Terminal--Snort Challenge (Part 1)

# What you can learn from this

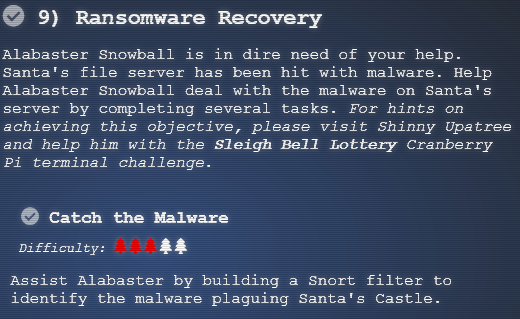
The [Snort Intrusion Detection System](https://www.snort.org/) (IDS) was one of the first open source IDSs. Snort’s rule system is now the de facto standard for the industry. An IDS is somewhat like antivirus for network traffic, in that it has rules based on signatures of network traffic that is known to be bad. An IDS incorporates other detection methods, such as IP address and domain name reputation lists and protocol analysis, but we will concentrate on rules.

In this exercise you will write a rule to detect the WannaCookie ransomware that has infected Kringle Castle. The emphasis is on writing a rule that is as general as possible to catch changes in the malware, but specific enough that it does not generate false positives. In this case, we cannot write a rule based on IP address or domain name, since these addresses change frequently. Instead we need to find the major characteristics of the packet and write a rule for those.

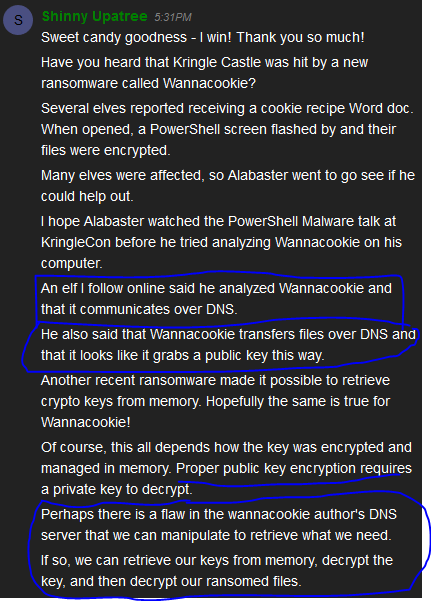
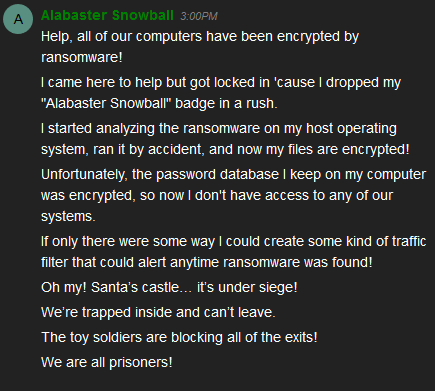
You will also see Regular Expressions used for matching. Regular expressions (or regex) are like wild cards on steroids. You can write incredibly detailed (and complicated) regular expressions that will match only what you want to match. Fortunately, our regex will be fairly simple

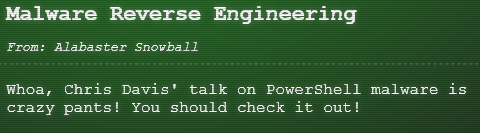
# Getting Started

Since you have solved the door scanner and forged a QR code for yourself, you can access Santa’s Secret Room. Alabaster will ask you to write a Snort rule.  


The objective is here. Notice that there are several steps to Objective 9.  


# Hints

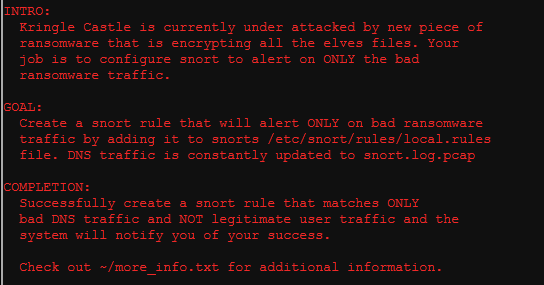
Both Alabaster and Shinny have important things to tell us.  


Alabaster also has a hint about Malware Reverse Engineering, but we will use that later. Right now, the important hints are that the malware communicates over DNS, and that we must write a Snort rule to stop it.  


# Getting started

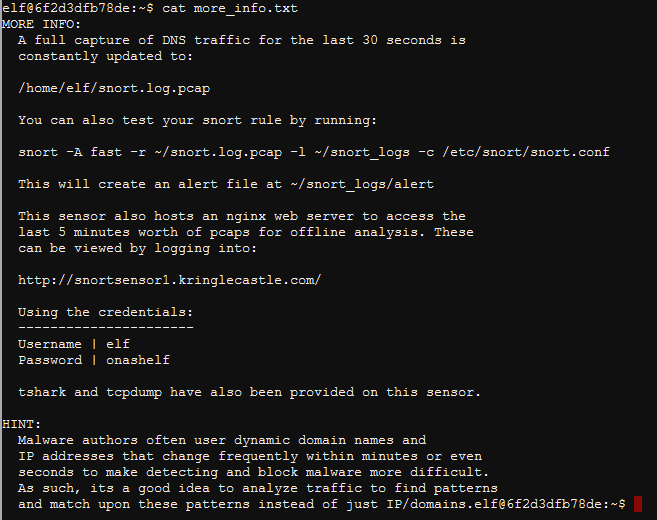
When you enter the terminal you will see some basic information you need to evaluate the malware network traffic. The opening screen will give you some important information.

GOAL: Create a snort rule that will alert ONLY on bad ransomware traffic  
Put the rule in /etc/snort/rules/local.rules on the terminal  
Check out ~/more\_info.txt for additional information



The moreinfo.txt file has additional tidbits.

A full capture of DNS traffic for the last 30 seconds is constantly updated to: /home/elf/snort.log.pcap  
test your snort rule by running:  
snort -A fast -r ~/snort.log.pcap -l ~/snort\_logs -c /etc/snort/snort.conf  
This will create an alert file at ~/snort\_logs/alert  
Note: there will also be a pcap file in ~/snort\_logs/ that will show you which packets your caught. Tshark and tcpdump have also been provided on this sensor so you can examine this pcap with caught packets.  
  
You can also download pcaps for offline analysis. You can examine the file in Wireshark to get ideas for rule creation  
<http://snortsensor1.kringlecastle.com/>   
Username: elf   
Password: onashelf



# The next step

Go to the Snort sensor link and download a pcap for analysis.

# Hand in

1. What is consistent from one packet to the next, that can be part of your rule? Remember, IP address and the domain of the server (like blahblah.com) can change and cause your rule to fail.
2. Is the port number always the same? Is the layer 4 protocol the same? What about the upper layer protocol?
3. Note: In DNS, if you look at the packet bytes pane (the bottom pane) you will see that the ascii for “period” never appears in the domain. Instead it is a hex number that gives the number of bytes in the next section. For example, [www.google.com](http://www.google.com) will be 03 www 06 google 03 com in the bytes pane. Is there anything consistent with those numbers?