial version expires.

Splunk 6.3.X

https://www.splunk.com/page/previous_releases#x86_64linux

Note: Download the deb package 64Bit. The lab used 6.3.3, but 6.3.4 will most likely work without issue. As of May 2016 6.4 is available, with a little tweaking if required, the lab should work with the newest version.

Download Debian 8.X Network Install

http://cdimage.debian.org/debian-cd/8.4.0/amd64/iso-cd/debian-8.4.0-amd64-netinst.iso

Note: While the lab used Debian 8.3 any version of 8.X will work. Just download the latest version of 8.X. The full installation version of Debian 8.X can be downloaded also, but the network install is significantly smaller and will be faster to download.

Steve Jaworski, jaworski.steve@gmail.com
Using Splunk to Detect DNS Tunneling

Steve Jaworski, jaworski.steve@gmail.com

Download Ubuntu 14.04 LTS Network Install


Note: Choose the amd64 image

Download pfSense

https://www.pfsense.org/download/

Note: Download the Latest Stable Version (Community Edition). Choose AMD64 Architecture and CD ISO Installer

Latest Stable Version (Community Edition)

This is the most recent stable release, and the recommended version for all installations. For upgrade information, see the Upgrade Guide.
13. Install the Hypervisor

Burn the VMware ISO to a CD/DVD or USB Key.

Install the hypervisor like any other operating system on the PC. Accept all the defaults for this installation.

Note: Depending on the network being setup, it is the choice of the installer to use a static or DHCP assigned IP address for the management interfaces of the hypervisor. Either option will work.

Document the Management IP and Root password created.

Install vSphere Client and Access the Hypervisor

On the management workstation, using InternetExplorer, Firefox, of Chrome access the management IP address of the VMware server.

1. Access [https://172.31.10.166](https://172.31.10.166)

2. Accept the Default Invalid Certificate, Click Proceed
3. Select “Download vSphere Client for Windows”
   a. This is a download from the internet, depending on the connection speed it may take some time

4. Save the File to a preferred location
5. Install the application, accept defaults
6. Open the vSphere Client
   a. Click the Start Button
   b. The application is located in All Programs/VMware

7. Enter the IP address of the VMware server IP Address, Username and Password
8. Click Login
9. Ignore the Certificate Warning

10. The default page should be the Summary tab. A successful connection has been made to the hypervisor
14. Create Virtual Switches, Port Groups, Promiscuous Mode

1. Select the Configuration Tab

2. Select Networking from the Hardware column

3. Select Add Networking (Look to the far right)
4. Choose Virtual Machine, Click Next

5. Choose “Create a vSphere standard switch” Uncheck any network adaptors, Click Next

6. Create Port Group; Add Network Label And VLAN ID, Click Next, Click Finish
   a. Network Label = Firewall Leg 1
   b. VLAN = 100

7. Scroll down in the vSphere Standard Switch window until the virtual switch that was just created appears and select Properties

8. Click Add to add remaining port groups, Step 4 and 6 will be repeated.
   a. Network Label = Firewall Leg 2 and VLAN = 200
   b. Network Label = Firewall Leg 3 and VLAN = 300
   c. When completed, the configuration will look like this

9. Configure Promiscuous Mode
   a. Select Firewall Leg 1, Click Edit, Select Security Tab
   b. Check the box to the right of Promiscuous Mode, In the drop down choose Accept

Steve Jaworski, jaworski.steve@gmail.com
c. Click OK

d. Repeat the same steps for “Firewall Leg 2”

For more information on promiscuous mode in VMware check out these links.

- [https://fojta.wordpress.com/2014/10/02/promiscuous-port-myth/](https://fojta.wordpress.com/2014/10/02/promiscuous-port-myth/)

Upload ISO images to VMware Server

1. Log into the VMware server using the vSphere Client
2. From the Summary Tab, find Datastore1 in the storage column under Resources, Right-Click and choose Browse Datastore

3. Create ISO Folder in the root “/” directory
4. Create Subfolder “Debian” and add additional nested folder “8”
5. Create Subfolder “Ubuntu” and add additional nested folder “14”
6. Create Subfolder “pfSense”

Steve Jaworski, jaworski.steve@gmail.com
7. Select the Debian/8 Folder, Click the Upload Button, Upload the Debian ISO Image
8. Repeat same steps for Ubuntu and pfSense

9. Close Datastore Browser Window

15. Install pfSense

1. Log into the VMware server using the vSphere Client
2. Select File, New, New Virtual Machine
3. Choose Typical and Click Next

Steve Jaworski, jaworski.steve@gmail.com
4. Give the pfSense firewall Name and Click Next

5. Choose Storage Location and Click Next
   a. The drive choose for this lab is dedicated to running most virtual machines

6. Select Guest Operation System, Choose Linux and Version “Other 3.x or later Linux (64-bit), Click Next

7. Create Networks
   a. Choose 4 NICs
   b. The NIC configuration should look like this
   c. Click Next

Steve Jaworski, jaworski.steve@gmail.com
8. Create Disk, Leave Defaults, Click Next

9. Review configuration, if every looks okay, Click Finish

10. The task window at the bottom will show when the virtual machine is completed.

11. Add ISO to pfSense virtual machine
   a. Expand the server tree if required
   b. Find pfSense virtual machine, Right Click and Choose Edit Settings
   c. Select CD/DVD drive 1
d. Select radio button next to “Datastore ISO File”

![Datastore ISO File selection](image1)

e. Choose Browse, find pfSense ISO image, Select, Click Open

![Browse Datastores](image2)

f. Review, Check the box next to “Connect at power on” if required, Then click OK

![pfSense - Virtual Machine Properties](image3)

12. Adjust Memory
   a. Edit Virtual Machine
   b. Select Memory
   c. Verify Memory Size is at 512 MB

![Memory Configuration](image4)

d. Click OK

13. Adjust Virtual Disk Controller

Steve Jaworski, jaworski.steve@gmail.com
a. Edit Virtual Machine  
b. Select SCSI Controller 0  
c. Select Change Type  
d. Choose “LSI Logic Parallel”  
e. Click Okay

14. Do not power on virtual machine at this time, leave powered off

16. Configure and start pfSense

1. Right Click on the Virtual Machine pfSense  
2. Choose Open Console  
3. Let Virtual Machine Boot  
4. When prompted, Click Inside the Guest Window and Press “I” to install
5. Accept Console Settings and Press Enter

6. Perform Quick/Easy Install

7. Tell the install okay
8. The install will continue

9. Install the standard Kernel

10. Reboot

11. Let the firewall boot, to the configuration prompt
    a. **IMPORTANT NOTE:** Take a snapshot of the interface mappings, they may be needed later for troubleshooting

Steve Jaworski, jaworski.steve@gmail.com
12. Choose N for VLAN configuration

13. Make vmx0 the WAN interface

14. For the LAN interface, just hit enter

15. Confirm and Proceed, Yes

16. View which IP address was assigned to the vmx0 interface, in this case it received the incorrect IP address. The interfaces need to be remapped. Take the snapshot from step 11 a. Compare the mac address to the virtual machine configuration
   a. Edit the virtual machine settings.
   b. Select the first network adapter, look for the MAC address to the right
   c. Now look at the snapshot table
d. Map the MAC addresses from the virtual machine network adapter and port group to the correct pfSense NIC.
e. In this case, the WAN interface vmx0 will map to Network Adapter 4 and change the port group to VM Network.

f. Change the configuration as required.

17. Enter “5” to reboot the firewall.

18. Following the Reboot, the WAN Interface, vmx0 now has the required subnet.

19. Login into pfsense from the workstation
   a. https://172.31.10.127
   b. Accept the invalid cert
   c. Login in with default credentials admin/pfsense

20. Follow the Wizard, Click Next

21. Click Next,

22. Fill out Hostname, Domain, DNS, Leave DNS Override Checked, and Click Next

Steve Jaworski, jaworski.steve@gmail.com
23. Set Time to lab admins preference, and Click Next

24. Leave WAN at Defaults, unless the lab admin requires a specific setting, Click Next

25. Create New Admin Password and Click Next

26. Reload Firewall

27. Continue to webConfigurator
28. Configure Anti-Lockout Rule on WAN Interface
   a. Duplicate this rule configuration
   b. Select Firewall then Select Rules
   c. Add the rules from step 28 A, before the Anti-Lockout Rule.
   d. Click the + icon
   e. Fill out the form
      i. Adjust source IP or subnet as required
f. Click Save

g. Add ICMP Rule

h. Before Applying Changes, the new rule set should look like this.

i. Select Apply Changes

j. Click Close
IMPORTANT NOTE: Before configuring any internal LAN interfaces make sure the WAN Anti-Lockout rule is in place, otherwise connectivity will be lost to the firewall from the management workstation. The firewall may have to reset to factory defaults forcing the installer to start the firewall configuration process over again.

29. Configure Interfaces
   a. Select Interfaces then Select Assign
      ![Select Assign](image)
   b. Add vmx1, Click + icon
      ![Add vmx1](image)
   c. Repeat steps for vmx2 and vmx3
   d. When complete, Interface Assignments will look like this
      ![Interface Assignments](image)
   e. Select Interface LAN
      i. Check the box next to Enable Interface
      ![Enable Interface](image)
      ii. Change Description to “LAN2”
      iii. Set IPv4 Configuration Type to “Static IPv4”
      ![IPv4 Configuration](image)
      iv. Set IP address to “172.31.20.1 /24”
      ![IPv4 Configuration](image)
   v. Click Save
   vi. Click Apply Changes

Steve Jaworski, jaworski.steve@gmail.com
f. Repeat the Same Steps for OPT1 and OPT2
   i. OPT1 Specs
      1. Rename to LAN 3
      2. Set IP to 172.31.30.1 /24
   ii. OPT2 Specs
       1. Rename to LAN 4
       2. Set IP to 172.31.40.1 /24

30. Create DHCP Scopes
   a. Select Services, then Select DHCP Server
   b. Configure DNS Scope and DNS Server for each LAN
   c. Start with LAN 2
      i. Create Range 172.31.20.100 to 172.31.20.150
      ii. Set DNS server to the Bind 9 Server 172.31.20.60
      iii. Click Save
   d. Repeat same steps for LAN 3 and 4
      i. LAN 3
         1. 172.31.30.100 to 172.31.30.150
         2. DNS = 172.31.20.60
      ii. LAN 4
         1. 172.31.40.100 to 172.31.40.150
         2. DNS = 172.31.20.60

31. Enable Syslog to Splunk
   a. Select Status then Select System Logs
b. Click the Settings Tab

c. Scroll to “Remote Logging Options”
   i. Source Address = WAN
   ii. IP Protocol = IPv4
   iii. Enable Remote Logging = Check the Box
   iv. Remote Syslog Server = IP Address of Splunk Server
   v. **IMPORTANT NOTE:** Notice the destination port is 516, NOT 514. This is not to conflict with other syslog services.
   vi. Remote Syslog Contents = Check the Box for Everything
   vii. Click Save
32. Finish Building Firewall Rule Set
33. Duplicate Rule Set for each interface in graphics below; do not forget to turn on logging for each rule.
17. Install Debian8 Virtual Machines

- Server Specs
- Bind 9
  - Server Name = Bind
  - domain = security.local
  - CPU = 1
  - IP Address = 172.31.20.60
  - NIC Card = Firewall Leg 1
  - HardDrive = 16 GB / Thin Provision
  - Datastore = 2
  - Memory = 2048 MB
  - Guest Operating System = Debian GNU/Linux 8 (64 Bit)

Steve Jaworski, jaworski.steve@gmail.com
The install steps here will be repeated four times, once for each server listed in the spec above. If the installer has experience with cloning, VMware Converter can be used to clone the first built virtual machine and make edits as required. The virtual machines can also be created with VMware Workstation 12 and then moved to the VMware server. How to use VMware converter and VMware workstation will not be covered here. VMware experience is helpful to speed up the installation of the virtual machines. If the installer has minimum VMware experience, the steps here will cover building each virtual machine individually; it will just take a little longer to setup the lab.

Note: Internet Access is required to install Debian and Unbuntu

Steve Jaworski, jaworski.steve@gmail.com
1. Create New Virtual Machine
2. Choose Typical, Click Next
3. Add Server Name from the Spec, Click Next
4. Choose the Datastore from the Spec, Click Next
5. Choose the Guest Operating System from the Spec, Click Next
6. Create Networking, from the Spec, Click Next
7. Create the Disk from the Spec, Click Next
8. Review the Specs
9. Before Clicking Finish, Check the box next to “Edit the virtual machine settings before completion”, Click Continue

10. Reviewing the Spec Change the configuration for memory and cpu

11. Add the ISO image from the Spec, Make sure “Connect at power on” is selected

12. Click Finish,
13. Watch the Recent Task Window to make sure there are no errors building the virtual machine

14. Right Click on Virtual Machine and Select Open Console

Steve Jaworski, jaworski.steve@gmail.com
15. Click the Green Power On Icon to start the virtual machine, watch the virtual machine boot.

16. Using the mouse, single left mouse click inside the virtual machine window
   a. When selected, the mouse will not be available to the workstation operation system, only the virtual machine
   b. To release the mouse from the virtual machine, press ctrl and alt together

17. Select Install and Press Enter

18. Select English and Press Enter

19. Select United States for Location, Press Enter

20. Select American English for Keyboard, Press Enter

21. Enter Hostname, Select Continue, Press Enter
22. Enter Domain Name, Select Continue, Press Enter

23. Enter Root Password, Select Continue, Press Enter

24. Verify Root Password, Select Continue, Press Enter

25. Document and store the root password somewhere secure
26. Enter Full Name, Select Continue, Press Enter
27. Enter User Name, Select Continue, Press Enter

Select a username for the new account. Your first name is a reasonable choice. The
username should start with a lower-case letter, which can be followed by any combination
of numbers and more lower-case letters.

Username for your account:
Steve

28. Enter password for user, Select Continue, Press Enter

A good password will contain a mixture of letters, numbers and punctuation and should be
changed at regular intervals.

Choose a password for the new user:

29. Re-enter password for User, Select Continue, Press Enter

Please enter the same user password again to verify you have typed it correctly.

Re-enter password to verify:

30. Select appropriate Time Zone, Press Enter

If the desired time zone is not listed, then please go back to
and select a country that uses the desired time zone (the country is located).

Select your time zone:

- Eastern
- Central
- Mountain
- Pacific
- Alaska
- Hawaii
- Arizona
- East Indiana
- Samoa

31. Select “Guided – use entire disk and set up LVM”, Press Enter

Steve Jaworski, jaworski.steve@gmail.com
32. Select VMware Virtual Disk, Press Enter

33. Select “Separate /home, /var, and /tmp partitions” and Press Enter
   a. IMPORTANT NOTE: When building the Splunk Server, choose “All files in one partition”, all other servers can use separate partitions.

34. Select “Yes” to write changes to disk and Press Enter

35. Select “Finish partitioning and write changes to disk” and Press Enter
36. Select “Yes” to write changes to disk and Press Enter

37. Base system will start to install

38. Choose a Mirror for package downloads, Select location closet to lab location, and Press Enter
39. Choose default ftp location and Press Enter

40. Leave HTTP proxy blank unless there is one, Select Continue, and Press Enter
41. Install will scan mirrors and start installing additional software

42. It’s up to the installer to popularity-contest, Choose Yes or No, and Press Enter

43. Choose Software Selection
   a. Only choose SSH Server and standard system utilities
   b. Deselect everything else
   c. Before selecting continue and pressing enter the selections should look exactly like this
   d. Select Continue and Press Enter

44. Packages will download and start to install
45. Select “Yes” and Press Enter to install Grub Boot Loader to MBR

![Select and install software](image)

Retrieving file 70 of 268

It seems that this new installation is the only operating system on this computer. If so, it should be safe to install the GRUB boot loader to the master boot record of your first hard drive.

Warning: If the installer failed to detect another operating system that is present on your computer, modifying the master boot record will make that operating system temporarily unbootable, though GRUB can be manually configured later to boot it.

Install the GRUB boot loader to the master boot record?

<Select option>

46. Select “/dev/sda” and Press Enter

![Select device](image)

You need to make the newly installed system bootable, by installing the GRUB boot loader on a bootable device. The usual way to do this is to install GRUB on the master boot record of your first hard drive. If you prefer, you can install GRUB elsewhere on the drive, or to another drive, or even to a floppy.

Device for boot loader installation:

Enter device manually

47. Select Continue and Press Enter to reboot

![Finish installation](image)

Installation complete, so it is time to boot into your new system. Make sure to remove the installation media (CD-ROM, floppies), so that you boot into the new system rather than restarting the installation.

48. The Linux Image will automatically boot.
Using Splunk to Detect DNS Tunneling

Steve Jaworski, jaworski.steve@gmail.com

49. Login to system with configured credentials.

50. Install sudo and add user to group
   a. Execute "su –l root"
   b. Execute "apt-get install sudo"
   c. Execute "sudo adduser yourusername sudo"
Using Splunk to Detect DNS Tunneling

Steve Jaworski, jaworski.steve@gmail.com

18. Install Ubuntu 14.04.4 LTS

- Server Specs
- DNS Server
  - Server Name = DNSTunnelServerDNSCAT2
  - IP Address = 172.31.30.101
  - NIC Card = Firewall Leg 2
  - HardDrive = 20 GB / Thick Provision Lazy Zeroed
  - Memory = 384 MB
  - Guest Operating System = Ubuntu Linux (64 Bit)
  - ISO = [datastore1] iso/Ubuntu/14/mini.iso

1. Follow the same steps from Install Debian8 to build the virtual machine up to installing the OS.
2. Open the Console
3. Start the Virtual Machine by clicking on the green start icon

4. Press Enter to Install

5. Choose Language and Press Enter

   - Czech  - Čeština
   - Danish  - Dansk
   - Dutch  - Nederlands
   - English  - English
   - Esperanto  - Esperanto
   - Estonian  - Eesti
   - Finnish  - Suomi

6. Choose Location and Press Enter

   - Philippines
   - Singapore
   - South Africa
   - United Kingdom
   - United States
   - Zambia
   - Zimbabwe
   - other

7. Do not detect keyboard layout and Press Enter

8. Set Keyboard Language and Press Enter

   - English (Cameroon)
   - English (Ghana)
   - English (Nigeria)
   - English (South Africa)
   - English (UK)
   - English (US)

9. Set Keyboard Layout and Press Enter
10. Enter Hostname and Press Enter

Please enter the hostname for this system.

The hostname is a single word that identifies your system to the network. If you don’t know what your hostname should be, consult your network administrator. If you are setting up your own home network, you can make something up here.

Hostname:

11. Choose Mirror Location and Press Enter

Choose a mirror of the Ubuntu archive

Please select an Ubuntu archive mirror. You should use a mirror in your country or region if you do not know which mirror has the best Internet connection to you.

Generally, your country code.archive.ubuntu.com is a good choice.

Ubuntu archive mirror:

12. Choose Mirror and Press Enter

Choose a mirror of the Ubuntu archive

If you need to use a HTTP proxy to access the outside world, enter the proxy information here. Otherwise, leave this blank.

The proxy information should be given in the standard form of http://[(username)][:password]@host[:port]/.

HTTP proxy information (blank for none):

13. Leave Proxy blank unless there is one, Select Continue, and Press Enter

14. Components will start loading

Steve Jaworski, jaworski.steve@gmail.com
15. Set Full name of User, Select Continue, and Press Enter

A user account will be created for you to use instead of the root account for non-administrative activities.

Please enter the real name of this user. This information will be used for instance as default origin for emails sent by this user as well as any program which displays or uses the user’s real name. Your full name is a reasonable choice.

Full name for the new user:

16. Set username, Select Continue, and Press Enter

Select a username for the new account. Your first name is a reasonable choice. The username should start with a lower-case letter, which can be followed by any combination of numbers and more lower-case letters.

Username for your account:

17. Create Password, Select Continue, and Press Enter

A good password will contain a mixture of letters, numbers and punctuation and should be changed at regular intervals.

Choose a password for the new user:

18. Verify Password

Please enter the same user password again to verify you have typed it correctly.

Re-enter password to verify:

19. Select No, for Encrypt Home Directory, Press Enter

You may configure your home directory for encryption, such that any files stored there remain private even if your computer is stolen.

The system will seamlessly mount your encrypted home directory each time you login and automatically unmount when you log out of all active sessions.

Encrypt your home directory?

20. Set time zone

Steve Jaworski, jaworski.steve@gmail.com
21. Partition disks, Select “Guided – use entire disk and set up LVM”, and Press Enter

22. Select Virtual Disk and Press Enter

23. Write changes to disk, Select Yes, and Press Enter

24. Leave Default Disk Space, Select Continue, and Press Enter
25. Write changes to disk, Select Yes, and press Enter

26. Base System will start to install

27. Select “Install security updates automatically” and Press Enter

28. Choose Software
   a. Select “Basic Ubuntu server”
   b. Select “OpenSSH server”
   c. Select Continue
d. Press Enter

![Software selection]

At the moment, only the core of the system is installed. To tune the system to your needs, you can choose to install one or more of the following predefined collections of software.

Choose software to install:

- Basic Ubuntu server
- OpenSSH Server
- MySQL server
- LAMP server
- Mail server
- PostgreSQL database
- Print server
- Samba file server
- Tomcat Java server
- Ubuntu Cloud Image (instance)
- Virtual Machine host
- DVD/CD creation and editing suite
- Audio recording and editing suite
- Kubuntu desktop
- Lubuntu Active
- Lubuntu desktop
- Lubuntu full
- Large selection of font packages
- Lubuntu Desktop
- Lubuntu minimal installation
- Mythbuntu additional roles
- Mythbuntu frontend

(Continue)

29. Packages will install

![Select and install software]

Retrieving file 77 of 164 (20s remaining)

30. Install GRUB on MBR, Select Yes and Press Enter

![Install the GRUB boot loader on a hard disk]

It seems that this new installation is the only operating system on this computer. If so, it should be safe to install the GRUB boot loader to the master boot record of your first hard drive.

Warning: If the installer failed to detect another operating system that is present on your computer, modifying the master boot record will make that operating system temporarily unbootable, though GRUB can be manually configured later to boot it.

Install the GRUB boot loader to the master boot record?

<Go Back> <Yes> <No>

31. Leave System Clock at UTC, Select Yes and Press Enter

![Virtual Machine host]

System clocks are generally set to Coordinated Universal Time (UTC). The operating system uses your time zone to convert system time into local time. This is recommended unless you also use another operating system that expects the clock to be set to local time.

Is the system clock set to UTC?

<Go Back> <Yes> <No>

32. Finish Install and Reboot, Select Continue, and Press Enter

![Finish the installation]

Installation complete, so it is time to boot into your new system. Make sure to remove the installation media (CD-ROM, floppies), so that you boot into the new system rather than restarting the installation.

<Go Back> <Continue>

Steve Jaworski, jaworski.steve@gmail.com
33. Login to Ubuntu Install

```
Ubuntu 14.04.4 LTS ubuntu tty1

ubuntu login: so0ap0x
Password:
Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 3.15.0-05-generic x86_64)

* Documentation: https://help.ubuntu.com/

System information as of Mon May 16 07:55:12 EDT 2016

System load: 0.16 Processes: 93
Usage of /: 5.0% of 18.75GB Users logged in: 0
Memory usage: 20% IF address for eth0: 172.31.10.242
Swap usage: 8%

Graph this data and manage this system at:
https://landscape.canonical.com/

0 packages can be updated.
0 updates are security updates.
```

34. No need to install sudo, sudo is installed by default on Ubuntu

35. Patch and update system
   a. sudo apt-get update
   b. sudo apt-get upgrade
      i. Allow any updates to install
   c. sudo apt-get dist-upgrade
      i. Allow any updates to install

36. Install VMware Open Tools
   a. Execute “sudo apt-get install open-vm-tools”
   b. Okay installing any dependencies

19. Install Various Linux Tools and Apps

Many of the tools are installed by default on Linux. Install the tools listed below

1. tcpdump
   a. Execute “sudo apt-get install tcpdump”
   b. Okay installing any dependencies
   c. Execute “sudo tcpdump –V”

Steve Jaworski, jaworski.steve@gmail.com
20. Install Splunk Free Enterprise

1. Using WinSCP or another SCP application upload the Splunk installation package to the Splunk Server
2. SSH to the Splunk Server
3. Navigate to the directory containing the Splunk package file
   ```bash
   $ pwd
   /home/supcapbox
   $ ls
   splunk-6.3.3-f44afece176d0-linux-2.6-amd64.deb
   $`
4. Install Splunk
   a. Execute “sudo dpkg --i Splunk Package File Name”
   ```bash
   $ sudo dpkg --i splunk-6.3.3-f44afece176d0-linux-2.6-amd64.deb
   Selecting previously unselected package splunk.
   (Reading database ... 30123 files and directories currently installed.)
   Preparing to unpack .../6.3.3-f44afece176d0-linux-2.6-amd64.deb ... complete
   $`
5. Start Splunk
   a. Execute “sudo /opt/splunk/bin/splunk start --accept-license --answer-yes”
   ```bash
   Waiting for web server at http://127.0.0.1:8000 to be available... Done
   If you get stuck, we’re here to help. Look for answers here: http://docs.splunk.com
   The Splunk web interface is at http://debian:8000
   `$
   b. Add Splunk to system startup
   i. Execute “sudo su -”
   ii. Execute “/opt/splunk/bin/splunk enable boot-start”

Steve Jaworski, jaworski.steve@gmail.com
iii. Execute “exit”

c. Verify Splunk is running
   i. Execute “sudo /opt/splunk/bin/splunk status”

6. From the management workstation, using Internet Explorer, Firefox, or Chrome to access Splunk

7. Create Admin password, document, and store securely
   a. Login with Admin and password “changeme”
   b. Add new password

8. Successful Login
21. **Install Splunk Apps**

1. From the management workstation, using Internet Explorer, Firefox, or Chrome access the Splunk Apps
   a. [https://splunkbase.splunk.com/](https://splunkbase.splunk.com/)
   b. Login

2. Search for these apps and download to the workstation
   a. Be sure to download the right version numbers
   b. **IMPORTANT NOTE:** There are most likely newer versions, however do not upgrade to the latest version if the older version is still available. If the installer is an advanced Splunk user, feel free to work with the latest version. Splunk Stream and Splunk CIM have version requirements. The download page shows which versions work with each other.
   c. Apps
      i. Splunk Stream 6.4.2
      ii. Splunk Common Information Model CIM 4.3.1
      iii. Technology Add-on for pfSense 2.0.6
      iv. Splunk Add-on for ISC Bind 1.0.0
      v. URL Toolbox 1.5
   d. Scroll to the bottom of the page and look for the Version dropdown
      i. If the version is still available, select it.
   e. Click the Download Button
   f. Review the License Agreements and Click Download
g. Save to a preferred location on the workstation

3. Select the Manage App Cog Icon in the upper left corner

4. Select Install app from file

5. The apps can be installed in any order, Splunk will need to be restarted after installing all the apps. Do not waste time rebooting Splunk between each app install.

6. After Uploading the first app, Choose Restart Later

7. Upload next app.

Steve Jaworski, jaworski.steve@gmail.com
8. When the last app is uploaded, choose Restart Splunk

9. After logging back in, review the list of installed apps and make sure they are there.

22. Create Index for DNS Logs

1. Log into Splunk
2. Choose Settings, then Indexes
3. Click New Index
4. Call the index “dns_tunnel_detection” and Click Save. Leave everything as defaults.
5. Look to see the index is in the list.

Steve Jaworski, jaworski.steve@gmail.com
23. Configure Splunk Listener

1. Log into the Splunk UI
2. Navigate to Settings, Then Forwarding and receiving
3. Select Configure receiving

Receive data
Configure this instance to receive data

Configure receiving

4. Click New

Receive data
Forwarding and receiving > Receive data

5. Enter 9996 for the port and click Save

6. Port 9996 should now show as enabled
24. Configure Stream App

1. Log into Splunk if not already logged in
2. Choose the Splunk Stream App. It can be found in two locations depending on the current location in Splunk.

<table>
<thead>
<tr>
<th>Default Page</th>
<th>If in another app such as search</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Default Page" /></td>
<td><img src="image2" alt="If in another app" /></td>
</tr>
</tbody>
</table>

3. Skip the tour, installers choice
4. Leave Defaults, Click Let’s get started

5. The main dashboard will display
6. Select Configuration and Configure Streams

7. Click New Stream

8. Choose Protocol DNS, Give the Name “Test_DNS_Tunnel_Detection”, Give a Description, and Click Next

9. Leave Aggregation Default as No and Click Next

10. Select Additional DNS Field not collected by default and Click Next
11. Do not add Filters, Click Next
12. Change Index to “dns_tunnel_detection”, make Status “Enabled”, and Click Next
   Settings
   Optionally, adjust Splunk App for Stream settings.

13. Leave defaultgroup and click Create Stream
14. Click Done

---

25. Configure URL Toolbox

1. SSH to the Splunk server
2. Execute “sudo su -“
3. Change to utbox directory
   a. Execute “cd /opt/splunk/etc/apps/utbox/bin”
4. Make backup copy of suffix_list_custom.dat
   a. Execute “cp –a suffix_list_custom.dat suffix_list_custom.dat.bak”
5. Make sure backup file was created
   a. Execute “ls -l”

```
-rw-r--r-- 1 root root  7615 May 15 17:05 suffix_list_custom.dat
-rw-r--r-- 1 root root  7615 May 15 17:05 suffix_list_custom.dat.bak
```
6. Using a text editor, edit suffix_list_custom.dat
   a. Add “lan” and “local” to the list

Steve Jaworski, jaworski.steve@gmail.com
b. **IMPORTANT NOTE:** This list has a large amount of TLDs, if the lab installer is using a different domain, make sure it’s in the list.

7. Run diff between active and backup file
   a. Execute “diff suffix_list_custom.dat suffix_list_custom.dat.bak”

26. Configure Splunk Add-on for ISC Bind 1.0.0

1. Create pfsense index
2. Login to the Splunk UI
3. Follow same steps as outlined in “Create Index for DNS Logs”
4. Call the index “bind”
5. SSH to the Splunk server
6. Execute “sudo su –”
7. Change to Splunk_TA_isc-bind” directory
   a. Execute “/opt/splunk/etc/apps/Splunk_TA_isc-bind"
   b. Execute “mkdir local”
   c. Execute “cd ./local”
   d. Execute “touch transforms.conf”
   e. Using a Text editor open the transforms.conf file
   f. Paste the data from this table into the file and save.

```
[isc_bind_query_extract_field_0]
#REGEX = (?:(?:s+queries:)?(?:s+(^[^:]+):)?s+client\s+([\w\-\.:\[\]\{\}]+)(?:s+view[s+\[\]\{\}]+)?s+query:\s+(?:([\w\-\.:\[\]\{\}]+))\s+response_code:\s+(?:([\w\-\.:\[\]\{\}]+))\s+flags:\s+(?:([\w\-\.:\[\]\{\}]+))\s+file_name:\s+(?:([\w\-\.:\[\]\{\}]+))\s+file_location:\s+(?:([\w\-\.:\[\]\{\}]+))\s+)(?:for\s+(?:([\w\-\.:\[\]\{\}]+))\s+((?:a|d|l|s)[^\s]+))?)$ 

[isc_bind_queryerror_extract_field_0]
REGEX = (?:(?:s+query-errors:)?(?:s+(^[^:]+):)?s+client\s+([\w\-\.:\[\]\{\}]+)(?:s+view[s+\[\]\{\}]+)?s+query:\s+failed\s+response_code:\s+(?:([\w\-\.:\[\]\{\}]+))\s+flags:\s+(?:([\w\-\.:\[\]\{\}]+))\s+file_name:\s+(?:([\w\-\.:\[\]\{\}]+))\s+file_location:\s+(?:([\w\-\.:\[\]\{\}]+))\s+(?:for\s+s+([\w\-\.:\[\]\{\}]+))?)$ 

[isc_bind_lameserver_extract_field_0]
REGEX = (?:(?:s+error-servers:)?(?:s+(^[^:]+):)?s+error\s+resolving\s+error\s+(?:[\w\-\.:\[\]\{\}]+))\s+resolved\s+((?:a|d|l|s)[^\s]+))?)$ 

```

Steve Jaworski, jaworski.steve@gmail.com
Using Splunk to Detect DNS Tunneling

Steve Jaworski, jaworski.steve@gmail.com

8. **IMPORTANT NOTE:** The default field extractions do not quite work correctly. This is the modification to make it work correctly. Make sure the transforms.conf file is in the local directory. In Splunk the local directory configuration, precede the default directory configurations.
9. Perform a diff between the default transforms.conf and local transforms.conf
   a. From the (/opt/splunk/etc/apps/Splunk_TA_isc-bind ) directory Execute “diff ./local/transforms.conf ./default/transforms”
   b. The output below shows which regular expressions were modified to fix the extractions

   ```
   #REGEX = (?::s+queries:) (?::s+([^::]+):)?::s+client::s+(\[\w-\.
   :\]\{1,100\})\#d\{1,5\}(?:s+\{[^
   ]\}+\})?:(?::s+view::s+[^::]+:)?::s+query::s+(\(?::\w-
   \.:\]\{1,100\})\]?s+\{[\w\s]*\}\]s+\{([\w\-\.
   :])\}\{1,100\})\]?s+\{([\w\s]*\}\]s+\{([\w\-\.
   :])\}\}$
   < #FORMAT = vendor_severity::$1 src::$2 src_port::$3 query::$4 record_class::$5
   record_type::$6 flag::$7 dest::$8
   ---
   > REGEX = (?::s+queries:) (?::s+([^::]+):)?::s+client::s+(\[\w-
   \.:\]\{1,100\})\#d\{1,5\}(?:s+\{[^
   ]\}+\})?:(?::s+view::s+[^::]+:)?::s+query::s+(\(?::\w-
   \.:\]\{1,100\})\]?s+\{[\w\s]*\}\]s+\{([\w\-\.
   :])\}\{1,100\})\]?s+\{([\w\s]*\}\]s+\{([\w\-\.
   :])\}\}\
   > FORMAT = vendor_severity::$1 src::$2 src_port::$3 query::$4 record_class::$5
   record_type::$6 flag::$7 dest::$8
   ```

10. Another option is to copy the transforms.conf file from the default directory to the local directory of the ISC Bind App. Then just replace the REGEX and FORMAT for stanza [isc_bind_query_extract_field_0].

27. Configure Splunk pfSense Add-On

   1. Create pfsense index
   2. Login to the Splunk UI
   3. Follow same steps as outlined in “Create Index for DNS Logs”
   4. Call the index “pfsense”
   5. Setup Splunk Listener
   6. SSH to the Splunk server
   7. Execute “sudo su –”
   8. Change directory to pfSense
      a. Execute “/opt/splunk/etc/apps/TA-pfsense”
   9. Create local directory
      a. Execute “mkdir local”
   10. Create inputs file
       a. Execute “touch inputs.conf”
   11. Using a text editor, edit inputs.conf
       a. Copy configuration in table below, Change IP address to match Splunk Server

```
[udp://172.31.10.199:516]
index=pfsense
sourcetype = pfsense
```

Steve Jaworski, jaworski.steve@gmail.com
**IMPORTANT NOTE:** Port 516 was chosen over port 514, not to conflict with other syslog configurations. In a production environment, having Splunk run the syslog collector is a bad idea. Every time Splunk restarts so does the syslog listener, resulting in lost data. Best practice is to setup a syslog-ng or rsyslog server, then install Splunk to monitor the files generated by syslog.

12. Restart Splunk
   a. `service splunk restart`
   b. OR
   c. `/opt/splunk/bin/splunk restart`

28. **Install Bind 9**

1. SSH to the Bind DNS Server
2. Execute “`sudo su -“”
3. Execute “`apt-get install bind9 bind9-doc bind9utils dnsutils`”
4. Install any dependencies

29. **Configure Bind 9**

**Important Note:** Adjust any IP Address and Subnets for the lab network.

1. SSH to the Bind DNS Server
2. Execute “`sudo su -“”
3. Execute “`cd /etc/bind`”
4. Using text editor, edit `named.conf.options`
5. Add any missing information from the table below. Save the file

```plaintext
acl trusted {
    172.16.0.0/12;
    127.0.0.1/32;
    localhost;
    localnets;
};

options {
    directory "/var/cache/bind";
    max-cache-size 2m; //maximum cache size of 2 MB
    cleaning-interval 1; //clean cache every 1 minute
    recursion yes;
    allow-query { any; }
    allow-recursion { trusted; }
    allow-query-cache { trusted; }
    edns-udp-size 512;

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk. See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
```

Steve Jaworski, jaworski.steve@gmail.com
// nameservers, you probably want to use them as forwarders.
// Uncomment the following block, and insert the addresses replacing
// the all-0's placeholder.

forwarders {
    8.8.8.8;
    8.8.4.4;
};

//============================================================
// If BIND logs error messages about the root key being expired,
// you will need to update your keys. See https://www.isc.org/bind-keys
//============================================================

dnssec-validation auto;

auth-nxdomain no;  # conform to RFC1035
listen-on-v6 { any; }
;
6. Create file named.conf.log, add the configuration from the table below

logging {
    channel default_channel {
        file "/var/log/named/default.log";
        print-time yes;
        print-category yes;
        print-severity yes;
        severity dynamic;
    };
    channel general_channel {
        file "/var/log/named/general.log";
        print-time yes;
        print-category yes;
        print-severity yes;
        severity dynamic;
    };
    channel notify_channel {
        file "/var/log/named/notify.log";
        print-time yes;
        print-category yes;
        print-severity yes;
        severity dynamic;
    };
    channel network_channel {
        file "/var/log/named/network.log";
        print-time yes;
        print-category yes;
        print-severity yes;
        severity dynamic;
    };
    channel queries_channel {
        file "/var/log/named/queries.log";
        print-time yes;

Steve Jaworski, jaworski.steve@gmail.com
Using Splunk to Detect DNS Tunneling

print-category yes;
print-severity yes;
severity dynamic;
}
channel query-errors_channel {
    file "/var/log/named/query-errors.log";
    print-time yes;
    print-category yes;
    print-severity yes;
    severity dynamic;
}
channel lame-servers_channel {
    file "/var/log/named/lame-servers.log";
    print-time yes;
    print-category yes;
    print-severity yes;
    severity dynamic;
}

7. Create file db.security.local, add the configuration from the table below
a. Important Note: This is the file that maps domain security.local to the DNS tunnel servers.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>@</td>
<td>IN</td>
<td>SOA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>security.local. root.security.local. (</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1402201600 ; Serial</td>
</tr>
<tr>
<td></td>
<td></td>
<td>604800 ; Refresh</td>
</tr>
<tr>
<td></td>
<td></td>
<td>86400 ; Retry</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2419200 ; Expire</td>
</tr>
<tr>
<td></td>
<td></td>
<td>604800 ) ; Negative Cache TTL</td>
</tr>
<tr>
<td>@</td>
<td>IN</td>
<td>NS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ns1.security.local.</td>
</tr>
<tr>
<td>ns1</td>
<td>IN</td>
<td>A</td>
</tr>
<tr>
<td>t30</td>
<td>IN</td>
<td>A</td>
</tr>
<tr>
<td>t40</td>
<td>IN</td>
<td>A</td>
</tr>
<tr>
<td>t1</td>
<td>IN</td>
<td>NS</td>
</tr>
<tr>
<td>t1ns</td>
<td>IN</td>
<td>A</td>
</tr>
<tr>
<td>t1ns</td>
<td>IN</td>
<td>A</td>
</tr>
<tr>
<td>t2</td>
<td>IN</td>
<td>NS</td>
</tr>
<tr>
<td>t2ns</td>
<td>IN</td>
<td>A</td>
</tr>
</tbody>
</table>

8. Edit the named.conf file, Add any missing entries from the table below

Steve Jaworski, jaworski.steve@gmail.com
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the // structure of BIND configuration files in Debian, *BEFORE* you customize // this configuration file.

// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
include "/etc/bind/named.conf.log";

9. Edit the named.conf.local file. Add any missing entries from the table below

//Manage Log Files
//include "/etc/bind/named.conf.log";

// Do any local configuration here

// Consider adding the 1918 zones here, if they are not used in your // organization
//include "/etc/bind/zones.rfc1918";

//My Zones
zone "security.local" {
    type master;
    file "/etc/bind/db.security.local";
    forwarders { };
};

zone "10.31.172.in-addr.arpa" {
    type master;
    notify no;
    file "/etc/bind/db.10.31.172";
};

zone "20.31.172.in-addr.arpa" {
    type master;
    notify no;
    file "/etc/bind/db.20.31.172";
};

zone "30.31.172.in-addr.arpa" {
    type master;
    notify no;
    file "/etc/bind/db.30.31.172";
};

zone "40.31.172.in-addr.arpa" {
//    type master;
//    notify no;
//    file "/etc/bind/db.40.31.172";
//}

10. Create file db.10.31.172 and add this configuration from the table below

Steve Jaworski, jaworski.steve@gmail.com
Using Splunk to Detect DNS Tunneling

Steve Jaworski, jaworski.steve@gmail.com

11. Create file db.20.31.172 and add this configuration from the table below

```
; BIND reverse data file for local loopback interface
;
@       IN      SOA     security.local. root.security.local. ( 
    1402201600 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    604800 ) ; Negative Cache TTL
;
@       IN      NS      ns1.security.local.
2       IN      PTR     dot2.security.local.
3       IN      PTR     dot3.security.local.
4       IN      PTR     dot4.security.local.
5       IN      PTR     dot5.security.local.
```

12. Create file db.30.31.172 and add this configuration from the table below

```
; BIND reverse data file for local loopback interface
;
@       IN      SOA     security.local. root.security.local. ( 
    1402201602 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    604800 ) ; Negative Cache TTL
;
@       IN      NS      ns1.security.local.
2       IN      PTR     ns1.security.local.
```

13. Create file db.40.31.172 and add this configuration from the table below

```
; BIND reverse data file for local loopback interface
;
@       IN      SOA     security.local. root.security.local. ( 
    1112201602 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    604800 ) ; Negative Cache TTL
;
```

Steve Jaworski, jaworski.steve@gmail.com
Create Logging Location for Bind Logs
   a. Execute “cd /var/log”
   b. Execute “mkdir named”
   c. Execute “chown -R root:bind /var/log/named”
   d. Execute “chmod -R 775 /var/log/named/”

Configure Local Resolver
   a. Execute “cd /etc”
   b. Edit resolv.conf and add the configuration from the table below

<table>
<thead>
<tr>
<th>domain</th>
<th>search</th>
<th>nameserver</th>
</tr>
</thead>
<tbody>
<tr>
<td>lan</td>
<td>lan</td>
<td>127.0.0.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>172.31.10.1</td>
</tr>
</tbody>
</table>

Start Bind
   a. Execute “service bind9 restart”
   b. Verify Bind is Running
      i. Execute “service bind9 status”

Verify resolver is working
   a. Execute “nslookup”
   b. Query List below, each should resolve
      i. www.google.com
      ii. ns1.security.local
      iii. 172.16.20.60
18. Verify Logging is working
   a. Execute “cd /var/log/named”
   b. Execute “ls -ltrah”

19. If bind will not start or displaying errors, review the steps and configuration again. The links below were the resources used to configure bind for the DNS Tunnel Lab

Helpful Links to Get Bind up and running
https://wiki.debian.org/Bind9#File_2Fetc_2Fbind_2Fnamed.conf.log
https://debian-administration.org/article/355/Two-in-one_DNS_server_with_BIND9
https://www.digitalocean.com/community/tutorials/how-to-configure-bind-as-a-caching-or-forwarding-dns-server-on-ubuntu-14-04

30. Install Splunk Universal Forwarder

Steve Jaworski, jaworski.steve@gmail.com
1. Download Universal Forwarder from Splunk
   a. https://www.splunk.com/page/previous_releases/universalforwarder
   b. Click on Linux X86_64

   - Linux x86
   - Linux x86_64
   - Linux PPC
   - Linux s390x

c. Download the matching version to Splunk Enterprise. In this case, choose the 6.3.3 deb package.

2. SCP or SFTP to Bind Server
   a. Upload the Splunk UF
   b. Also, upload the Splunk ISC Bind Technology Add-on that was downloaded for the Splunk Server
      i. File Name = splunk-add-on-for-isc-bind_100.tgz

3. SSH to the Bind Server
4. Sudo to Root, Execute “sudo su –“
5. Install Heavy Forwarder
   a. Navigate to directory, where the Splunk UF was uploaded
   b. Execute “dpkg -i splunkforwarder-6.3.3-f44afce176d0-linux-2.6-amd64.deb”

6. Start Heavy Forwarder
   a. Execute “/opt/splunkforwarder/bin/splunk start --accept-license”

Steve Jaworski, jaworski.steve@gmail.com
7. Add to startup
   a. Execute “/opt/splunkforwarder/bin/splunk enable boot-start”

31. Configure Splunk Universal Forwarder

1. SSH to the Bind Server
2. Sudo to Root, Execute “sudo su –”
3. Point Forwarder to Splunk Server
   a. Execute “cd /opt/splunkforwarder/etc/system/local”
   b. Execute “touch inputs.conf”
   c. Using a text editor open outputs.conf”, copy the config in the table below, and save the file
   d. IMPORTANT NOTE: Change the IP address of the Splunk server if appropriate.

   ```plaintext
   [tcpout]
   defaultGroup = default-autolb-group
   
   [tcpout:default-autolb-group]
   server = 172.31.10.176:9996
   ```

4. Install the Splunk ISC Bind TA
   a. Move to the directory where the file splunk-add-on-for-isc-bind_100.tgz was uploaded.
   b. Execute “tar -xvf splunk-add-on-for-isc-bind_100.tgz -C /opt/splunkforwarder/etc/apps/”
   c. Execute “cd /opt/splunkforwarder/etc/apps/Splunk_TA_isc-bind”
   d. Execute “mkdir local”
   e. Execute “cd local”
   f. Execute “touch inputs.conf”

Steve Jaworski, jaworski.steve@gmail.com
Using Splunk to Detect DNS Tunneling

Steve Jaworski, jaworski.steve@gmail.com

Using a text editor open inputs.conf”, copy the config in the table below, and save the file

<table>
<thead>
<tr>
<th>Source</th>
<th>Type</th>
<th>Disabled</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>[monitor:///var/log/named/queries.log]</td>
<td>source = isc:bind:query, disabled = 0, index = bind</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[monitor:///var/log/named/query-errors.log]</td>
<td>source = isc:bind:queryerror, disabled = 0, index = bind</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[monitor:///var/log/named/network.log]</td>
<td>source = isc:bind:network, disabled = 0, index = bind</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[monitor:///var/log/named/notify.log]</td>
<td>source = isc:bind:transfer, disabled = 0, index = bind</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[monitor:///var/log/named/lame-servers.log]</td>
<td>source = isc:bind:lasserver, disabled = 0, index = bind</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. Restart the Forwarder
   a. Execute “service splunk restart “
   b. OR
   c. Execute “/opt/splunforwarder/bin/splunk restart”

32. Install Iodine

1. SSH to DNS Client
2. Execute “sudo apt-get iodine
3. Verify Iodine is installed
4. Execute “sudo iodine -v”

5. Repeat steps on DNS Tunnel Server

33. Install DNSCAT2 Server

1. SSH to the Ubuntu Server
2. Sudo to root, Execute “sudo su -“
3. Install git, make, and g++
   a. Execute “apt-get -y install git make g++”

4. Install Ruby Repository from bright box .dot
   a. Execute “sudo apt-get install software-properties-common”
   b. Execute “sudo apt-add-repository ppa:brightbox/ruby-ng”
   c. Execute “sudo apt-get update”

5. Install Ruby and Ruby-Dev
   a. Execute “apt-get install ruby2.2”
   b. Execute “apt-get install ruby2.2-dev”

6. Install Bundler
   a. Execute “gem2.2 install bundler”

7. Clone DNScat2 from GitHub
   a. Execute “cd /opt”
   b. Execute “git clone https://github.com/iagox86/dnscat2.git”
   c. Execute “ls –l”
      i. dnscat2 directory should be in /opt

8. Build DNSCAT2
   a. Execute “cd /opt/dnscat2/server”
   b. Execute “bundle install”

9. Test Ruby
   a. Execute “ruby dnscat2.rb –help

Steve Jaworski, jaworski.steve@gmail.com
34. **Install DNSCAT2 Client**

1. SSH to DNS Client Machine
2. Sudo to root, Execute “sudo su -“
3. Install git, make, and g++
   a. Execute “apt-get -y install git make g++”
4. Clone DNScat2 from GitHub
   a. Execute “cd /opt”
   b. Execute “git clone https://github.com/iagox86/dnscat2.git”

```
root@debian:/opt# git clone https://github.com/iagox86/dnscat2.git
Cloning into 'dnscat2'... Protocol 'https' supports DNS over HTTPS
remote: Counting objects: 6476, done.
remote: Total 6476 (delta 0), reused 0 (delta 0), pack-reused 6476
remote: Receiving objects: 100% (6476/6476), 3.78 MiB | 2.15 MiB/s, done.
remote: Resolving deltas: 100% (4474/4474), done.
root@debian:/opt#  
```

5. Build DNSCAT2 Client
   a. Execute “cd /opt/dnscat2/client”
   b. Execute “make”

---

Steve Jaworski, jaworski.steve@gmail.com

Helpful Links

- [https://www.brightbox.com/docs/ruby/ubuntu/](https://www.brightbox.com/docs/ruby/ubuntu/)
- [https://zeltser.com/c2-dns-tunneling/](https://zeltser.com/c2-dns-tunneling/)
6. Start Client  
   a. Execute “./dnscat -v”

   ```bash
   root@debian:/opt/dnscat2/client# ./dnscat -v
dnscat2 v0.05 (client)
   root@debian:/opt/dnscat2/client#
   ```

Helpful Links  
https://github.com/iagox86/dnscat2/blob/master/README.md#client

35. Alexa Top 1 Million Download and Modify

1. SSH to the DNS Tunnel Client Machine  
2. Execute “cd /opt”  
3. Execute “mkdir /1milldomains”  
4. Execute “cd /1milldomains”  
5. Execute “wget http://s3.amazonaws.com/alexa-static/top-1m.csv.zip”  
6. Execute “unzip top-1m.csv.zip”  
7. Execute “tail top-1m.csv”

   ```bash
   root@dnstunnelclient:/opt/
   999991,city.ck.ua
   999992,globalink.ca
   999993,gestirent.it
   999994,naturwarenkaufhaus.e
   999995,salmaster.net
   999996,akukong.com
   999997,altonsports.co.kr
   999998,supplylogix.com
   999999,sigovs.com
   1000000,diyhomelife.com
   ```

8. Execute “sed 's/\([0-9]*\),/\g' top-1m.csv >> domainnames.txt”  
9. Execute “tail domainnames.txt”

Steve Jaworski, jaworski.steve@gmail.com
Using Splunk to Detect DNS Tunneling

Steve Jaworski, jaworski.steve@gmail.com

Execute “wc -l domainnames.txt”, wc counted 1 million rows

```
1000000 domainnames.txt
```

Execute “split -n 10 domainnames.txt”

Execute “ls”, There are now 10 files starting with x

```
-rw-r--r-- 1 root root 1531436 May 15 12:51 xaa
-rw-r--r-- 1 root root 22203263 May 14 22:33 top-lm.csv
-rw-r--r-- 1 root root 9976083 May 15 01:55 top-lm.csv.zip
```

13. Execute “tail and wc -l “on one of the X files, Notice how the domains are evenly split.

```
Root@dnstunnelclient:/opt/testdomain# tail xaa
oceans-nadia.com	notre-planete.info
gihs.org
spilemax.de
dansdelhi.com
lahey.org
palcomp3.top
freemostation.com
citydc.com
arcoct@dnstunnelclient:/opt/testdomain# wc -l xaa
100679 xaa
```

14. The Top 1 Million Domains are ready for use.

36. Install and Configure DNS Grind 1.0

1. SSH to DNS Client
2. Sudo to Root
3. Install perl modules
   a. Execute “perl -MCPAN -e shell
   b. Choose Yes for automatic configuration

Steve Jaworski, jaworski.steve@gmail.com
c. Install Any Missing Modules
   i. Net::DNS
   ii. Socket
   iii. IO::Handle
   iv. IO::Select
   v. Getopt::Std
      1. Execute “install Net::DNS”
   vi. Repeat for each module
   vii. Some Modules may already be installed
   viii. Execute “Exit”

4. Download Grind
   a. Execute “cd /opt”
   b. Execute “wget http://pentestmonkey.net/tools/dns-grind/dns-grind-1.0.tar.gz”
   c. Execute “tar xvf dns-grind-1.0.tar.gz”
   d. Execute “cd ./dns-grind-1.0/”
   e. Execute “./dns-grind.pl -h”

Helpful Links:
http://pentestmonkey.net/tools/misc/dns-grind

37. Start an Iodine tunnel
1. SSH to the DNSTunnelServer and DNSTunnelClient
2. On the Server
   a. Execute “sudo iodined -f -P letstunnel 10.10.10.1 t1.security.local”
   b. Look for this output

3. On the Client
   a. Make sure the /etc/resolv.conf file is pointed to the Bind DNS server

Steve Jaworski, jaworski.steve@gmail.com
b. Look for this output

```
$ cat /etc/resolv.conf
domain security.local
search security.local
nameserver 172.31.20.60
```

c. Execute “sudo iodine -f -P letstunnel -r t1.security.local”

d. Look for this output

```
[sudo] password for s0apbox:
Opened dns0
Opened IPv4 UDP socket
Sending DNS queries for t1.security.local to 172.31.20.60
Autodetecting DNS query type (use -I to override).
Using DNS type NULL queries
Version ok, both using protocol v 0x00000002. You are user #0
Setting IP of dns0 to 10.10.10.2
Setting MTU of dns0 to 1130
Server tunnel IP is 10.10.10.1
Skipping raw mode
Using DNSO extension
Switching upstream to codes Base128
Server switched upstream to codes Base128
No alternative downstream codes available, using default: (Raw)
Switching to lazy mode for low-latency
Server switched to lazy mode
Autoprobing max downstream fragment size... (skip with -m fragsize)
768 ok, 1152 ok, ...1536 not ok...1248 not ok...1200 not ok...1152 ok...1104 ok...1088 ok...1072 ok...1024 ok...1000 ok...992 ok...960 ok...944 ok...928 ok...912 ok...896 ok...880 ok...864 ok...848 ok...832 ok...816 ok...800 ok...784 ok...768 ok...752 ok...720 ok...704 ok...680 ok...664 ok...640 ok...624 ok...608 ok...592 ok...576 ok...560 ok...544 ok...528 ok...512 ok...504 ok...496 ok...480 ok...464 ok...448 ok...432 ok...416 ok...400 ok...384 ok...368 ok...352 ok...336 ok...320 ok...304 ok...288 ok...272 ok...256 ok...240 ok...224 ok...208 ok...192 ok...176 ok...160 ok...144 ok...128 ok...112 ok...96 ok...80 ok...64 ok...48 ok...32 ok...16 ok...8 ok...4 ok...2 ok...
Setting downstream fragment size to max 1186...
Connection setup complete, transmitting data.
```

4. Open a second SSH session to the client
   a. Execute “hostname”
   b. Execute “ssh yourusername@10.10.10.1”
   c. Login
   d. Execute “hostname”
   e. Look for this output

```
$s0apbox@dntunnelclient:$ hostname
dntunnelclient
$s0apbox@dntunnelclient:~ $ ssh s0apbox@310.10.10.1
s0apbox@310.10.10.1's password:
The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.
No mail.
Last login: Fri May 20 20:40:30 2016 from 10.10.10.2
$s0apbox@dntunnelserver:$ hostname
dntunnelserver
$s0apbox@dntunnelserver:~ $
```

Steve Jaworski, jaworski.steve@gmail.com
Helpful Links:
http://code.kryo.se/iodine/README.html

Steve Jaworski, jaworski.steve@gmail.com
38. Start a DNSECAT2 tunnel

1. SSH to the DNSTunnelServer/DNSECAT2 and DNSTunnelClient
   a. On the Server
      i. sudo to root
      ii. CD to the DNSECAT2 / Server Directory
      iii. Execute “ruby dnscat2.rb -d domain=t2.security.local -c test -u -k”
      iv. Look for this output

2. On the Client
   i. CD to DNSECAT 2 / Client Directory
   ii. Execute “sudo ./dnscat --secret=test t2.security.local”
   iii. Look for this output

3. On the Server
   i. Follow the instructions to Ping in the output below
Output on the Client

```
Session established!
Got a command: COMMAND_PING [request] :: request_id: 0x0001 :: data: ARIHVEDEXHQLVJG9M4VJHLYIDEvYX8
[[ WARNING ]] :: Got a ping request! Responding!
Response: COMMAND_PING [response] :: request_id: 0x0001 :: data: ARIHVEDEXHQLVJG9M4VJHLYIDEvYX8
```

Helpful Links:
https://github.com/iagox86/dnscat2/blob/master/README.md
https://zeltser.com/c2-dns-tunneling/

Steve Jaworski, jaworski.steve@gmail.com
39. Basic Splunk Searches

1. Log into the Splunk UI
2. Launch the Search and Reporting App

3. View All Data
   a. In the Search window type in “*”
   b. Click Search, Magnifying Glass
   c. See the results in the bottom

4. View DNS Bind Data Only
   a. In the Search window type in “sourcetype=isc*”
   b. Click Search, Magnifying Glass
   c. Click Event Field Sourcetype (column on the left)
d. Look at the different Bind Sources

5. View Stream App Data
   a. In the Search window type in “sourcetype=stream*”
   b. Click Search, Magnifying Glass
   c. Click Event Field Sourcetype (column on the left)

   d. Look at the different Stream Sources
   e. Click Event Field Source (column on the left)
f. Look at the different stream sources
   i. Notice the Test_DNS_Tunnel_Detection
   ii. This source was configured in the Configure Stream App Section

6. View pfSense
   a. Repeat similar steps above
   b. sourcetype=pfSense*
   c. Output should look similar

Steve Jaworski, jaworski.steve@gmail.com
7. If the searches here work, all the searches in the core of the paper (pages 1 to 30) should also work.

40. Troubleshooting

1. Routing
   a. Static routes may need to be added to the upstream router or management workstation
   b. Anything outside the virtual environment will need to know how to route into the lab
   c. Running route print on the management workstation, persistent routes were added to know how to get to the lab virtual machines. The gateway address 172.31.10.199 is the external side of the pfSense firewall.

<table>
<thead>
<tr>
<th>Persistent Routes:</th>
<th>Network Address</th>
<th>Netmask</th>
<th>Gateway Address</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>169.254.0.0</td>
<td>255.255.0.0</td>
<td>169.254.177.219</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>169.254.0.0</td>
<td>255.255.0.0</td>
<td>172.31.10.103</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>169.254.0.0</td>
<td>255.255.0.0</td>
<td>192.168.137.1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>172.31.40.0</td>
<td>255.255.255.0</td>
<td>172.31.10.199</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>172.31.30.0</td>
<td>255.255.255.0</td>
<td>172.31.10.199</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>172.31.20.0</td>
<td>255.255.255.0</td>
<td>172.31.10.199</td>
<td>1</td>
</tr>
</tbody>
</table>

Steve Jaworski, jaworski.steve@gmail.com
2. **Splunk Stream not pulling data from all interfaces**
   a. By default Splunk Stream pulls data from all interfaces
   b. Make sure all interfaces are up
   c. To bring an interface up
      i. Execute “sudo ifconfig *interfacename* up”
   d. If the interface was down, while Splunk was running, Splunk needs to be restarted
   e. Configure the interface to be up at boot
      i. Edit `/etc/networking/interfaces`

```
allow-hotplug eth1
iface eth1 inet manual
pre-up ifconfig $IFACE up
post-down ifconfig $IFACE down
```

Helpful Links
http://docs.splunk.com/Documentation/StreamApp/6.3.0/DeployStreamApp/ConfigureStreamForwarder
<table>
<thead>
<tr>
<th>Event Name</th>
<th>Location</th>
<th>Dates</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enterprise Defense Summit &amp; Training 2019</td>
<td>Redondo Beach, CAUS</td>
<td>Jun 03, 2019 - Jun 10, 2019</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Zurich June 2019</td>
<td>Zurich, CH</td>
<td>Jun 03, 2019 - Jun 08, 2019</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Munich July 2019</td>
<td>Munich, DE</td>
<td>Jul 01, 2019 - Jul 06, 2019</td>
<td>Live Event</td>
</tr>
<tr>
<td>SEC450 Security Ops-Analysis Beta 1</td>
<td>Crystal City, VAUS</td>
<td>Jul 08, 2019 - Jul 13, 2019</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Pittsburgh 2019</td>
<td>Pittsburgh, PAUS</td>
<td>Jul 08, 2019 - Jul 13, 2019</td>
<td>Live Event</td>
</tr>
<tr>
<td>DFIR Summit &amp; Training 2019</td>
<td>Austin, TXUS</td>
<td>Jul 25, 2019 - Aug 01, 2019</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Riyadh July 2019</td>
<td>Riyadh, SA</td>
<td>Jul 28, 2019 - Aug 01, 2019</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Boston Summer 2019</td>
<td>Boston, MAUS</td>
<td>Jul 29, 2019 - Aug 03, 2019</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Prague August 2019</td>
<td>Prague, CZ</td>
<td>Aug 12, 2019 - Aug 17, 2019</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Minneapolis 2019</td>
<td>Minneapolis, MNUS</td>
<td>Aug 12, 2019 - Aug 17, 2019</td>
<td>Live Event</td>
</tr>
<tr>
<td>Security Writing NYC: SEC402 Beta 2</td>
<td>OnlineNYUS</td>
<td>Jun 01, 2019 - Jun 02, 2019</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS OnDemand</td>
<td>Books &amp; MP3s OnlyUS</td>
<td>Anytime</td>
<td>Self Paced</td>
</tr>
</tbody>
</table>