

# Argus: Practical BotNet Detection

Written by: geek00L[20070418]

What is argus?

The network Audit Record Generation and Utilization System. The Argus Open Project is focused on developing network activity audit strategies that can do real work for the network architect, administrator and network user.

I'm lazy to explain myself hence I suggest you look at it at the link below if you want to know more about it -

<http://qosient.com/argus/>

There are many things that can be done if you are using argus as your tactical tool to collect flow/session data in your network security operation, here I will show you some interesting features of argus client tools (typically ragrep and radump) combining with the idea that taken from ourmon approach. When comes to botnet detection, using port based detection is generally good but not good enough. I will demonstrate the easy way of detecting the Botnet not only based on port itself but looking into the content payload at the same time.

I specify the filter with the destination port range greater or equal to 6660 but less or equal to 6675. However with this itself not enough to give you confirmation of whether the host has connected to the botnet, I have extended it by turning on the connection state -z. So what does this ragrep actually help, ragrep can perform regular expression matching and grep the matching flow, I build this regex '(JOIN|PRIVMSG|P[IO]NG)' since they are the IRC commands that usually in used. It generates the interesting result for me -

```
shell>ragrep -z -i -e '(JOIN|PRIVMSG|P[IO]NG)' -R * -s +ltime - dst port  
gte 6660 and lte 6675
```

```
18:30:28.234316    18:30:30.255023          tcp      1.2.3.4.33231    <?>  
195.197.175.21.6666    6          6          416          678          E  
21:29:34.513852    21:29:35.283850          tcp      1.2.3.4.33246    ->  
195.197.175.21.6669    5          5          389          3970          sSE  
18:30:28.234316    18:30:30.255023          tcp      1.2.3.4.33231    <?>  
195.197.175.21.6666    6          6          416          678          E  
21:29:34.513852    21:29:35.283850          tcp      1.2.3.4.33246    ->  
195.197.175.21.6669    5          5          389          3970          sSE
```

The destination port 6666 and 6669 are commonly used in irc botnet arsenal and we have just found the flows to those ports with the matching of the strings (JOIN, PRIVMSG, PING, PONG) as well. We confirm the successful connection setup within these two hosts when we spotted the connection states changes sSE and E which indicates tcp 3 ways handshake has completed and connection has established, further looking into the flow user data giving us full confidence to make decision, I try to filter the transaction of host 195.197.175.21 -

```
shell>radump -L0 -nnR * -s +suser:32 +duser:32 - host 195.197.175.21
```

```

23:54:05.177156          6      195.197.175.21.6669    <?>
    1.2.3.4.33246          2      1          167          100          CON
s[32]="PING :Helsinki.FI.EU.Und" d[12]="PONG"

23:54:06.187115          6      195.197.175.21.6666    <?>
    1.2.3.4.33231          2      1          167          100          CON
s[32]="PING :Helsinki.FI.EU.Und" d[12]="PONG"

23:58:34.192085          6      195.197.175.21.6669    <?>
    1.2.3.4.33246          1      1          122          66          CON
s[32]=":Shedulesk!~broadcast@61" d[0]=""

23:58:34.192185          6      195.197.175.21.6666    <?>
    1.2.3.4.33231          1      1          122          66          CON
s[32]=":Shedulesk!~broadcast@61" d[0]=""

23:59:28.197256          6      195.197.175.21.6669    <?>
    1.2.3.4.33246          1      1          115          66          CON
s[32]=":Germany!~Sod@61.73.147" d[0]=""

23:59:28.197361          6      195.197.175.21.6666    <?>
    1.2.3.4.33231          1      1          115          66          CON
s[32]=":Germany!~Sod@61.73.147" d[0]=""

```

Looking at the user data bytes s[] and d[], it tells you most of the things you need to know, again here I demonstrate the power of argus that is outstanding. In fact if you deploy argus correctly, you won't be worried if your history full content data collection is flushed(lack of capacity) since argus will still provide the data set that you need. And with the flow constructed fragmentation attack to bypass the detection of strings matching becomes trivial for attacker, therefore this is helpful to compliment you intrusion detection system as well.

I'm currently using argus 3.0 release candidate version, many people have thought that argus is not much into development but joining argus mailing list itself may give you joy as you may see that argus is actively developed and improved with the helps of the community. Feel free to try out argus 3.0 where you can download at -

<ftp://qosient.com/dev/argus-3.0>

There are many other argus client tools that I haven't shown the usabilities here, feel free to install and try out now.

Big thanks go to

- Carter Bullard(Argus father)
- Richard Bejtlich(Taosecurity)
- All the members in Argus mailing list and Freenode #snort-gui

#### Reference:

- <http://web.cecs.pdx.edu/~jrb/jrb.papers/sruti06/sruti06.pdf>
- [http://www.cert.org/flocon/2006/presentations/botserver2006\\_ppt.pdf](http://www.cert.org/flocon/2006/presentations/botserver2006_ppt.pdf)