



User Guide

Blue Team Training Toolkit (BT3)

26th of June 2016

Version	Date	Comment
1.0	26/06/2016	Initial version.

Juan J. Güelfo
Lead IT-security consultant at Encrypto AS

Table of Contents

1. Introduction.....	3
1.1 What is Blue Team Training Toolkit?.....	4
1.2 Who Should Use Blue Team Training Toolkit?.....	5
1.3 System Requirements.....	5
1.4 Disclaimer.....	6
1.5 License.....	6
2. Getting Started with Blue Team Training Toolkit.....	7
2.1 Download and Installation.....	7
2.2 Directory Structure.....	8
2.3 Interactive Command-Line Interface.....	9
3. BT3 Module: Maligno.....	10
3.1 Getting Started.....	11
3.2 Malware Profiles.....	13
3.3 Setting up Maligno.....	15
4. BT3 Module: Pcapteller.....	17
4.1 Getting Started.....	18
4.2 Setting up Pcapteller.....	20
4.3 Creating a Network Diversion.....	21
5. Known Bugs and Limitations.....	23

1. Introduction

Until the past decade, common threats against computer systems could be stopped by anti-virus software and firewalls. Nowadays, these two countermeasures can be easily bypassed by attackers, and they just offer a basic degree of protection. Moreover, IT personnel are required to have specialized skills within computer network defense analysis and incident response in order to detect, analyze and react effectively to computer threats.

Computer network defense analysis is a broad topic and skills can be acquired with different methods. Common training techniques are based on studying network traffic that could be either live or previously captured.

In any of these situations, the production and acquisition of network traffic requires an attack scenario with supporting infrastructure. The goal is to successfully monitor the network traffic while the attack is in progress. The result allows a blue team to improve their skills, test the detection tools deployed as part of an organization's IT infrastructure, and ultimately exercise their incident response plan.

Currently, the possibilities for training and improving in these disciplines have important constraints mainly related to these criteria:

- **Difficulty of implementation**
This criterion describes how difficult it is to create, configure and maintain an environment where the attack scenario is going to be executed. The difficulty of implementation is usually related to the amount of time required for the tasks. An ideal environment would involve low-time and low-work requirements.
- **Cost**
This criterion defines the amount of resources required for the correct implementation of the attack scenario. The lower the cost is, the smaller amount of money an organization will need to invest on its training program. Alternatively, low costs will allow organizations to design more complete training programs with the same budget.
- **Risk**
This criterion describes the danger that a production network faces when an attack scenario is executed during a training session. Risk can be understood as the combination of likelihood and impact associated to an event. Therefore, the lower the risk is, the safer the training environment will be.
- **Realism**
This criterion describes the level of detail that a training environment replicates based on what a real case would be. The higher the realism is, the closer to reality the training environment will be.

Typically, the criteria described in the previous sections tend to present themselves with important dilemmas, which force organizations to prioritize one criterion over others, or just reach a compromise that falls far from an optimal training session.

Let's illustrate such dilemmas with two common examples:

- **Efficiency versus Realism**
Network traffic produced in attack scenarios (purposed for training sessions) can be captured and saved as PCAP files. From a training perspective, such files contain a "story" specific to the environment where it was captured, and it can be used again by a blue team, for example when training new members or reviewing a training exercise.

This reusability may not be optimal when multiple organizations cooperate and exchange network traffic, in an attempt to conduct more efficient training sessions. Using network traffic produced by

external parties removes the creation of new attack scenarios from the equation. This reduces the cost and the preparation of a training session. However, it usually translates into less realism, since the use of network traffic produced in external networks will not match the organization's environment.

An ideal situation would allow organizations to cooperate, exchange network traffic and customize it to their needs. This would reduce costs and difficulty of implementation, while increasing network traffic reusability and realism.

- **Risk versus Realism**

In order to train computer network defense analysts and reach an advanced skill level, it is essential to create realistic attack scenarios that can generate relevant network traffic. In many cases, real pieces of malware are used in such scenarios, so computer network defense analysts can train with real indicators. However, this practice comes with an inherent risk.

On one hand, an attempt to reduce risk usually results in less realistic training sessions (e.g. not training in production environments). On the other hand, realistic scenarios tend to elevate risk. An optimal scenario should allow organizations to train in safe conditions, while keeping a high degree of realism.

1.1 What is Blue Team Training Toolkit?

Blue Team Training Toolkit (BT3) is an attempt to introduce improvements in current computer network defense analysis training. Based on adversary replication techniques, and with reusability in mind, BT3 allows individuals and organizations to create realistic computer attack scenarios, while reducing infrastructure costs, implementation time and risk.

The Blue Team Training Toolkit is written in Python, and it follows an open source FreeBSD license.

The most important features of BT3 include:

- **Adversary replication and malware simulation**

BT3 includes the latest version of Encripto's Maligno. This module is designed with a client-server architecture, and it allows blue teams to simulate malware infections or targeted attacks with specific C&C communications in a safe manner.

BT3 is also shipped with multiple malware communication profiles that ensure a "plug & play" experience, when planning and preparing a training session. Furthermore, malware profiles can be developed easily, something that contributes to lower preparation costs and better cooperation.

- **Network traffic manipulation and replay**

BT3 includes Encripto's Pcapteller, a module designed for traffic manipulation and replay. Pcapteller can customize and replay network traffic stored in PCAP files. This allows blue teams not only to re-create scenarios where computer attacks or malware infections occurred, but also make it look like everything is really happening in their own network.

- **Ease of use and flat learning curve**

Information security tools usually implement their own options, syntax and commands. Mastering a tool can therefore take some time.

To ensure usability from the first moment, and not waste lots of valuable time, BT3 uses an interactive command-line interface inspired by Rapid7's Metasploit Framework (MSF). Since MSF is a tool well-known by information security professionals, it makes sense to provide some degree of familiarity. This means that learning how to use BT3 should take a minimum effort, and most blue teams will be able to focus on their training session, rather than figuring out how to use a new tool.

- **Blue team cooperation and network traffic reusability**

On one hand, BT3 can contribute with flexible malware communication profiles that can be exchanged or distributed among organizations. Also, it helps blue teams train with a high degree of realism, without the need of using real malware. This is a key area that solves the “Risk versus Realism” dilemma.

On the other hand, BT3 offers a platform that improves efficiency, by reducing preparation time and infrastructure costs. The ability to customize captured network traffic can allow organizations to reuse and exchange PCAP files, while keeping a decent degree of realism. This reusability also ensures a better Return On Investment, since the network traffic of a training session can be customized and reused without setting up the whole original attack scenario. This addresses the “Efficiency versus Realism” dilemma.

Despite BT3 aims for blue teams, it could also become a powerful resource for red teams. In such context, BT3 module could assist with the creation of a decoy or a diversion during an engagement.

Let’s consider advanced security assessments that result in access to the target’s internal network. Such access could be obtained in multiple ways, for example by using social engineering against employees, compromising weak internet-facing systems, or just as starting point if the engagement assumes compromise.

In environments with tight network countermeasures and a (proactive) blue team in place, red teams must measure their movements across the target network, in order to fly under the radar.

Occasionally, red teams may perform actions in the network that could draw a blue team’s attention. Using BT3 in combination with VPN pivoting, red teams could create a network diversion. In other words, this could make a blue team see ghosts, letting a red team hide in plain sight.

1.2 Who Should Use Blue Team Training Toolkit?

Blue Team Training Toolkit has been designed for computer network defense analysis training, and it could be used by public and private organizations, as well as training institutions such as universities.

In addition, BT3 could assist red teams during specific scenarios that may occur during the course of a security engagement.

1.3 System Requirements

BT3 requires the following minimum hardware configuration:

- +500 Mhz processor.
- 1 GB RAM available.
- 100 MB available disk space.
- 10/100 Mbps network interface.

The following operating systems are officially supported by BT3:

- Kali Linux x64, with Python 2.7.
- Security Onion 14.04, with Python 2.7.
- Ubuntu 14.04 LTS / 16.04 LTS, with Python 2.7.

BT3 has been successfully tested on physical hosts and virtual machines (VirtualBox 5.0).

The software should also run on other Debian-based distributions. However, no further testing has been done so far.

BT3 depends on “python2.7”, “python-scapy” and “python-ipcalc” packages. It also uses OpenSSL for generating a server certificate during the installation process. The BT3 installer will take care of these dependencies automatically. Given the nature of the functionality implemented in BT3, the software must run with root or sudo privileges.

Clients generated by BT3’s Maligno module have been successfully tested on Windows and Linux hosts. Clients can be executed as regular Python scripts, or compiled with PyInstaller 2.x / 3.x. Successful script execution or PyInstaller compilation will require Python 2.7. No elevated privileges are required in order to run Maligno client scripts.

1.4 Disclaimer

Blue Team Training Toolkit (BT3) can only be used for legal activities.
Use this software at **your own risk**.

It is the user's responsibility to obey all applicable laws.
The developer or Encripto AS assume no liability, and are not responsible for any misuse or damage caused by this program.

Any of the trademarks, service marks, collective marks, design rights, personality rights or similar rights that are mentioned, used or cited in this document is property of their respective owners.

Read the license section in this document for more details.

1.5 License

Blue Team Training Toolkit (BT3) is licensed under the FreeBSD license.
Read <http://www.freebsd.org/copyright/freebsd-license.html> for more details.

Blue Team Training Toolkit (BT3). Written by Juan J. Güelfo.
Copyright 2013-2016 Encripto AS. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY ENCRIPTO AS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

IN NO EVENT SHALL ENCRIPTO AS, THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of Encripto AS.

2. Getting Started with Blue Team Training Toolkit

This section is going to cover the most fundamental aspects of Blue Team Training Toolkit (BT3) that will get you started in no time.

2.1 Download and Installation

BT3 is distributed as a tarball file, and it can be downloaded from <https://www.encrypto.no/tools>. Once the file is on your hard disk, proceed to extract it and run the installer as shown below:

```
root@kali:~# tar -xvzf BT3-1.0.tar.gz
BT3-1.0/
BT3-1.0/modules/
BT3-1.0/modules/maligno_client.py
BT3-1.0/modules/__init__.py
BT3-1.0/modules/menu.py
BT3-1.0/modules/pcapteller.py
BT3-1.0/modules/maligno.py
BT3-1.0/BT3.py
```

Fig. 1: Terminal output during tarball extraction

```
root@kali:~# cd BT3-1.0/
root@kali:~/BT3-1.0# ./install.sh
```

Fig. 2: Running the BT3 installer

```
=====
| Blue Team Training Toolkit - Install Script |
| by Juan J. Guelfo, Encripto AS (post@encrypto.no) |
=====

[*] Installing dependencies...
Hit:1 http://nl.mirror.babylon.network/kali kali-rolling InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

Fig. 3: Installer progress

During the installation process, the installer will proceed to create a self-signed server certificate that can be used with BT3's Maligno module. The certificate generation process will require some information. At this point of the installation, you will have the opportunity to use default values by pressing "Enter", or providing your own. Be aware default values could trigger IDS signatures under certain circumstances.

```
[*] Creating folders..
[+] Directory 'pcaps' successfully created.
[+] Directory 'certs' successfully created.

[*] Generating server key and certificate..

Generating a 2048 bit RSA private key
..+++
.....+++
writing new private key to 'server.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:

[*] Generating PEM...
[+] Certificate successfully generated.

[+] Installation completed!
```

Fig. 4: Self-signed certificate generation completes the installation

As soon as the certificate is generated, the installer will place it in the “certs” folder. You may add extra certificates (PEM format) to this folder for later use, if desired.

At this point, the installation process should be complete.

2.2 Directory Structure

After successfully running the installation process, your Blue Team Training Toolkit installation folder should contain a few relevant directories:

- **certs**
This folder contains SSL/TLS certificates that can be used with BT3’s Maligno module. Additional certificates (PEM format) can be placed in this directory before the module is run. After the installation process, the folder should contain a self-signed certificate ready for use.
- **pcaps**
This directory contains PCAP files (libpcap format) containing captured network traffic, which can be used with BT3’s Pcapteller module. New PCAP files must be placed in this folder before the module is run. This folder will be empty right after completing the installation process. This means that the user will have to add new PCAP files in order to successfully run the Pcapteller module.
- **profiles**
This folder contains malware profiles that can be used with BT3’s Maligno module. BT3 is shipped with multiple profiles which are ready for use. New profiles can be added to this folder before running the Maligno module.

2.3 Interactive Command-Line Interface

Blue Team Training Toolkit offers an interactive command-line interface with syntax completion. This section will cover the most relevant commands supported by the application.

- You may start the interface by running “python BT3.py” from your Linux terminal.

```

root@kali:~/BT3-1.0# python BT3.py

      Blue Team
      Training Toolkit

~~~~~

      Blue Team Training Toolkit (BT3) v1.0
      By Juan J. Guelfo | Encripto AS | post@encripto.no

~~~~~

BT3 > █
    
```

Fig. 5: Running Blue Team Training Toolkit

- A quick command overview can be obtained with the “help” command.

```

BT3 > help

  Command      Description
  -----
  back         Exit current selected module and return to main menu.
  bt3update    Check for software updates.
  exit        Exit the Blue Team Training Toolkit.
  help        Display help menu.
  show modules Display supported application modules.
  use <module> Select an application module.
  version     Display software version.

BT3 > █
    
```

Fig. 6: Help menu displaying general commands

- The application’s current version can be displayed with “version”, while “bt3update” will check for new updates.

```

BT3 > version

[*] You are running Blue Team Training Toolkit (BT3) version 1.0
    
```

Fig. 7: Results of the “version” command

```
BT3 > bt3update

[*] Checking for updates...
[+] There is a new version of the Blue Team Training Toolkit!
[+] Check https://www.encripto.no/tools for more information.

BT3 > █
```

Fig. 8: Blue Team Training Toolkit can check for new updates on demand

- Supported application modules can be displayed with “show modules”.

```
BT3 > show modules

Module      Description
-----
maligno     Attack simulation with customized malware indicators.
pcapteller  Network traffic manipulation and replay.

BT3 > █
```

Fig. 9: List of tools (modules) contained in BT3

3. BT3 Module: Maligno

Maligno is a module designed for attack simulations that require risk-free / fictive malware infections, or targeted attacks with specific C&C communications. The module follows a client-server architecture, where the server component is hosted by the same computer where BT3 is running, and the client component can be deployed on different machines if desired.

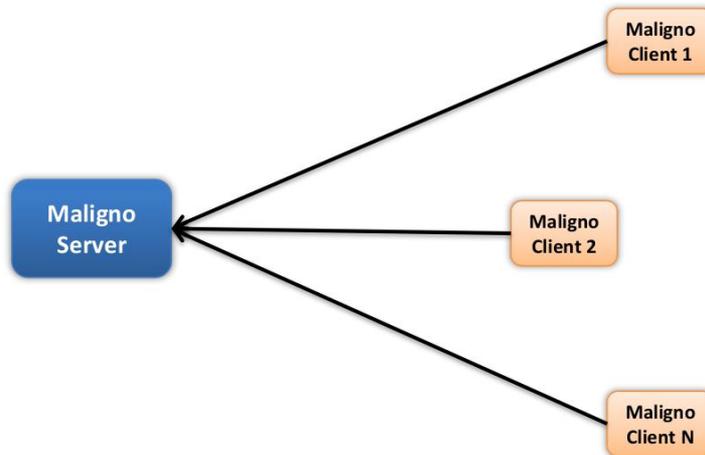


Fig. 10: Maligno clients can be distributed among multiple machines

Currently, Maligno server is integrated in the Blue Team Training Toolkit, and it runs on any of the supported operating systems covered in the system requirements section. However, Maligno clients can run on any operating system (e.g. Microsoft Windows, or Linux) as long as Python 2.7 is installed. Maligno clients can also run on Windows when compiled with PyInstaller. At the moment, client-server communications are handled via HTTP or HTTPS, since these are two of the most popular protocols used by malware these days.



Fig. 11: Maligno module components communicate over HTTP or HTTPS

Maligno clients are proxy aware, and they can handle themselves in multiple environments. Different proxy capabilities have been implemented in Maligno clients so far. These capabilities depend on what operating system a Maligno client is running on. The table listed below summarizes what connection scenarios are possible on different client platforms.

Blue Team Training Toolkit - Maligno Client			
Platform	Proxy Auth.	WPAD Auth.	Connectivity
Windows	Unauthenticated	Unauthenticated	Successful
	Basic	Basic	
	NTLM	NTLM	
*nix / OS X	Unauthenticated	Unauthenticated	Successful
	Basic	Basic	

3.1 Getting Started

The module can be invoked with “use maligno” directly from the BT3 command-line interface. You should note that the BT3 command prompt changes based on the current module in use.

```
BT3 > use maligno
BT3 ~ maligno > |
```

Fig. 12: Maligno module ready for use after invocation

- The current module version can be checked with the “version” command.

```
BT3 ~ maligno > version
[*] You are running Maligno version 3.0
BT3 ~ maligno > |
```

Fig. 13: Maligno version command output

- Maligno supports a range of general commands, which can be displayed with “help”.

```
BT3 ~ maligno > help

Command          Description
-----
back             Exit current selected module and return to main menu.
exit             Exit the Blue Team Training Toolkit.
genclient       Generate a client with the current configured settings.
help            Display help menu.
run             Run the module with the given options.
set <option> <value> Set module option.
show options    Display module options.
show profiles   Display malware indicator profiles.
version        Display module version.

BT3 ~ maligno > |
```

Fig. 14: List of commands supported by the module

- Module options and their current values can be listed with “show options”.

```
BT3 ~ maligno > show options
Name          Setting      Required    Description
-----
LHOST         80           True        IP address or FQDN to expose the C2 server on.
LPORT         standard     True        TCP Port to listen for connections.
PROFILE       False        True        Profile containing malware network indicators.
SSL           False        False       Enable server SSL/TLS support.
SSL_CERT      server.pem   False       Server certificate to use with SSL/TLS support.

BT3 ~ maligno > █
```

Fig. 15: Module options and their current values

Maligno Module Options	
Name	Description
LHOST	<p>Defines the IP address or Fully-Qualified Domain Name (FQDN) where the Maligno server component will be exposed.</p> <p>This value is actively used by the Maligno client generation process. Maligno clients will attempt connections to the IP address or FQDN provided by this option.</p>
LPORT	<p>Defines the TCP port to listen for incoming connections.</p> <p>This value is actively used by the Maligno client generation process. Maligno clients will attempt connections to the port provided by this option.</p>
PROFILE	<p>Defines the name of the Maligno malware profile in use. Valid profiles can be listed with “show profiles”.</p>
SSL	<p>Defines whether the Maligno server component will support SSL/TLS for incoming connections.</p>
SSL_CERT	<p>Defines the server certificate in use when SSL/TLS support is enabled. The self-signed certificate generated during the installation process is used by default.</p> <p>Additional certificates can be used, as long as they are placed in the “certs” directory, within the BT3’s installation folder.</p>

- Module option values can be set with the “set” command, the desired option and its new value.

```
BT3 ~ maligno > set LHOST 192.168.1.10
[+] LHOST => 192.168.1.10
BT3 ~ maligno > █
```

Fig. 16: Setting a new option value

- Available malware profiles can be listed with “show profiles”.

```
BT3 ~ maligno > show profiles

Profile      Description
-----
cookie       Default profile with cookie header and random elements.
cryptowall_3 Cryptowall v3 ransomware profile.
etumbot      Etumbot APT backdoor profile.
ghostnet_asp Ghostnet APT profile based on ASP technology.
ghostnet_php Ghostnet APT profile based on PHP technology.
harnig       Harnig trojan downloader profile.
havex        Havex trojan profile.
nuclear      Nuclear exploit kit file delivery profile.
oldrea       Oldrea APT backdoor profile.
putterpanda  Putter Panda APT profile.
standard     Default profile without random elements.
standard_random Default profile with random elements.
upatre       Upatre banking trojan profile.
zemot        Zemot trojan profile.

BT3 ~ maligno > █
```

Fig. 17: Maligno malware profiles ready for use

- Once all required module options have been configured with valid values, it will be possible to generate a Maligno client script. Maligno clients can be generated directly from the BT3 command-line interface, with the “genclient” command. The generated client script will be placed at the “clients” folder, and it will be ready for deployment.

```
BT3 ~ maligno > genclient

[*] Generating Maligno client...
[+] Maligno client successfully generated! Check the "clients" folder.

BT3 ~ maligno > █
```

Fig. 18: Successful Maligno client generation

```
root@kali:~/BT3-1.0/clients# ls -l
total 28
-rw-r--r-- 1 root root 26558 Jun 23 20:23 maligno_client_standard.py
```

Fig. 19: Generated clients are placed in a specific location

- Maligno server can be started with the “run” command. All module options are validated during this process.

```
BT3 ~ maligno > run

[*] Maligno is up and running. Press [CTRL+C] to stop...
```

Fig. 20: Maligno server is running and waiting for connections

3.2 Malware Profiles

Maligno’s malware profiles are programmed in Python, and they follow an intuitive structure. The profiles are very flexible, and they can be customized either with simple modifications or with very complex functionality.

Malware profiles are located in the “profiles” directory, which can be found within the Blue Team Training Toolkit’s installation folder. The profiles are divided in well structured areas:

Maligno Malware Profile Structure		
Class	Purpose	Value Visibility
Info	Gathers general information about the Maligno malware profile.	BT3’s command-line interface.
Request	Defines the indicators that Maligno clients will use when sending requests.	Network traffic.
Response	Defines the indicators that Maligno server will use when responding to client requests.	Network traffic.
Network	Defines protocol specific configurations, as well as communication parameters.	Network traffic.

The table listed below explains the purpose of each class attribute:

Maligno Malware Profile Attributes		
Class	Attribute	Purpose
Info	author	Defines who made the profile.
	description	Provides a summary of what kind of communications or malware indicators the profile attempts to simulate. This field will be visible on the BT3’s “show profile” command output.
	license	Describes the license model applied to the profile.
	references	Includes links to threat intelligence reports or other materials that backup the behavior represented by the profile.
Request	method	HTTP request method to use by client requests. Possible values are “GET”, “POST” or “HEAD”.
	URI	Defines the URI portion of HTTP client requests. The attribute is a list. When several comma-separated values are provided, Maligno clients will pick a URI randomly for each request.
	body	Defines the body portion of HTTP client requests. The body should be used with POST requests. However, BT3 will not complain if a body is provided with other request methods (even if the requests are malformed).
	headers	Defines the HTTP headers included in HTTP client requests. Maligno clients will attempt to honor the header order. The attribute is a list of comma-separated dictionaries.

Response	code	Defines the HTTP response code (type of response) sent by the server.
	banner	Defines the web server banner disclosed by the server, which is included as a response header.
	body	Defines the response body. This is the actual data sent in the response.
	headers	Defines the HTTP headers included in HTTP server responses. Maligno server will attempt to honor the header order. However, this is not guaranteed. The attribute is a list of comma-separated dictionaries.
Network	protocol	Defines the type of HTTP protocol to be used in client requests. Possible values are "HTTP/1.0" or "HTTP/1.1".
	encoding	<p>Defines the type of encoding to apply to the response body. This will give the server response a different look on the wire. Possible values are "None", "Base64", "Hex" or "Bin".</p> <p>Please, note that the encoding applies to the whole response body. If you would like to encode just specific parts of the response body, you should use "None" as encoding, and implement your own encoding logic within the profile's functionality. Check the modules shipped in BT3 for implementation examples.</p>
	delay	<p>Defines the amounts of seconds that Maligno client will wait before sending a request. The value is a non-negative value (greater or equal to zero).</p> <p>Note that a delay of "0" seconds will generate a huge amount of requests in a short period of time.</p>
	jitter	<p>Defines a random deviation that will be added to the delay time. The value is understood as percentage of the delay time.</p> <p>For example, a delay time of 10 seconds and a jitter of 50% will result in a maximum waiting time of 15 seconds.</p>

3.3 Setting up Maligno

This section will illustrate how to setup up BT3's Maligno with a practical example. In this case, Maligno will be used during the simulation of a targeted attack. A piece of malware known as "Havex" or "Oldrea" has been actively used against western energy companies in the past.

Symantec has documented several cases in a report that describes network indicators associated to Havex. BT3 includes a Maligno malware profile based on such report, and it will mimic the malware's network behavior without risking any infection.

Before starting the actual setup, this case will assume that a blue team has already deployed some minimal infrastructure for network traffic monitoring. In addition, Snort with ET GPL rule set will be used as Intrusion Detection System.

- **Step 1: Configure the module options**

In this case, the “oldrea” profile should be configured as well as the server’s IP address. Communications will go over HTTP and they will use the standard port TCP 80 (default).

```
BT3 > use maligno
BT3 ~ maligno > set PROFILE oldrea

[+] PROFILE => oldrea

BT3 ~ maligno > set LHOST 192.168.1.10

[+] LHOST => 192.168.1.10

BT3 ~ maligno > show options

  Name          Setting      Required  Description
  ----          -
  LHOST         192.168.1.10 True       IP address or FQDN to expose the C2 server on.
  LPORT         80           True       TCP Port to listen for connections.
  PROFILE       oldrea       True       Profile containing malware network indicators.
  SSL           False        False      Enable server SSL/TLS support.
  SSL_CERT      server.pem   False      Server certificate to use with SSL/TLS support.

BT3 ~ maligno > █
```

Fig. 21: Module options after configuration

- **Step 2: Generate and deploy your Maligno client script**

A Maligno client script should be successfully generated once the module has been configured. Client scripts should be then deployed on those hosts that will simulate the infection or should be considered as compromised.

```
BT3 ~ maligno > genclient

[*] Generating Maligno client...
[+] Maligno client successfully generated! Check the "clients" folder.

BT3 ~ maligno > █
```

Fig. 22: Successful Maligno client generation

- **Step 3: Start the server and run the client**

The Maligno server component can be started directly from BT3’s interactive interface. The Maligno client, on the other hand, should be invoked from the machines where the scripts were deployed.

```
BT3 ~ maligno > run

[*] Maligno is up and running. Press [CTRL+C] to stop...

=====

[*] New request from 192.168.1.11...

192.168.1.11 - - [24/Jun/2016 17:04:14] "POST /wp08/wp-includes/dtc
la.php?d=285745296322896178920098FD80-20&v1=038&v2=170393861&q=5265
882854508EFCF958F979E4 HTTP/1.1" 200 -

[+] Request served!
[*] End of request.

=====
```

Fig. 23: Maligno server running and receiving a client request during the course of the exercise

```

=====
Blue Team Training Toolkit (BT3)
Maligno module v3.0

By Juan J. Guelfo | Encripto AS | post@encripto.no
=====

[*] Maligno client module is running. Press [CTRL+C] to stop...

[*] Preparing request #5...
[*] Sending request via direct connection...
[+] Request sent...
[*] Sleeping 10s...
    
```

Fig. 24: Maligno client output during execution

• **Step 4: Traffic analysis**

The network communications should present patterns based on the malware indicators configured in the profile. Network equipment as well as your packet captures should register the activity at this point.

Src IP	SPort	Dst IP	DPort	Pr	Event Message
192.168.1.10	80	192.168.1.11	42327	6	ET TROJAN Havex RAT CnC Server Response HTML Tag

Fig. 25: Snort IDS alert triggered by the network activity

```

POST /wp08/wp-includes/dtcla.php?d=285745296322896178920098FD80-206v1=0386v2=170393861&q=5265882854508EFC958F979E4 HTTP/1.1
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US) AppleWebKit/525.19 (KHTML, like Gecko) Chrome/1.0.154.36
Safari/525.19
Host: toons.freesexycomics.com
Content-Length: 0
Cache-Control: no-cache

HTTP/1.1 200 OK
Server: Apache/1.3.37 (Unix)
Date: Fri, 24 Jun 2016 15:16:32 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive
Cache-Control: no-cache
Content-length: 1231

9f65-html<<head><meta http-equiv='CACHE-CONTROL' content='NO-CACHE'></head><body>No data!<!--havexQlpo0TFBWSZTW
WVYDIB0sD//////////4oB+93V V Xu69DuN7XyZds9y
t49QuesVXhmVLdDxpVSXZ3QWbBERja0twTFdJV3NGV0RuUmSLY01Ya2N1Y3lCbLVLaHZxcVdzTLRPNcKwkhCYVWwZnZUVZaUtDRHRJV0NpY0N0S2lwanLsS
25myWLDUNsT0ZjWFB4BHliduL3Y3RSwXVjSHN5bVRXUFVpUXNRUGLEcm5LbIppaktIRZNTS25semJYVHNQTUFjd1JB2YVmqmthY0dI20Z3cFBWUULFYWJ3VHLWe
E5NbVZORGNmeVnpSxw1BEFaTEVLbw9VUER3dFl0bFZaV1lIWkZLVLD01lCbKvqUp3ZGRGQmh4ZVd3c2lUeUZEUndXUW16VW55EJ3V11BRHRIZG52UHRUOKt0U
3d0SEh2VnVWVwT1lESk9ZundhexZV0mLQZw1Q09wa3JWU3Za0khlCU1YaXlqTxp0Yud6bmxoWHRVd2RS cmxHd1lGRXpYREpTQ1NSUK1aQXluand4U2tGck9Cd
WFTZEEdScnJnZut2cW9jQ1NjC0d1bmZiCwNTU3BDeLpLWgdhR3JW09pMcWlPTVZIUfVncINhQ01FU1BseLROskhYb0lNenJsyYXJ0nZaFBjbxVtbfNhcKpzRgtVU
ZZRVHhsY0pTVGNjblBkQVBNblJlVmtQUTYRFNqZ2NzWlVtcEFFVxG5IphdVNBdFh1V0lLRnG0U2zaX0WgdrZUXckRhRGN0dEF1V3lVtkHZlFtYXdxRU5Kc
GR0RBKTLdxdmRRGhCwkhieHBSRlV6T25zeUxU0FkeGV6R2lReLFEemptY0hSSERJUFN1SmsFVEZvVmxCcFNBDULmR0lnewJYeEt6Q3NodExwkbpNR3Vgck1LR
KLHSFZ5cU9Z5GpRS1F0RWRudFRJT1B3VUV0U0NtS1hYZEXIR2tC+yUW3zFTXWA0stsCwCckdW5
AHS06vbbCu7GputPt5CSfgPCAkxCA00ICmsqLiACGYEhAQ3v9eDM92D/8XckU4UJBlWYNA==havex--></body></head>
    
```

Fig. 26: One of the HTTP requests captured during the course of the exercise (UTC time zone)

4. BT3 Module: Pcpatteller

Pcpatteller is a module designed for network traffic manipulation and replay. It allows organizations to re-create a recorded network traffic scenario that occurred in a foreign network, as it really happened in their own infrastructure.

In a nutshell, Pcpatteller reads network packets from a PCAP file, and replays them into the network. The module allows packet manipulation (MAC and IP addresses at this moment) prior to replay, so it is possible to customize the traffic with specific addresses that fit your environment.

The module is useful if you want to re-create scenarios where computer attacks or malware infections occurred. Using such scenarios as a base, Pcpatteller will allow you to reuse existing PCAP files and make everything look like the attack is really happening in your own network. Pcpatteller can help you improving your blue team's network security monitoring skills, or creating network diversions during red team operations.

4.1 Getting Started

The module can be invoked with “use pcapteller” directly from the BT3 command-line interface. You should note that the BT3 command prompt changes based on the current module in use.

```
BT3 > use pcapteller
BT3 ~ pcapteller > |
```

Fig. 27: Pcapteller module ready for use after invocation

- The current module version can be checked with the “version” command.

```
BT3 ~ pcapteller > version
[*] You are running Pcapteller version 1.2
BT3 ~ pcapteller >
```

Fig. 28: Pcapteller version command output

- Pcapteller supports a range of general commands, which can be displayed with “help”.

```
BT3 ~ pcapteller > help

Command          Description
-----
back             Exit current selected module and return to main menu.
exit            Exit the Blue Team Training Toolkit.
help           Display help menu.
run            Run the module with the given options.
set <option> <value> Set module option.
show options   Display module options.
version        Display module version.

BT3 ~ pcapteller >
```

Fig. 29: List of commands supported by the module

- Module options and their current values can be listed with “show options”.

```
BT3 ~ pcapteller > show options

Name          Setting  Required  Description
-----
FILE          True     True      Pcap file to replay in libpcap format.
FRAGMENTATION False    True      Fragment packets during replay. Useful for networks with low MTU.
INTERFACE     True     True      Network interface to replay the packets with.
MTU           1500    True      MTU to use with packet fragmentation.
PCAP_IP_LIST  False   False     Comma-separated list of IP addresses to replace as seen on the pcap file.
PCAP_MAC_LIST False   False     Comma-separated list of MAC addresses to replace as seen on the pcap file.
REAL_TIME     False   False     Honor inter-arrival delays while replaying packets.
WIRE_IP_LIST  False   False     Comma-separated list of IP addresses to replay as seen on the wire.
WIRE_MAC_LIST False   False     Comma-separated list of MAC addresses to replay as seen on the wire.

BT3 ~ pcapteller > |
```

Fig. 30: Module options and their current values

Pcatteller Module Options	
Name	Description
FILE	Defines the PCAP file (libpcap format) to manipulate and replay. The file must be located in the “pcaps” directory, within the BT3’s installation folder.
FRAGMENTATION	Defines whether packet fragmentation should be enabled. This option is useful in cases where the PCAP file includes packets larger than the target network’s MTU.
INTERFACE	Defines the local network interface to use in order to replay the packets.
MTU	Defines the MTU size when packet fragmentation is in use. The MTU size must be an integer between 1 and 9000 bytes.
PCAP_IP_LIST	Defines a comma-separated list of IP addresses to replace. The IP addresses must be defined as seen in the PCAP file. The IP addresses defined in the PCAP_IP_LIST will be replaced by the same element index of the WIRE_IP_LIST.
PCAP_MAC_LIST	Defines a comma-separated list of MAC addresses to replace. The MAC addresses must be defined as seen in the PCAP file. The MAC addresses defined in the PCAP_MAC_LIST will be replaced by the same element index of the WIRE_MAC_LIST.
REAL_TIME	Defines whether packet inter-arrival timing should be honored. When this option is enabled, Pcatteller will not inject the network packets at once. Instead, it will honor the time elapsed between the arrival of each packet contained in the PCAP file. This provides a very realistic timing when analyzing a chain of events produced by the simulation.
WIRE_IP_LIST	Defines a comma-separated list of IP addresses to inject during the traffic manipulation phase. Such addresses will be visible on the wire during the traffic replay. The IP addresses defined in the PCAP_IP_LIST will be replaced by the same element index of the WIRE_IP_LIST.
WIRE_MAC_LIST	Defines a comma-separated list of MAC addresses to inject during the traffic manipulation phase. Such addresses will be visible on the wire during the traffic replay. The MAC addresses defined in the PCAP_MAC_LIST will be replaced by the same element index of the WIRE_MAC_LIST.

- Module option values can be set with the “set” command, the desired option and its new value.

```
BT3 ~ pcatteller > set INTERFACE eth0
[+] INTERFACE => eth0
BT3 ~ pcatteller > █
```

Fig. 31: Setting a new option value

- Once all required module options have been configured with valid values, Pcatteller can begin to replay packets with the “run” command. All module options are validated prior to execution.

```
BT3 ~ pcaprunner > run

[*] Pcaprunner started at 00:50:12. Press [CTRL+C] to stop.

[*] Reading "/root/BT3-1.0/pcaps/test.pcap"...
[+] 783 packet(s) found.

[*] Processing 783 of 783 packet(s) | Error: 0 packet(s).
[*] Replaying packet(s) via eth0...
[+] Replay complete.

[*] Pcaprunner finished at 00:50:13.

BT3 ~ pcaprunner >
```

Fig. 32: Successful packet replay with Pcaprunner

4.2 Setting up Pcaprunner

This section is going to demonstrate how BT3’s Pcaprunner module can be used during a simple training session. This case will use a public PCAP file that contains an attack scenario involving an exploit kit delivering ransomware. This PCAP file describes a chain of events where host “192.168.122.70” is the victim.

Source	Destination	Protocol	Info
192.168.122.70	144.76.161.38	TCP	49203 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
144.76.161.38	192.168.122.70	TCP	http > 49203 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1367 SACK_PERM=1
192.168.122.70	144.76.161.38	TCP	49203 > http [ACK] Seq=1 Ack=1 Win=65616 Len=0
192.168.122.70	144.76.161.38	HTTP	GET /indexing_raspberries_rejuvenation_sushis/415213137352185210 HTTP/1.1
144.76.161.38	192.168.122.70	TCP	http > 49203 [ACK] Seq=1 Ack=621 Win=15872 Len=0
144.76.161.38	192.168.122.70	TCP	[TCP segment of a reassembled PDU]
192.168.122.70	144.76.161.38	TCP	49203 > http [ACK] Seq=621 Ack=1368 Win=65616 Len=0

Fig. 33: Fragment of the original PCAP file with an attacker IP address and the victim (192.168.122.70)

Let’s consider an organization that would like to use such resource for a training session. The organization is interested in using its current security countermeasures and configurations in production. The production network is using a class B internal IPv4 addressing schema (172.31.0.0/16). For this example, the victim machine will become “172.31.10.11”.

In this case, the following module options should be configured:

```
BT3 ~ pcaprunner > set PCAP_IP_LIST 192.168.122.70
[+] PCAP_IP_LIST => 192.168.122.70

BT3 ~ pcaprunner > set WIRE_IP_LIST 172.31.10.11
[+] WIRE_IP_LIST => 172.31.10.11

BT3 ~ pcaprunner > show options
Name          Setting      Required  Description
----          -
FILE          test.pcap    True      Pcap file to replay in libpcap format.
FRAGMENTATION False        True      Fragment packets during replay. Useful for networks with low MTU.
INTERFACE     eth0         True      Network interface to replay the packets with.
MTU           1500        True      MTU to use with packet fragmentation.
PCAP_IP_LIST  192.168.122.70 False     Comma-separated list of IP addresses to replace as seen on the pcap file.
PCAP_MAC_LIST          False     False     Comma-separated list of MAC addresses to replace as seen on the pcap file.
REAL TIME     False        False     Honor inter-arrival delays while replaying packets.
WIRE_IP_LIST  172.31.10.11 False     Comma-separated list of IP addresses to replay as seen on the wire.
WIRE_MAC_LIST          False     False     Comma-separated list of MAC addresses to replay as seen on the wire.

BT3 ~ pcaprunner >
```

Fig. 34: Module options prior to traffic manipulation and replay

The result of the customized traffic injected into the network is described in the screenshots below.

```
BT3 ~ pcaprunner > run

[*] Pcaprunner started at 19:55:59. Press [CTRL+C] to stop.

[*] Reading "/root/BT3-1.0/pcaps/test.pcap"...
[+] 783 packet(s) found.

[*] Processing 783 of 783 packet(s) | Error: 0 packet(s).
[*] Replaying packet(s) via eth0...
[+] Replay complete.

[*] Pcaprunner finished at 19:56:00.

BT3 ~ pcaprunner > █
```

Fig. 35: Running BT3's Pcaprunner module

Source	Destination	Protocol	Info
172.31.10.11	144.76.161.38	TCP	49203 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
144.76.161.38	172.31.10.11	TCP	http > 49203 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1367 SACK_PERM=1 WS=128
172.31.10.11	144.76.161.38	TCP	49203 > http [ACK] Seq=1 Ack=1 Win=65616 Len=0
172.31.10.11	144.76.161.38	HTTP	GET /indexing_raspberries_rejuvenation_sushis/415213137352185210 HTTP/1.1
144.76.161.38	172.31.10.11	TCP	http > 49203 [ACK] Seq=1 Ack=621 Win=15872 Len=0
144.76.161.38	172.31.10.11	TCP	[TCP segment of a reassembled PDU]
172.31.10.11	144.76.161.38	TCP	49203 > http [ACK] Seq=621 Ack=1368 Win=65616 Len=0

Fig. 36: Fragment of the manipulated PCAP file with attacker IP address and the victim (172.31.10.11)

Since Pcaprunner injects the manipulated network traffic into the production network, existing security countermeasures can detect and alert about possible threats. This example shows how an Intrusion Detection System (Snort with ET GPL rule set) would react to the manipulated traffic.

Src IP	SPort	Dst IP	DPort	Pr	Event Message
172.31.10.11	49203	144.76.161.38	80	6	ET POLICY Outdated Windows Flash Version IE
172.31.10.11	49203	144.76.161.38	80	6	ET CURRENT_EVENTS Possible Angler EK Flash Exploit URI Structure Jan 21 2015
144.76.161.38	80	172.31.10.11	49205	6	ET CURRENT_EVENTS Angler EK XTEA encrypted binary (11) M2
144.76.161.38	80	172.31.10.11	49205	6	ET CURRENT_EVENTS Angler EK XTEA encrypted binary (13)
172.31.10.11	49206	54.93.182.214	80	6	ET POLICY Possible External IP Lookup ipinfo.io
172.31.10.11	49207	104.27.143.176	80	6	ET TROJAN Win32/Teslacrypt Ransomware HTTP CnC Beacon M2
172.31.10.11	62658	8.8.4.4	53	17	ET TROJAN TeslaCrypt/AlphaCrypt Variant .onion Proxy Domain (lq3ahljcleont3xx)
172.31.10.11	60626	8.8.4.4	53	17	ET POLICY DNS Query to .onion proxy Domain (tor2web)
192.251.226.206	443	172.31.10.11	49218	6	ET CURRENT_EVENTS Tor2Web .onion Proxy Service SSL Cert (1)

Fig. 37: Alerts generated by Intrusion Detection System (Snort) during the execution of the example

4.3 Creating a Network Diversion

In environments with tight network countermeasures and a (proactive) blue team in place, a red team must measure their movements across the target network, in order to fly under the radar. But, what if this is not possible? What if the red team needs to perform actions that could potentially draw the blue team’s attention?

Using BT3’s Pcaprunner module in combination with VPN pivoting, a red team could create a network diversion. In other words, this could make a blue team see ghosts through packet captures and/or deployed Intrusion Detection Systems. Here you have an example on how this works in practice:

- **Step 1: Assumptions**

Let’s assume that the red team has already deployed a VPN tunnel towards the target network. The red team has also some basic target network visibility. In other words, they know about MAC addresses or the IP address schema of the target network.

For the sake of this explanation, the target network will be “192.168.1.0/24”, with a Palo Alto appliance (MAC address “00:1b:17:00:00:02”) as gateway. The target network is also running Snort as Intrusion Detection System.

The red team has also a PCAP file containing the chain of events and the network indicators related to an exploit kit attack with a successful ransomware infection.

Alternatively, network traffic with custom indicators could be generated and captured with other tools, such as BT3’s Maligno module and Wireshark.

- **Step 2: Preparing your ghosts**

Based on information gathered during the engagement, the red team should pick a set of MAC addresses that fits the target environment. The same applies to internal IP addresses that may be used as decoys, in an attempt to draw the blue team’s attention.

In this specific example, the PCAP file was downloaded from <http://malware-traffic-analysis.net>, and it shows host “10.3.23.102” as victim. The MAC address of the gateway used by such host is “20:e5:2a:b6:93:f1”.

No.	Time	Source	Destination	Protocol	Len
1	15:42:17.722	10.3.23.102	23.229.205.72	TCP	
2	15:42:17.844	23.229.205.72	10.3.23.102	TCP	
3	15:42:17.844	10.3.23.102	23.229.205.72	TCP	
4	15:42:17.845	10.3.23.102	23.229.205.72	HTTP	
5	15:42:17.845	23.229.205.72	10.3.23.102	TCP	

▶ Frame 3: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)					
▼ Ethernet II, Src: HewlettP_1c:47:ae (00:08:02:1c:47:ae), Dst: Netgear_b6:93:f1 (20:e5:2a:b6:93:f1)					
▶ Destination: Netgear_b6:93:f1 (20:e5:2a:b6:93:f1)					
▶ Source: HewlettP_1c:47:ae (00:08:02:1c:47:ae)					
Type: IPv4 (0x0800)					
Padding: 000000000000					
▶ Internet Protocol Version 4, Src: 10.3.23.102, Dst: 23.229.205.72					
▶ Transmission Control Protocol, Src Port: 49309 (49309), Dst Port: 80 (80), Seq: 1, Ack: 1, Len: 0					

Fig. 38: Fragment of the original contents of the PCAP file

- **Step 3: Sending traffic**

In order to deploy a realistic decoy that can drive network countermeasures crazy, and hopefully confuse the blue team, the red team will manipulate and replay traffic with BT3’s Pcapteller module over the existing VPN tunnel.

In this case, the original host under attack will be replaced with “192.168.1.111” (a random host in the target network), and the original gateway’s MAC address will be replaced with the Palo Alto appliance’s “00:1b:17:00:00:02”. All manipulated traffic will be replayed over the VPN tunnel interface “vpn0”. With such decisions made, Pcapteller can be configured like this:

```
BT3 - pcapteller > show options
-----
Name          Setting          Required  Description
-----
FILE          decoy.pcap       True      Pcap file to replay in libpcap format.
FRAGMENTATION False            True      Fragment packets during replay. Useful for networks with low MTU.
INTERFACE     vpn0              True      Network interface to replay the packets with.
MTU           1500             True      MTU to use with packet fragmentation.
PCAP_IP_LIST  10.3.23.102      False     Comma-separated list of IP addresses to replace as seen on the pcap file.
PCAP_MAC_LIST 20:e5:2a:b6:93:f1 False     Comma-separated list of MAC addresses to replace as seen on the pcap file.
REAL_TIME     False            False     Honor inter-arrival delays while replaying packets.
WIRE_IP_LIST  192.168.1.111   False     Comma-separated list of IP addresses to replay as seen on the wire.
WIRE_MAC_LIST 00:1b:17:00:00:02 False     Comma-separated list of MAC addresses to replay as seen on the wire.

BT3 - pcapteller >
```

Fig. 39: Module options prior to traffic manipulation

For even a more realistic look, “REAL_TIME” support could be enabled on Pcapteller. This would honor inter-arrival packet delays during the actual replay.

- **Step 4: Results**

Once the network traffic is replayed over the VPN tunnel, the countermeasures placed on the target network should register the “fake activity”.

Encrypto AS – Blue Team Training Toolkit

Src IP	SPort	Dst IP	DPort	Pr	Event Message
192.168.1.111	49331	64.20.35.186	80	6	ET TROJAN Suspicious Accept in HTTP POST - Possible Alphacrypt/TeslaCrypt
192.168.1.111	49331	64.20.35.186	80	6	ET TROJAN Alphacrypt/TeslaCrypt Ransomware CnC Beacon
103.27.87.88	80	192.168.1.111	49333	6	ET TROJAN Alphacrypt/TeslaCrypt Ransomware CnC Beacon Response

Fig. 40: Snort alerts triggered by the network diversion

Even if the blue team goes into a packet level, Wireshark will display the replayed traffic as if the infection really happened. The traffic should reflect the manipulation of both MAC and IP addresses.

Source	Destination	Protocol	Length	Info
192.168.1.111	64.20.35.186	TCP	66	49331 > 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
64.20.35.186	192.168.1.111	TCP	60	80 > 49331 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
192.168.1.111	64.20.35.186	TCP	60	49331 > 80 [ACK] Seq=1 Ack=1 Win=16445440 Len=0
192.168.1.111	64.20.35.186	HTTP	922	POST /sysstr.php HTTP/1.1 (application/x-www-form-urlencoded)
64.20.35.186	192.168.1.111	TCP	60	80 > 49331 [ACK] Seq=1 Ack=869 Win=64240 Len=0

Fig. 41: Fragment of the replayed traffic (network decoy)

▶ Destination: PaloAlto_00:00:02 (00:1b:17:00:00:02)

Fig. 42: Destination MAC address after replay

5. Known Bugs and Limitations

Blue Team Training Toolkit is in constant development and bugs could always happen. The following lines gathers known bugs and limitations.

- BT3's Maligno profiles with "Transfer-Encoding" header set to "chunked" are not handled properly. The value is deliberately sent as "chuncked" as a workaround.
- BT3's Maligno client HTTP(S) proxy awareness works with static proxies and WPAD when executed on Windows and non-Windows platforms. Supported authentication methods are anonymous, basic and NTLM (NTLM only on Windows).

WPAD is not a standard implementation. It just detects all possible proxies in the PAC and uses the first one that allows internet access. This implementation ensures internet connectivity also under some non-standard proxy configurations.

- BT3's PcapTeller may replay DNS response packets in an inconsistent or corrupted manner under certain situations. The issue seems to be caused by Scapy, and it is under investigation.
- Encrypto AS does not provide free support for this tool.

Feel free to contact post@encrypto.no for feedback, bug reports or feature requests.