

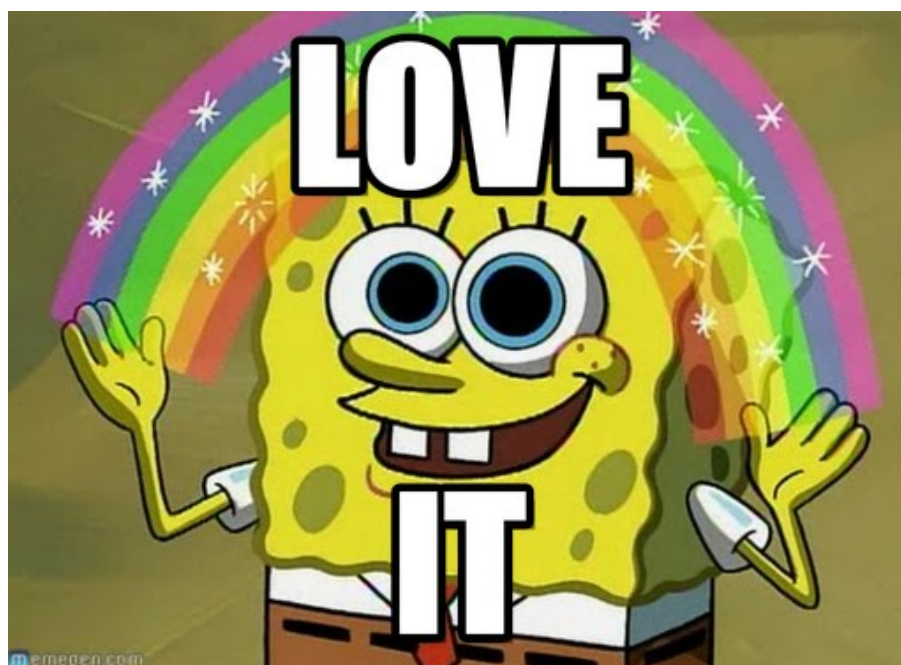
13
SEP
2017

HOW-TO, TECHNICAL BRO IDS, DNSCAT2, HOW TO, METERPRETER, POWERSHELL EMPIRE, RITA

Let's Go Hunting! How to Hunt Command & Control Channels Using Bro IDS and RITA

[Logan_Lembke](#)//

Here at BHIS, we ♥ Bro IDS.



Imagine... Bro IDS Everywhere!

If you haven't encountered Bro IDS before, checkout this [webcast](#) on [John's](#) Youtube channel discussing the need for Bro IDS and what it can offer your local blue team.

Readying Your Weapons: Installing Bro IDS



Bro IDS requires a UNIX like operating system such as Linux, Mac OS, or BSD.

Bro installations are generally tailored to their environment. As such, there are several ways to get started with Bro. The [official installation instructions](#) suggest compiling Bro from source. While this approach will provide you with extra goodies, a packaged binary will do just fine for offline packet capture analysis.

In order to install a packaged version of Bro IDS:

- Visit <https://www.bro.org/download/packages.html>
- Find and run the instructions for your operating system

- Install the

```
bro-aux
```

package as well

- Add

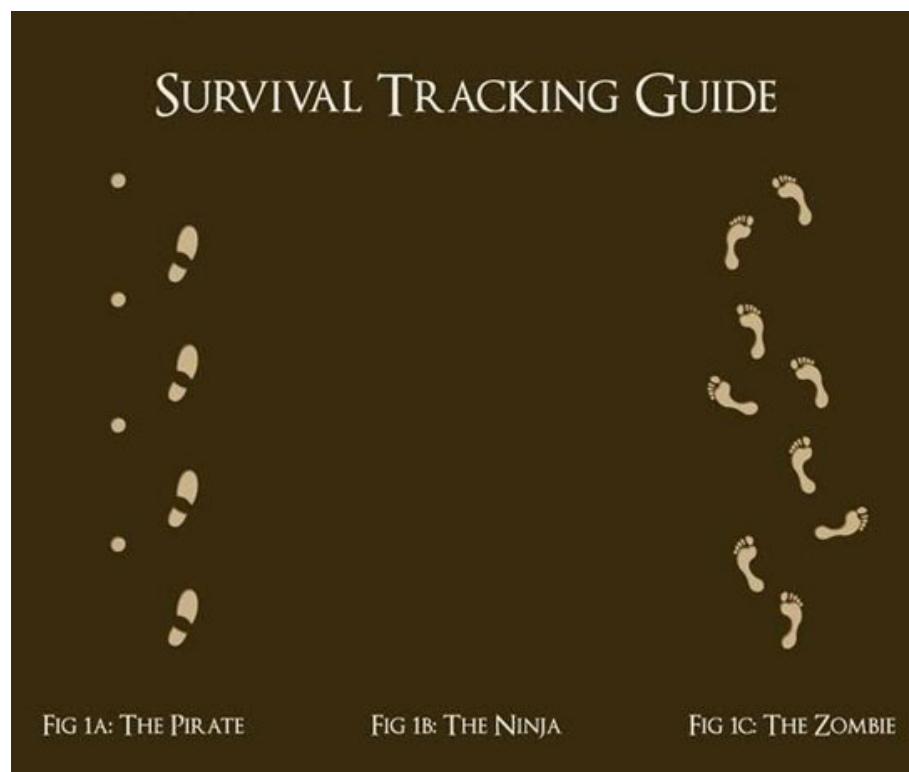
```
/opt/bro/bin
```

to your

\$PATH

Alternatively, I've put together an [installation_script](#) for Debian based systems which will compile Bro IDS from source with all of its optional dependencies.

Understanding the Tracks



We'll Even Catch the Ninja

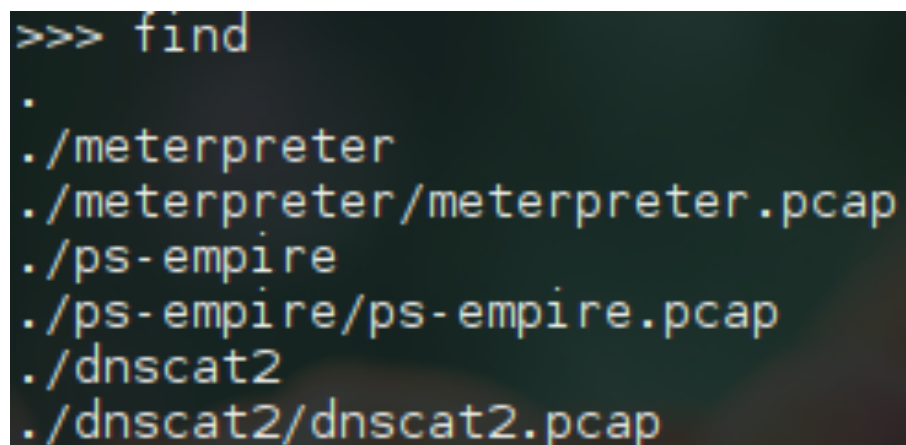
Bro IDS may be used to directly analyze a tapped network; however, Bro is also able to analyze raw pcap files. Included below are three sample packet captures. Each capture contains the traffic produced by an infected machine (10.200.201.29) communicating back to an attack server. Before continuing, download the following files:

- Dnscat2 (Command and control using DNS queries)
- Powershell Empire (Command and control using HTTPS connections)
- Meterpreter (Command and control using TCP connections)

After downloading each of the individual packet captures, open up a terminal, and move each file into its own directory. Bro IDS writes its analysis results out to the current working directory, and we don't want to confuse the results from the different packet captures. Finally, extract each file with

```
gunzip [sample.pcap.gz]
```

Running "find" from the top level directory should yield something similar to this:

A terminal window with a dark background and light-colored text. The command '>>> find' has been entered, and the output lists the contents of the current directory and its subdirectories: '.', './meterpreter', './meterpreter/meterpreter.pcap', './ps-empire', './ps-empire/ps-empire.pcap', './dnscat2', and './dnscat2/dnscat2.pcap'.

```
>>> find
.
./meterpreter
./meterpreter/meterpreter.pcap
./ps-empire
./ps-empire/ps-empire.pcap
./dnscat2
./dnscat2/dnscat2.pcap
```

Once the files are in their individual folders, we need to run Bro. In each of the individual folders, run

```
bro -C -r [sample.pcap] local "Site::local_nets += { 10.0.0.0/8
```

This will produce a number of logs in each directory. The 700

```
-C
```

flag tells Bro to ignore the packet checksums, the

```
-r
```

flag tells Bro to read a pcap file, and the rest lets Bro know that the 10.x.x.x/8 subnet is our local network.

In a real world scenario, Bro produces an extraordinarily large amount of data to sift through. While the [official documentation](#) is actively maintained, it is spread across multiple web pages. Alternatively, Critical Stack has put together [a helpful handout](#) explaining each of the logs.

Easy Game: Dnscat2 (DNS Tunneling C2)





The Original DNSCat Logo: Isn't He Cute?

Dnscat2 has been mentioned a couple of times before on the BHIS blog. We showed that the tool could [bypass Cylance](#), and Luke presented his [rewrite of the tool using Powershell](#). If you're unfamiliar with dnscat2, I encourage you to take a look at our earlier posts before continuing.

Conn.log

The connection log is the most important Bro log to review. Per the Bro IDS website, "[The connection log] manages the tracking/logging of general information regarding TCP, UDP, and ICMP traffic. For UDP and ICMP, "connections" are to be interpreted using flow semantics (sequence of packets from a source host/port to a destination host/port)."

These "flow semantics" catch dnscat2 red-handed.

Normally, when looking at a packet capture, UDP traffic is seen as a stream of individual datagrams sent across the network. However, Bro IDS groups these connections together as long as they happen at a reasonable rate over a unique socket pair. This means Bro IDS can easily point out long UDP "sessions."

In the conn.log produced by analyzing dnscat2.pcap, you should see the following line

```
1503528301.909886      CoPfoo4LI4g4NNUFOe      10.200.201.29      337
201.2      53      udp      dns      2467.745404      402129      639484
      0      Dd      4837      537565      4837      774920      (empty)
```

This line shows that our infected host, 10.200.201.29, issued thousands of consecutive dns queries over the period of 2,467.7 minutes (41 hours)! *Any long running connections should be immediately suspect, especially if they happen to be running over dns.*

Dns.log

The DNS log is one of the most helpful logs for identifying user behavior. While most traffic is secured by TLS and hidden from analysis, we can still find out which sites our individual hosts have connected to via their dns lookups.

The DNS log produced by the Dnscat2 is especially gnarly. I recommend using

```
less -S dns.log
```

in order to view the file. The

```
-S
```

option prevents word wrapping.

Upon opening the file, you will notice that all of the requests share a common “super” domain:

My command and control server is the authoritative name server for this domain. As such, any dns queries for a subdomain of

sirknightthe.chickenkiller.com

will be sent to it. The final subdomains are generated by the dnscat2 client in order to send data back to the C2 server. Since the dnscat2 client needs to encode all of its data in these subdomains, it needs to produce a large number of them. *In order to catch this DNS tunneling behavior, we need to keep a count of the subdomains we have seen for a given “super” domain. After gathering this data, we look for abnormally high ranking counts.*

However, there may be another way to catch Dnscat2.

By default, the Dnscat2 client sends out MX, CNAME, and TXT record queries. While CNAME queries will appear in almost every network environment, MX and TXT queries are somewhat rare. *An abnormal influx of MX, CNAME, or TXT records may indicate that a dns tunnel is operating on your network.*

Upping the Difficulty: Powershell Empire (Reverse HTTPS C2)





Powershell Empire is one of the most used post-exploitation tool kits available. In the sample linked above, a python based implant was ran on a Linux machine. This infected machine then called back to a Powershell Empire C2 server over HTTPS.

Conn.log

Unfortunately, Powershell Empire doesn't keep a single TCP session alive so we can't use the same long connection analysis we used earlier for dnscat2. Rather, it "beacons." After you open the connection log produced by the Powershell Empire capture, look at the recorded timestamps. If you look closely, you will see that the implant called back to the C2 server every 5 seconds. *Using frequency analysis, we can clearly spot this beaconing behavior. Alternatively, we can simply look for hosts which have made a large number of connections to a single external host over the course of a day.*

Unfortunately, this beaconing behavior is not so readily apparent in real world packet captures. Connections from other systems clutter up the connection log and it is difficult to check the timestamps directly. Beyond the "needle in the haystack" problem, "jitter" may be introduced to the connection. Jitter randomly adds delays

between the beacons, throwing off the “every 5 seconds” relation we had noticed before. However, advanced frequency analyses have been shown to detect beaconing behavior even in the presence of jitter.

Alternatively, look at the fields labelled

```
orig_bytes
```

and

```
resp_bytes
```

These are extremely regular. These fields measure how many bytes were sent to and from our infected host over each TCP connection. Unfortunately, these fields may slightly vary over the course of the infection. As a hacker pivots or exfiltrates data from a system, more or less data may be sent.

Ssl.log

While SSL and TLS secure most of our data, Bro IDS is able to get around this by harvesting unencrypted connection metadata and logging it to the SSL log.

In this capture, almost every connection was made over TLS. You can prove this to yourself by comparing the connection and SSL logs. In fact, you can relate the log entries using their second field,

```
uid
```

Bro analyzes each connection in several different ways and uses these UUIDs to relate the analysis results.

In the SSL log we see the same beaconing behavior; however, we see something more interesting. Each connection was encrypted with a self signed certificate. *By default, most hacking tools use self-signed certificates. This makes it easy to catch lazy hackers.*

If you're interested in learning more about the certificates used for each connection, look at the corresponding entries in the x509 and files logs.

Seeing Through the Camouflage: Meterpreter (Reverse TCP C2)



Even Camo is Digital Now

A Meterpreter connection can be established using either a reverse TCP transport or a reverse HTTP(S) transport, meaning Meterpreter has a few different ways to call back home. The HTTPS transport is similar to that of Powershell Empire. However, the TCP transport maintains an active TCP connection throughout the infection. In this capture, I elected to use the reverse TCP transport. I've purposefully left the Meterpreter packet capture dirty in hopes that you can sift through the data in order to find the infection.

Bro-Cut

Learning how to use tools like grep, cut, and awk makes this problem tractable. However, Bro IDS also includes a python tool called bro-cut. Similar to dnscat2, we are looking for long connections. Bro-cut allows us to throw away the fields we aren't interested in.

```
cat conn.log | bro-cut uid id.orig_h id.resp_h duration | sort  
d -n 5
```

will display the top 5 connections by duration. From here, we grab the UID of the top connection and grep out the full connection details. For me, the top connection is labeled

```
CFRuW5gJrBirOIYZ4
```

and I run

Your UID may be different.

After running a whois search on the destination, we see that the IP address is part of the Amazon EC2 cloud. While I conducted this capture in EC2, it should be clear that *long connections to cloud services should be immediately suspect*.

Grep, cut, awk, bro-cut, sort, head, tail, and the rest of the standard *nix utilities are essential for making use of the logs produced by Bro IDS in the real world.

Other Logs

The Meterpreter packet capture is a bit dirty. While this makes finding the discussed infection a bit harder, it demonstrates some of Bro's more advanced capabilities. The http log shows the results of upgrading a host's APT package manager and installing Elinks, a console based web browser. The software log boils this information down and tells us that 10.200.201.29 ran several versions of APT in addition to the ELinks web browser. Over time, the known_hosts log and known_services log can be used in conjunction with the software log in order to build up an inventory of a tapped network. Beyond these files, Bro IDS offers a multitude of [interesting monitoring capabilities](#) including full file captures, blacklist analyses, and more.

Hunting With Robots: RITA

Hunting through logs by hand takes time and practice. However, software has been developed to address this problem. Rather than stringing along a variety of *nix commands across a slew of terminals, we can use software to direct our search. Enter RITA, [Real Intelligence Threat Analytics](#).



Don't We All?

RITA reads logs produced by Bro IDS and extracts as many interesting features from the dataset as possible. RITA finds dnscat2 by spotting long lasting connections as well as by counting subdomains. Additionally, RITA has a special beaconing module which uses advanced techniques from frequency analysis in order to find beaconing hosts. Powershell Empire and other beaconing

software is easily spotted after running RITA. Meterpreter is no different. RITA is able to see through the camouflage and show you the target.

In order to get started with RITA, head over to [our project page](#), and install the program alongside John.

Alternatively, visit the [GitHub page](#) and follow the instructions listed there.

In short:

- Spin up an instance of Ubuntu 16.04 (or similar)

- Install git if it isn't installed

```
sudo apt install git
```

- Clone RITA

```
git clone https://github.com/ocmdev/rita.git
```

- Run the installer

```
chmod +x install.sh; sudo ./install.sh
```

- Source your .bashrc

```
source ~/.bashrc
```

- Move the Bro logs from earlier to your RITA system if they are not there
- Start MongoDB

```
sudo systemctl start mongod
```

- In the top level directory containing the three sample folders, run

```
rita import -i [meterpreter folder] -d Meterpreter
```

```
rita import -i [ps-empire folder] -d Powershell-Empire
```

```
rita import -i [dnscat2 folder] -d DNSCat2
```

- Analyze the ingested data

```
rita analyze
```

- Create the report

```
rita html-report
```

- Finally, open the file

```
rita-html-report/index.html
```

in a web browser

You should see the following display:

To view individual databases, click on any of the links below.

Meterpreter

Powershell-Empire

DNSScat2

To begin with, open up Meterpreter, and click on long connections at the top.

OFFENSIVE CounterMeasures	RITA	Viewing: Meterpreter	Beacons	Blacklisted	DNS	Scans	Long Connections	
Long URLs	User Agents							RITA on
Source	Source Port	Destination	Destination Port	Duration	Protocol			
10.200.201.29	51896	18.220.61.97	80	37042.611743	tcp			
10.200.201.29	51886	18.220.61.97	80	930.75256	tcp			
10.200.201.29	51892	18.220.61.97	80	252.549776	tcp			

Here you will see the results we found earlier with bro-cut. Next, open up Powershell-Empire and click on the beacons tab.

OFFENSIVE CounterMeasures											
RITA		Viewing: Powershell-Empire		Beacons	Blacklisted	DNS	Scans	Long Connections			
Long URLs		User Agents									RITA on
TS score	Source	Destination	Connections	Avg. Bytes	Intvl. Range	Intvl. Mode	Intvl. Mode Count	Intvl. Skew	Intvl. Dispersion	TS Duration	
1.000	10.200.201.29	18.220.208.40	652	3809.113	1	5	639	0.000	0	1.000	

RITA clearly shows the beacon. The field “TS score” stands for the timestamp score, meaning that based on the connection timestamps, there was a perfect beacon from 10.200.201.29 to 18.220.208.40. This beacon occurred most frequently at 5 second intervals, and beaconsed 652 times. Finally, go to the dnscat2 results and click on DNS.

OFFENSIVE CounterMeasures	RITA	Viewing: DNSScat2	Beacons	Blacklisted	DNS	Scans	Long Connections	
Long URLs	User Agents							RITA on
Subdomain	Visited	Domain						
4852	4837	com						
4851	4837	chickenkiller.com						
4850	4837	sirknightthe.chickenkiller.com						

Here, we can see that there were 4,850 subdomains of

sirknightthe.chickenkiller.com

that were queried. In addition, if you go to the long connections tab, you will see the same results from earlier.

Currently, RITA does not alert on anything. Rather it is to be used as an assistant — a tactical tool. In addition to the analyses we discussed earlier, RITA performs blacklist checks, network scan detection, long URL analysis, and analysis of user agent strings. Going forward, RITA will be a test bed for a multitude of other analyses, replacing the searches we normally do by hand.

Hacking Leaves Tracks. Now We Can Follow Them.

Hackers may try to cover their tracks, but inevitably Bro IDS will record their movements. The beauty of Bro IDS is that it just needs a network tap. It can run on an entirely separate network. Unless a hacker gains physical control of the system, they will not defeat Bro. However, Bro is not a cure all. It simply produces too much data. Any signs of hacking are mixed and muddled with everyday traffic.

Bro IDS is able to produce terabytes of data. Yet, in order to extract value out of it, we need to either invest hundreds of man hours in manual analysis or in automation. Our hope is that RITA will solve this problem and aid you in your hunts to come.

Dropbox Links:

- Dnscat2 (Command and control using DNS queries)
 - <https://www.dropbox.com/s/gdlnfwj6qeiz2cf/ps->

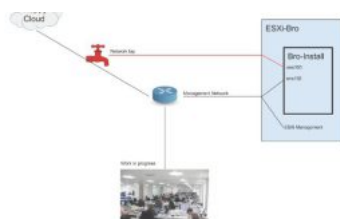
[empire.pcap.gz?dl=0](https://www.dropbox.com/s/gdlnfwj6qeiz2cf/ps-empire.pcap.gz?dl=0)

- Powershell Empire (Command and control using HTTPS connections)
 - <https://www.dropbox.com/s/gdlnfwj6qeiz2cf/ps-empire.pcap.gz?dl=0>
- Meterpreter (Command and control using TCP connections)
 - <https://www.dropbox.com/s/izmku9nqysjiwq0/dnscat2.dl=0>

Share this:



Related



How to Monitor Network Traffic with Virtualized Bro 2.51 on Ubuntu 16.04.2 on ESXi 6.5



WEBCAST: RITA

February 27, 2017
In "tool"

August 3, 2017
In "How-To"

```
mirrors.kernel.org/archlinux/iso/2016.02.01/
: of /archlinux/iso/2016.02.01/

-2016.02.01-dual.iso          01-Feb-2016 15:36 7
-2016.02.01-dual.iso.sig      01-Feb-2016 16:00 :
-2016.02.01-dual.iso.torrent  01-Feb-2016 16:00 +
bootstrap-2016.02.01-i686.tar.gz 01-Feb-2016 16:10 1
bootstrap-2016.02.01-i686.tar.gz.sig 01-Feb-2016 16:10 :
bootstrap-2016.02.01-x86_64.tar.gz 01-Feb-2016 16:12 1
bootstrap-2016.02.01-x86_64.tar.gz.sig 01-Feb-2016 16:12 :
i5                             01-Feb-2016 16:12 :
txt                             01-Feb-2016 16:12 :
```

Check Your Image

February 24, 2016
In "Industry"

< WEBCAST- WWHF Lab Exploration: Hands-on RF Attacks

LINKS



LOOKING FOR SOMETHING?

SUBSCRIBE TO THE BHISBLOG

Don't get left in the dark! Enter your email address and every time a post goes live you'll get instant notification!



New to Information
Security?

START HERE

Our non technical
articles & posts

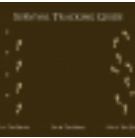
READ

some of our most

POPULAR

technical posts

RECENT POSTS



Let's Go Hunting! How to Hunt Command & Control Channels Using Bro IDS and RITA

Logan Lembke// Here at BHIS, we ♥ Bro IDS.



WEBCAST- WWHF Lab Exploration: Hands-on RF Attacks

David Fletcher// Join David as he takes a look at one



Upcoming Event in Arizona

John Strand// Hello all in Arizona! Just wanted to let

BROWSE BY CATEGORY

Event

Glossary of Terms

How-To

Industry

Informational

Interview

News

Non-Technical

Reference

Technical

tool

Webcasts

BROWSE BY TOPIC

2FA anti-virus AV Blue Team bypassing AV C2 Cylance dnscat2 encryption
hacking infosec Kill your AV Linux macros MailSniper meterpreter Microsoft MS Word
Nessus Nmap Outlook OWA passwords password spraying pen-testing
penetration testing pentest Pentesting phishing
PowerShell PowerShell Empire privacy Purple Team Red Team red teaming social
engineering steganography tool tools Ubuntu VM VPN Vulnerabilities webcast
Windows

ARCHIVES

Archives

Select Month

▼



BLACK HILLS INFORMATION SECURITY

115 W. Hudson St. Spearfish, SD 57783 | 701-484-BHIS

© 2017

LINKS



SEARCH THE SITE