

Terminal--Snort Challenge (Part 1)

What you can learn from this

The [Snort Intrusion Detection System](#) (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.

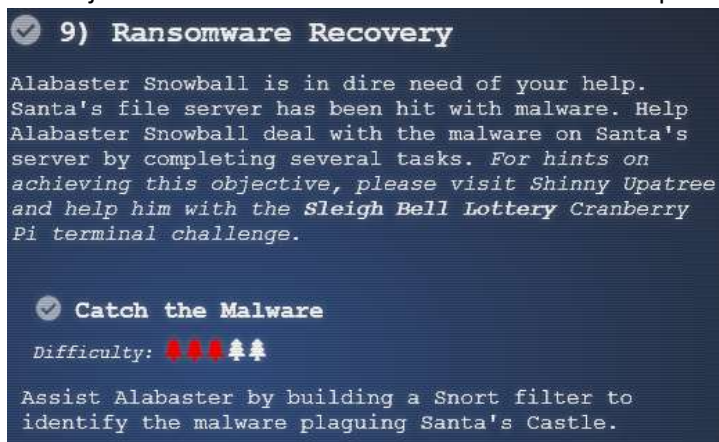
You will also see Regular Expressions used for matching. Regular expressions (or regex) are like wild cards on steroids. You can write incredibly detailed (and complicated) regular expressions that will match only what you want to match. Fortunately, our regex will be fairly simple

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.

The screenshot shows a chat interface with two messages. The first message is from Alabaster Snowball at 3:00PM, where he explains that his computers are encrypted by ransomware and that he is stuck because he dropped his 'Alabaster Snowball' badge. He mentions that he tried to analyze the ransomware but got locked out, and that the password database on his computer is encrypted. The second message is from Shinny Upatree at 5:31PM, where he congratulates Alabaster and shares information about a new ransomware called Wannacookie. Shinny mentions that several elves reported receiving a cookie recipe Word doc that, when opened, flashed a PowerShell screen and encrypted files. He also notes that many elves were affected, so Alabaster went to see if he could help. Shinny then shares a tip from an elf he follows online, stating that Wannacookie communicates over DNS and transfers files over DNS, and that it looks like it grabs a public key this way. He also mentions that a recent ransomware made it possible to retrieve crypto keys from memory, and that hopefully the same is true for Wannacookie. Finally, he suggests that there might be a flaw in the wannacookie author's DNS server that they can manipulate to retrieve what they need, and that if so, they can retrieve their keys from memory, decrypt the key, and then decrypt their ransomed files. Several lines in Shinny's message are circled in blue.

A **Alabaster Snowball** 3:00PM
Help, all of our computers have been encrypted by ransomware!
I came here to help but got locked in 'cause I dropped my "Alabaster Snowball" badge in a rush.
I started analyzing the ransomware on my host operating system, ran it by accident, and now my files are encrypted!
Unfortunately, the password database I keep on my computer was encrypted, so now I don't have access to any of our systems.
If only there were some way I could create some kind of traffic filter that could alert anytime ransomware was found!
Oh my! Santa's castle... it's under siege!
We're trapped inside and can't leave.
The toy soldiers are blocking all of the exits!
We are all prisoners!

S **Shinny Upatree** 5:31PM
Sweet candy goodness - I win! Thank you so much!
Have you heard that Kringle Castle was hit by a new ransomware called Wannacookie?
Several elves reported receiving a cookie recipe Word doc. When opened, a PowerShell screen flashed by and their files were encrypted.
Many elves were affected, so Alabaster went to go see if he could help out.
I hope Alabaster watched the PowerShell Malware talk at KringleCon before he tried analyzing Wannacookie on his computer.
An elf I follow online said he analyzed Wannacookie and that it communicates over DNS.
He also said that Wannacookie transfers files over DNS and that it looks like it grabs a public key this way.
Another recent ransomware made it possible to retrieve crypto keys from memory. Hopefully the same is true for Wannacookie!
Of course, this all depends how the key was encrypted and managed in memory. Proper public key encryption requires a private key to decrypt.
Perhaps there is a flaw in the wannacookie author's DNS server that we can manipulate to retrieve what we need.
If so, we can retrieve our keys from memory, decrypt the key, and then decrypt our ransomed files.

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.

```
Malware Reverse Engineering
From: Alabaster Snowball
-----
Whoa, Chris Davis' talk on PowerShell malware is
crazy pants! You should check it out!
```

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

INTRO:

Kringle Castle is currently under attacked by new piece of ransomware that is encrypting all the elves files. Your job is to configure snort to alert on ONLY the bad ransomware traffic.

GOAL:

Create a snort rule that will alert ONLY on bad ransomware traffic by adding it to snorts /etc/snort/rules/local.rules file. DNS traffic is constantly updated to snort.log.pcap

COMPLETION:

Successfully create a snort rule that matches ONLY bad DNS traffic and NOT legitimate user traffic and the system will notify you of your success.

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

```

elf@6f2d3dfb78de:~$ cat more_info.txt
MORE INFO:
  A full capture of DNS traffic for the last 30 seconds is
  constantly updated to:

  /home/elf/snort.log.pcap

  You can also 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

  This sensor also hosts an nginx web server to access the
  last 5 minutes worth of pcaps for offline analysis. These
  can be viewed by logging into:

  http://snortsensor1.kringlecastle.com/

  Using the credentials:
  -----
  Username | elf
  Password | onashelf

  tshark and tcpdump have also been provided on this sensor.

HINT:
  Malware authors often use dynamic domain names and
  IP addresses that change frequently within minutes or even
  seconds to make detecting and block malware more difficult.
  As such, it's a good idea to analyze traffic to find patterns
  and match upon these patterns instead of just IP/domains.elf@6f2d3dfb78de:~$

```

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 will be 03 www 06 google 03 com in the bytes pane. Is there anything consistent with those numbers?