User Guide

Blue Team Training Toolkit (BT3)

3 September 2018

Version	Date	Comments
2.8	03/09/2018	Changes introduced by BT3 v2.8 have been included. New examples illustrating Maligno usage with proxy servers have been included.

Juan J. Güelfo
Lead IT security consultant at Encripto AS

Table of Contents

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 network analysis and incident response in order to detect, analyze and react effectively to computer threats.

Network analysis and incident response is a broad topic, and skills can be learned 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 with 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 three 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.

- **Risk versus Cost**
  Running low-risk training sessions tends to increase costs, because more resources and preparation are required. Assuming a training session is going to be conducted in a production network, organizations will typically try to reduce risk as much as possible. Two common scenarios can represent the dilemma.

  On one hand, if real malware samples are used, reverse engineering or other research against the sample should be conducted. This will provide the organization with clear guidelines of how to work with the sample, and what to expect if something goes wrong. Reverse engineering requires extra preparation time and knowledge, which is usually translated into higher costs. If the organization does not want to spend such amount of resources, it should be prepared to accept a higher risk during the training session.

  On the other hand, organizations could use specialized commercial software for malware simulation and/or an external Red Team. While this alternative tends to be a safe approach, it rapidly increases the costs of the training session. Companies with significant resources and mature security programs are usually the ones who can benefit from this approach, rather than organizations with constraints.

1.1 What is Blue Team Training Toolkit?

Blue Team Training Toolkit (BT3) is designed for network analysis training sessions, incident response drills and red team engagements. 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 you to simulate malware infections or targeted attacks with specific C&C communications in a safe manner.

  BT3 is also shipped with multiple malware indicator profiles that ensure a "plug & play" experience, when planning and preparing a training session, incident response drill or red team engagement. Furthermore, malware indicator 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 you not only to re-create scenarios where computer attacks or malware infections occurred, but also make it look like everything is really happening in your own network.

- **Malware sample simulation**
  BT3 includes Encripto's Mocksum, which provides access to a collection of files that mimic malware samples via MD5 hash collisions. The files downloaded via Mocksum allow you to simulate and plant realistic artifacts, without the risk of handling real malware. This is useful during training sessions, incident response drills and red team engagements.

  In a nutshell, these artifacts are harmless files that produce the same MD5 checksum as real malicious files. In many cases, the harmless artifacts also get detected by anti-virus software.

- **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, 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 indicator 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" and the "Risk versus Cost" dilemmas.

  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 allows 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.

- **Content subscription (optional)**
  The Blue Team Training Toolkit has API powers. By creating a free content subscription account, you get access to training content ready for use. It includes realistic network traffic related to a wide range of network attacks, mock malware samples with hash collisions, as well as important malware indicator profiles. Get the training content you need, right at your fingertips!

  A BT3 content subscription user account provides access to both free and premium content. Premium content can be downloaded by using pre-paid credits directly from the BT3 command line interface. It follows a Personal or Enterprise license. By purchasing content credits, you get the most out of your cyber security training sessions, incident response drills and red team engagements.

  Premium content can be downloaded by using pre-paid credits directly from the BT3 command line interface (more details are covered later). It follows a Personal or Enterprise license . By purchasing content credits, you can get the most out of your cyber security training sessions, incident response drills and red team engagements.

  Content subscription is an optional feature in the Blue Team Training Toolkit. This means that BT3 can still be used in offline mode if desired, with the same experience as in version 1.x.

Despite BT3 aims for blue teams, it is also a powerful resource for red teams. In such context, BT3 modules can assist with the production of network indicators, or decoys during a red team 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. By using BT3 in combination with VPN pivoting, red teams can create a network diversion. In other words, they can make a blue team see ghosts, letting their red team hide in plain sight.

1.2 Who Should Use Blue Team Training Toolkit?

Blue Team Training Toolkit is designed for network analysis training sessions, incident response drills and red team engagements. It could be used by public and private organizations, as well as training institutions such as universities.

1.3 System Requirements

Blue Team Training Toolkit requires the following minimum hardware configuration:

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

Access to online material provided by a Blue Team Training Toolkit content subscription has the following minimum requirements:

- 256 kbps internet connection.
- An active subscription bound to Personal or Enterprise license. Downloading premium content is optional, and requires pre-paid credits available in your account.

The following operating systems are officially supported by Blue Team Training Toolkit:

- Kali Linux x64, with Python 2.7.
- Ubuntu 16.04 LTS / Ubuntu 18.04 LTS, with Python 2.7.

Blue Team Training Toolkit requires **Python 2.7.9** or newer. Python 3.x is not supported at the moment.

BT3 has been successfully tested on physical hosts and virtual machines (VirtualBox 5.x). 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", "python-six" 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 Blue Team Training Toolkit License

1.6 Blue Team Training Toolkit Content Subscription Terms and Conditions

Before you start using Blue Team Training Toolkit content subscription, you need to read carefully and accept the terms and conditions listed at https://www.bt3.no/terms-conditions/

1.7 Blue Team Training Toolkit Content Subscription Privacy Policy

Before you start using Blue Team Training Toolkit content subscription, you need to read carefully and agree with the privacy policy listed at https://www.bt3.no/privacy-policy/

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

Blue Team Training Toolkit is distributed as a tarball file, and the latest version can be downloaded from https://www.encripto.no/tools. Once the file is on your hard disk, proceed to extract it and run the installer as shown below.

Please, note that the screenshots are illustrative. Make sure you type folder and file names correctly (according to your downloaded BT3 tarball and final deployment folder), as the folders and file names shown by the screenshots may not fit your environment.

Fig. 1: Fragment of a terminal output during tarball extraction

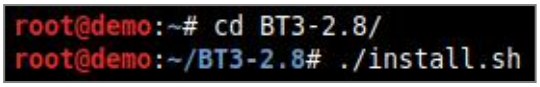

Fig. 2: Running the BT3 installer

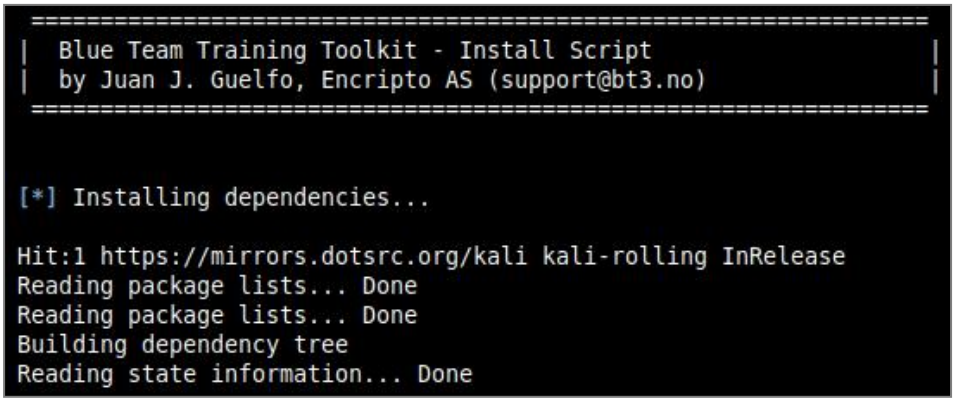

Fig. 3: Installer progress

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

Your Blue Team Training Toolkit installation folder should contain a few relevant directories. These will be created automatically by the installation process, or when the software is started for the first time:

- **certs**
  This folder contains SSL/TLS certificates that can be used with BT3's Maligno module. Additional certificates (in 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.

- **mockfiles**
  This directory contains mock malware samples downloaded via BT3's mocksum module. Mock files from this folder will be ready for deployment during your training session or security engagement.

- **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 or download new PCAP files in order to successfully run the Pcapteller module.

- **profiles**
  This folder contains malware indicator 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 or downloaded 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.

- **Starting Blue Team Training Toolkit**
  You may start the interface by running "python BT3.py" from your Linux terminal, with root or sudo privileges.

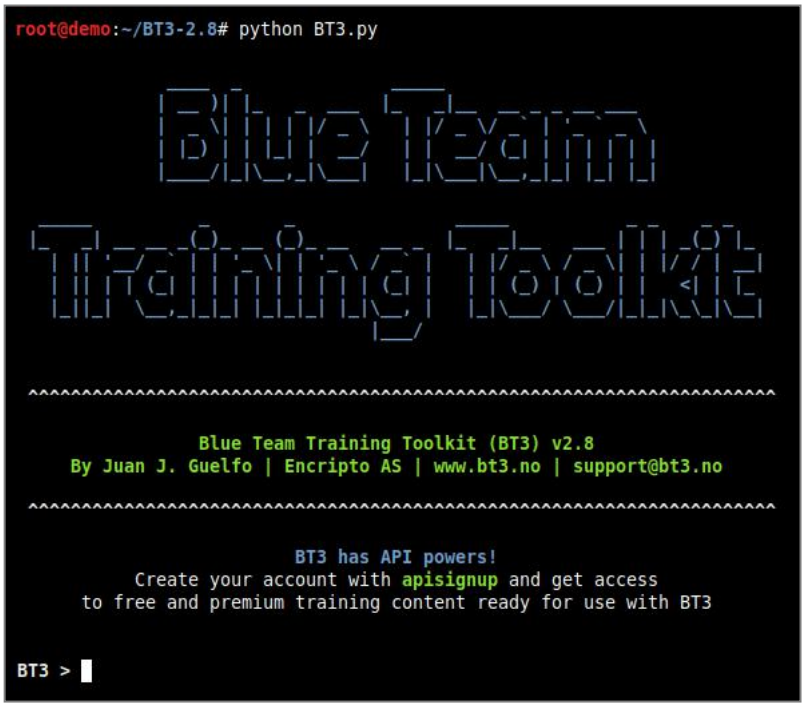

Fig. 5: Running Blue Team Training Toolkit

- **Help overview**
  A quick command overview can be obtained with the "help" command.

```
BT3 > help

    Command             Description
    -------             -----------
    apiconnect          Connect to Blue Team Training Toolkit API with valid credentials.
    apidelete           Delete your Blue Team Training Toolkit content subscription account.
    apidisconnect       Disconnect from Blue Team Training Toolkit API, and work in offline mode.
    apinewcreds         Start a Blue Team Training Toolkit API password change or account recovery.
    apiredeem <code>    Redeem a Blue Team Training Toolkit credit voucher.
    apisignup           Create a new Blue Team Training Toolkit content subscription account.
    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.
    resource <file>     Run a sequence of Blue Team Training Toolkit commands from a resource file.
    show modules        Display supported application modules.
    show subscription   Display Blue Team Training Toolkit content subscription details.
    use <module>        Select an application module.
    version             Display software version.

BT3 >
```

Fig. 6: Help menu displaying general commands

- **Resource files**
  Blue Team Training Toolkit supports resource files, which allow you to script module commands in a simple manner. Let's consider a resource file "test.rc" containing the following instructions:

- use maligno
- set LHOST 192.168.1.10
- set PROFILE standard
- genclient
- run

Invoking the “resource” command, with the resource file name as an argument, should execute all the instructions automatically.

```
BT3 > resource test.rc

[*] Running resource file...


[+] LHOST => 192.168.1.10


[+] PROFILE => standard


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


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

Fig. 7: Loading a "test.rc" resource file

Resource files should be able to run any actions supported by a module. However, resource files can only execute commands within a single module in use.

- **Version check and updates**
  The application’s current version can be displayed with “version”, while “bt3update” will check for new updates. The update mechanism is able to download and deploy new updates on demand. Updates will be deployed in a new folder at the same directory level as the existing installation. This means that the existing installation will remain as it is without modifications, which reduces the risk for inconsistencies or data loss.

```
BT3 > version

[*] You are running Blue Team Training Toolkit (BT3) version 2.7
```

Fig. 8: Results of the "version" command

```
BT3 > bt3update

[*] Checking for updates...

[*] You are running the latest version of the Blue Team Training Toolkit.
```

Fig. 9: Blue Team Training Toolkit can check for new updates on demand

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

```
BT3 > show modules

    Module            Description
    ------            -----------
    maligno           Attack simulation with customized malware indicators.
    mocksum           Repository of harmless files mimicking malware samples via hash collisions.
    pcapteller        Network traffic manipulation and replay.
```

Fig. 10: List of tools (modules) contained in BT3

2.4 Blue Team Training Toolkit Content Subscription API

Blue Team Training Toolkit offers an optional content subscription via an online API, which includes realistic network traffic related to a wide range of network attacks, mock malware samples, and important malware indicator profiles. The online library is growing constantly, and ensures a "plug & play" experience, when planning and preparing a training session.

The following sections document the most important aspects of creating and managing a subscription account.

- **Content subscription account creation**
  New content subscription accounts can be created with "apisignup". This should start a wizard that will guide you through the creation process.

```
BT3 > apisignup
```

Fig. 11: Starting the BT3 API account creation process

The first step during the account creation process will require you to read and accept the Blue Team Training Toolkit terms and conditions, and privacy policy. These can be found at https://www.bt3.no/terms-conditions/ and https://www.bt3.no/privacy-policy/ respectively.

```
[*] You are about to create a BT3 content subscription account.
[*] Before doing so, you need to go through the steps described below.
[*] You may press [CTRL+C] to abort this account registration at any time.

   1. Terms and Conditions
      BT3 content subscription accounts are free, and they are bound to a license.
      You must read the terms and conditions listed at https://www.bt3.no/terms-conditions

   2. Privacy Policy
      Your name and e-mail address will be gathered during the registration process.
      Encripto AS will use this information as follows:

      - To give you access to the BT3 content subscription service.
      - To be in touch with you, once you have become a user.
      - To send you information about BT3 updates and marketing (you may unsubscribe at any time).

      You can read our complete privacy policy at https://www.bt3.no/privacy-policy

   3. Acceptance and Consent
      Do you agree with the terms and conditions, as well as with the privacy policy? [Y/N]: Y
```

Fig. 12: Step 1 - Accepting the Blue Team Training Toolkit terms and conditions, and privacy policy

The second step will gather some basic information about you (full name).

```
[*] What may I call you?

    First Name: Test
     Last Name: Account
```

Fig. 13: Step 2 - Some basic personal information will be gathered

The third step will require you to choose a content license (Personal or Enterprise), according to the terms and conditions already accepted in the previous steps.

```
[*] Would you like to use a Personal or Enterprise content license?
[*] Check https://www.bt3.no/terms-conditions for more details.

    License [P/E]: E
```

Fig. 14: Step 3 - Selecting a content subscription license

The forth step ask you to provide a valid e-mail address, which will be used as user name and for password recovery purposes. BT3 will proceed to create your account once your e-mail address is provided. You will be able to verify your account and set credentials as soon as this process is finished.

```
[*] What is your e-mail address?
[*] This will be used as your username, and for account verification purposes.

    E-mail: user@yourdomain.com

[+] Your account has been successfully created!

[*] We will now proceed to verify your e-mail address and set your account credentials.
    Would you like to continue? [Y/N]: Y
```

Fig. 15: Step 4 - Providing an e-mail address as user name

The fifth step will proceed to verify the given e-mail address. A security code will be sent via e-mail, and should be provided as account verification proof.

```
[*] You are about to set new credentials for your BT3 API account.
[*] A security code will be sent to your registered e-mail address during this process.

[*] You may press [CTRL+C] to abort this new credentials request at any time.
[*] If you do so, you may type 'apinewcreds' to restart this process.

    E-mail Code:
```

Fig. 16: Step 5 - E-mail verification process

The last step will allow you to set your credentials.

```
[*] It's time to choose a good password.
[*] The minimum length is 10 characters, as well as a mix of lowcase and uppercase letters, digits and punctuation characters.
[*] Passwords will not be echoed during this process.

    New Password:
         Confirm:
```

Fig. 17: Step 6 - Setting credentials

The your account should be ready for use at this point.

```
[+] Your credentials were successfully updated.
[*] You may now type 'apiconnect' and log in with your account.
```

Fig. 18: Successful account creation

- **Content subscription account authentication**
  Existing accounts can authenticate directly from Blue Team Training Toolkit with the “apiconnect” command. This will require an e-mail address as user name and its associated password.

```
BT3 > apiconnect

[*] Please, enter your BT3 API credentials (password not echoed)

      E-mail:
```

Fig. 19: Account authentication process

```
[*] Hello Test, Welcome to the BT3 with API powers!
[*] Last login: 2018-01-25 19:27:41 (UTC)
[*] Last failed login: Never
```

Fig. 20: Welcome message after successful authentication

- **Content subscription account log out**
  Authenticated accounts can log out by using the “apidisconnect” command.

```
BT3 > apidisconnect

[*] Disconnected from BT3 API. You are working in offline mode now.
```

Fig. 21: Disconnecting from the BT3 API

- **Content subscription account details**
  Authenticated accounts can check subscription details by typing “show subscription”.

```
BT3 > show subscription

[*] Blue Team Training Toolkit - Content Subscription Details

            User: Test Account
          E-mail:
         License: Enterprise
      Last login: 2018-01-25 19:28:32 (UTC)
Last failed login: Never


[*] Blue Team Training Toolkit - Content Credit Details

         Balance: 0 credit(s)
    Last Purchase: Never
    Credit Expiry: 2018-01-24 11:50:10 (UTC)

    Content credits allow you to download premium training material.
    Read more about this topic at https://www.bt3.no/content-credits/


[*] Blue Team Training Toolkit - Content Enterprise License

    License terms can be found at https://www.bt3.no/terms-conditions
```

Fig. 22: API subscription details

- **Content subscription account credentials reset**
  Registered accounts may request a password change or account recovery by invoking “apinewcreds”.

```
BT3 > apinewcreds
```

Fig. 23: Requesting a new set of credentials

The first step of this process will require a valid e-mail address associated with an existing account. A security code will be sent to such address, and the code must be provided as verification proof.

```
[*] You are about to set new credentials for your BT3 API account.
[*] A security code will be sent to your registered e-mail address during this process.

[*] You may press [CTRL+C] to abort this new credentials request at any time.
[*] If you do so, you may type 'apinewcreds' to restart this process.

        E-mail:
```

Fig. 24: Step 1 - Account verification during new credentials request

The last step will require you to provide a new set of credentials.

```
[*] It's time to choose a good password.
[*] The minimum length is 10 characters, as well as a mix of lowcase and uppercase letters, digits and punctuation characters.
[*] Passwords will not be echoed during this process.

    New Password:
         Confirm:
```

Fig. 25: Step 2 - Setting new credentials

The new credentials will be ready for use as soon as the process has been completed.

Fig. 26: Successful credentials reset

- **Content subscription credit voucher redemption**
  Users who have purchased content credits will gain access to a credit voucher. The voucher can be redeemed with by invoking “apiredeem”, with the voucher code as argument. Please, note that redeeming a code requires an authenticated API session.

Fig. 27: Successful voucher redemption

- **Content subscription account deletion**
  Your content subscription account can be deleted at any time by invoking “apidelete” while being authenticated with your user account. Beware any information associated with your user account, credit balance and licensed materials will be lost once the command is completed. This operation cannot be reverted.

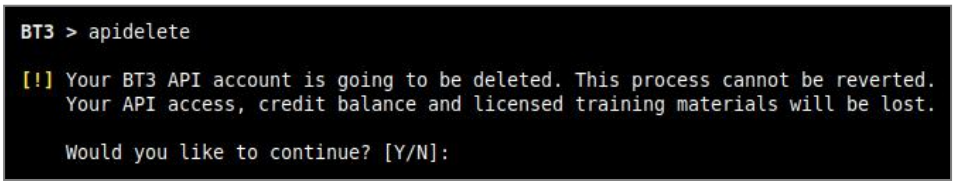

Fig. 28: API account deletion will require confirmation

Once the account is deleted, you will be able to use the Blue Team Training Toolkit in offline mode. Any training materials previously downloaded to your hard disk will not be destroyed during the content subscription account deletion process.

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.

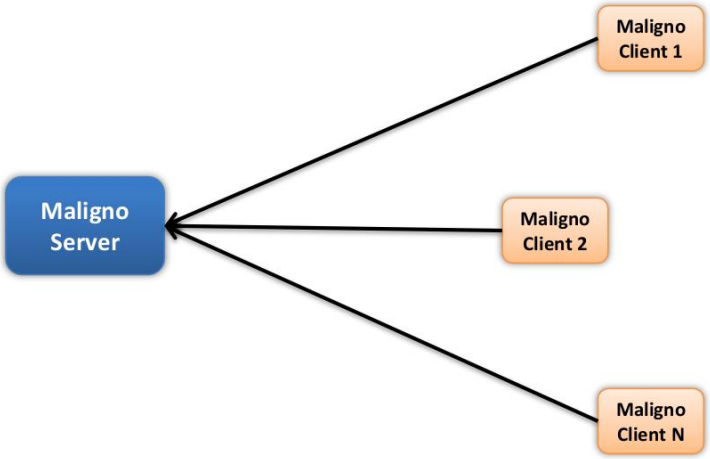

Fig. 29: Maligno clients can be distributed among multiple machines

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. 30: 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 Basic NTLM	Unauthenticated Basic NTLM	Successful
*nix / OS X	Unauthenticated Basic	Unauthenticated Basic	Successful

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. 31: Maligno module ready for use after invocation

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

```
BT3 ~ maligno > version

[*] You are running Maligno version 3.8
```

Fig. 32: Maligno version command output

- **Module help overview**
  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.
    download <profile>      Download a given profile from the Blue Team Training Toolkit cloud.
    exit                    Exit the Blue Team Training Toolkit.
    genclient               Generate a client with the current configured settings.
    help                    Display help menu.
    info <profile>          Display detailed information about a malware indicator profile.
    run                     Run the module with the given options.
    search <string>         Find malware indicator profiles based on a given string.
    set <option> <value>    Set module option.
    show downloads          Display a history of malware indicator profiles downloaded from the cloud.
    show interfaces         Display available network interfaces.
    show options            Display module options.
    show profiles           Display all available malware indicator profiles.
    show profiles cloud     Display malware indicator profiles available in the cloud.
    show profiles disk      Display malware indicator profiles available on your computer.
    show profiles free      Display free malware indicator profiles available in the cloud.
    show profiles premium   Display premium malware indicator profiles available in the cloud.
    version                 Display module version.
```

Fig. 33: List of commands supported by the module

- **Module network interfaces overview**
  Available network interfaces can be displayed with the “show interfaces” command. This is useful for checking the IP address assigned to your computer, without leaving the BT3 console.

```
BT3 ~ maligno > show interfaces

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::212:34ff:fe56:7800  prefixlen 64  scopeid 0x20<link>
        ether 00:12:34:56:78:00  txqueuelen 1000  (Ethernet)
        RX packets 330  bytes 212471 (207.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1218  bytes 98353 (96.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Fig. 34: Listing available network interfaces

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

```
BT3 ~ maligno > show options

    Name             Setting     Required    Description
    ----             -------     --------    -----------
    LHOST                        True        IP address or FQDN to expose the C2 server on.
    LPORT            80          True        TCP Port to listen for connections.
    PROFILE                      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.
    SSL_TRUST        True        False       Disable Maligno client SSL/TLS certificate verification.
```

Fig. 35: Module options and their current values

Maligno Module Options	
Name	**Description**
LHOST	Defines the IP address or Fully-Qualified Domain Name (FQDN) where the Maligno server component will be exposed. 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.
LPORT	Defines the TCP port to listen for incoming connections. This value is actively used by the Maligno client generation process. Maligno clients will attempt connections to the port provided by this option.
PROFILE	Defines the name of the Maligno malware indicator profile in use. Valid profiles can be listed with “show profiles”.
SSL	Defines whether the Maligno server component will support SSL/TLS for incoming connections.
SSL_CERT	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. Additional certificates can be used, as long as they are placed in the “certs” directory, within the BT3’s installation folder.
SSL_TRUST	Defines whether the Maligno client component will disable SSL/TLS certificate verification. By default, certificate verification is disabled. Therefore, Maligno clients will accept self-signed certificates automatically when establishing HTTPS connections with Maligno server. If certificate verification is enabled, Maligno server will be required to use a certificate issued by a trusted certificate authority. If certificate validation is enabled, but Maligno server uses a self-signed certificate, a manual deployment of the server certificate chain must be done prior to client execution. In such case, the certificate chain must be imported onto the client machine's certificate trust, typically handled by the client machine’s operating system.

- **Module option configuration**
  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. 36: Setting a new option value

- **Module material list**
  Available malware indicator profiles can be listed with “show profiles”. If a content subscription account is already authenticated, the command will retrieve available profile information from the online library.

```
BT3 ~ maligno > show profiles

    File             Size (MB)    Location    Date          Price      Description
    ----             ---------    --------    ----          -----      -----------
    ghostnet_php.py  0.003        Disk        2012-01-01                Ghostnet APT profile based on PHP technology.
    oldrea.py        0.003        Disk        2014-07-07                Oldrea APT backdoor profile.
    zemot.py         0.003        Disk        2014-11-23                Zemot trojan profile.
    adposhel_1       0.008        Cloud       2016-08-19    1 Cred.     Adposhel adware checkin blocked by web proxy.
    aridviper_1      0.003        Cloud       2016-08-11    1 Cred.     Operation Arid Viper. Malware indicators for interesting document.
    aridviper_2      0.003        Cloud       2016-08-11    1 Cred.     Operation Arid Viper. Malware indicators for uninteresting document.
    bandarchor       0.004        Cloud       2016-08-27    1 Cred.     Bandarchor ransomware profile.
    bergard          0.004        Cloud       2016-09-07    1 Cred.     Bergard trojan, related to the C0d0so APT group and an attack against Forbes.com.
    bitcoin_miner    0.003        Cloud       2016-08-25    1 Cred.     Bitcoin miner malware profile.
    cerber_check     0.003        Cloud       2016-08-14    1 Cred.     Cerber ransomware public IP address check.
```

Fig. 37: Example with a few Maligno malware indicator profiles ready for use on disk

```
BT3 ~ maligno > show profiles

    Profile           Location    Date          Price    Description
    -------           --------    ----          -----    -----------
    adposhel_1        Cloud       2016-08-19    1        Adposhel adware checkin blocked by web proxy.
    arid_viper_1      Cloud       2016-08-11    1        Operation Arid Viper. Malware indicators for interesting document.
    arid_viper_2      Cloud       2016-08-11    1        Operation Arid Viper. Malware indicators for uninteresting document.
    cerber_check      Cloud       2016-08-14    1        Cerber ransomware public IP address check.
    conficker_check   Cloud       2016-08-12    1        Conficker malware checking external IP address.
    conficker_sink    Cloud       2016-08-12    1        Conficker malware with sinkhole response.
    cookie            Cloud       2016-06-26    0        Default profile with cookie header and random elements.
    CORESHELL         Cloud       2016-08-12    1        CORESHELL APT malware profile.
    core_bot          Cloud       2016-08-10    1        Core Bot banking trojan.
    cryptowall_v3     Cloud       2015-02-13    0        Cryptowall v3 ransomware profile.
    enfal             Cloud       2016-08-10    1        Enfal (aka Lurid). Malware used in targeted attacks.
    etumbot           Cloud       2014-07-01    0        Etumbot APT backdoor profile.
    explosive         Cloud       2016-08-11    1        Explosive APT malware used by Volatile Cedar campaign.
```

Fig. 38: Fragment of the online profile library

More targeted profile listing can be achieved with “show profiles cloud”, “show profiles disk”, “show profiles free” and “show profiles premium”. These commands will present all malware indicator profiles available online, profiles found locally on your computer, profiles which can be downloaded for free, and profiles which can be downloaded with the use of content credits respectively.

- **Material search**
  Malware indicator profiles can be easily found with the “search” command. Searches use the profile name or profile description as criterion.

```
BT3 ~ maligno > search standard

    File             Size (MB)    Location    Date          Price      Description
    ----             ---------    --------    ----          -----      -----------
    standard         0.003        Cloud       2016-06-26    0 Cred.    Default profile with static elements.
    standard_random  0.003        Cloud       2016-06-26    0 Cred.    Default profile with random elements.


[*] Search results: 2
```

Fig. 39: Search results presented by the module

- **Material information**
  Detailed information about a given malware indicator profile can be shown with the “info” command. The expected command argument is the profile to present. Note that malware indicator profiles downloaded to your local disk will have a “.py” extension, while those online do not.

```
BT3 ~ maligno > info standard.py

[*] Blue Team Training Toolkit - Malware Indicator Profile Details

          Name: standard.py
          Date: 2016-06-26
          Size: 0.003 MB
      Location: Disk
        Author: Juan J. Guelfo @ Encripto AS
   Description: Default profile with static elements.
    References: https://www.encripto.no/en/services/network-security-monitoring/

[!] Do you need more information?
[!] Check the reference link above for further details on what to expect from this training material.
```

Fig. 40: Details about a malware indicator profile found on disk

- **Material download**
  Authenticated content subscription accounts will have access to the Blue Team Training Toolkit online library, with both free and premium training content. Such material is ready for use and offers a "plug & play" experience.

  Premium training content has a price, which will be deducted from the user's existing content credit balance. Premium downloads require users to have enough credit balance in order to complete the download. Free online content, on the other hand, can be downloaded without restrictions.

  Downloading online resources can be done with the "download" command, and the material name provided as an argument.

```
BT3 ~ maligno > download standard

[!] This download will deduct 0 credit(s) from your content credit balance.
    Would you like to continue? [Y/N]: y

[*] Downloading...
[+] Congratulations! New training material is now available on your disk.
```

Fig. 41: Successful material download

- **Material download history**
  The training material download history associated with your subscription account can be retrieved with "show downloads".

```
BT3 ~ maligno > show downloads

    Name            Type                      Timestamp (UTC)
    ----            ----                      ---------------
    oldrea          Malware indicator profile 2018-01-25 14:47:24
    oldrea          Malware indicator profile 2018-01-25 14:48:53
    cookie          Malware indicator profile 2018-01-25 15:38:40
    tinba           Malware indicator profile 2018-01-25 15:40:06
    tinba           Malware indicator profile 2018-01-25 15:40:13
    potao           Malware indicator profile 2018-01-25 15:43:23
    zemot           Malware indicator profile 2018-01-25 18:35:20
    oldrea          Malware indicator profile 2018-01-25 19:33:30
    zemot           Malware indicator profile 2018-01-25 19:33:36
    ghostnet_php    Malware indicator profile 2018-01-25 19:33:47
    standard        Malware indicator profile 2018-01-25 19:39:05


[*] Downloads: 11
```

Fig. 42: Material download history

- **Maligno client generation**
  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 stored in 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. 43: Successful Maligno client generation

```
root@demo:~/BT3-2.8/clients# ls -l
total 28
-rw-r--r-- 1 root root 27569 Aug 31 16:26 maligno_client_standard.py
```

Fig. 44: Generated clients are placed in a specific location

- **Module execution**
  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. 45: Maligno server is running and waiting for connections

3.2 Malware Indicator Profiles

Maligno's malware indicator 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 indicator 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 Indicator Profile Structure		
Class	**Purpose**	**Value Visibility**
Info	Gathers general information about the Maligno malware indicator 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 Indicator Profile Attributes		
Class	**Attribute**	**Purpose**
Info	author	Defines who created the malware indicator profile.
	Date	Describes when the profile was created.
	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", "PUT", "HEAD", "DELETE", "TRACE", "OPTIONS", "DEBUG" or "PATCH".
	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	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”. 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.
	delay	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). Note that a delay of “0” seconds will generate a huge amount of requests in a short period of time.
	jitter	Defines a random deviation that will be added to the delay time. The value is understood as percentage of the delay time. For example, a delay time of 10 seconds and a jitter of 50% will result in a maximum waiting time of 15 seconds.

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 with Havex. BT3 includes a Maligno malware indicator 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 ruleset 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 LHOST 192.168.1.10

[+] LHOST => 192.168.1.10

BT3 ~ maligno > set PROFILE oldrea

[+] PROFILE => oldrea

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.
    SSL_TRUST        True            False       Disable Maligno client SSL/TLS certificate verification.

BT3 ~ maligno >
```

Fig. 46: 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. 47: 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. 48: Maligno server running and receiving a client request during the course of the exercise

```
 ======================================================================
|                 Blue Team Training Toolkit (BT3)                     |
|                      Maligno module v3.8                             |
|                                                                      |
|  By Juan J. Guelfo | Encripto AS | www.bt3.no | support@bt3.no        |
 ======================================================================

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

[*] Preparing request #2...
[*] Sending request via direct connection...
[+] Request sent...
[*] Sleeping 11s...
```

Fig. 49: 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 and 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. 50: Snort IDS alert triggered by the network activity

Fig. 51: One of the HTTP requests captured during the course of the exercise (UTC time zone)

3.4 Using Maligno Clients with a Proxy Server

As covered in previous sections, 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 Basic NTLM	Unauthenticated Basic NTLM	Successful
*nix / OS X	Unauthenticated Basic	Unauthenticated Basic	Successful

In order to enable Maligno client proxy support, the client machine's operating system must be configured properly. If you wish to execute the Maligno client component with proxy support on Windows, the correct configuration should be provided via the system proxy configuration. These options are typically available as an advanced setting for Internet Explorer or Edge browsers.

Enabling Web Proxy Auto Discovery (WPAD) will allow Maligno clients to auto detect proxy servers in your network. If credentials were required during the WPAD process (in cases where the PAC is protected), Maligno clients will prompt you to enter credentials.

Fig. 52: Enabling Web Proxy Auto Discovery (WPAD) - Windows 10

Fig. 53: Enabling Web Proxy Auto Discovery (WPAD) - Windows 7

Alternatively, a specific proxy server could be provided manually instead. The following examples will use proxy server "192.168.1.15" listening on port "8080". In this case, the given proxy server will be contacted directly by Maligno clients without prior auto detection. If the proxy server requires authentication, Maligno clients will prompt you to enter credentials.

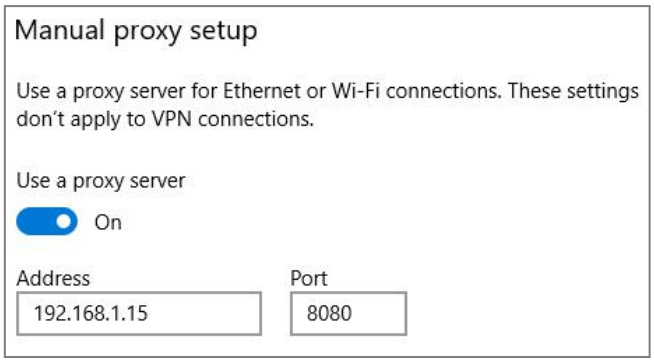

Fig. 54: Manual proxy server configuration - Windows 10

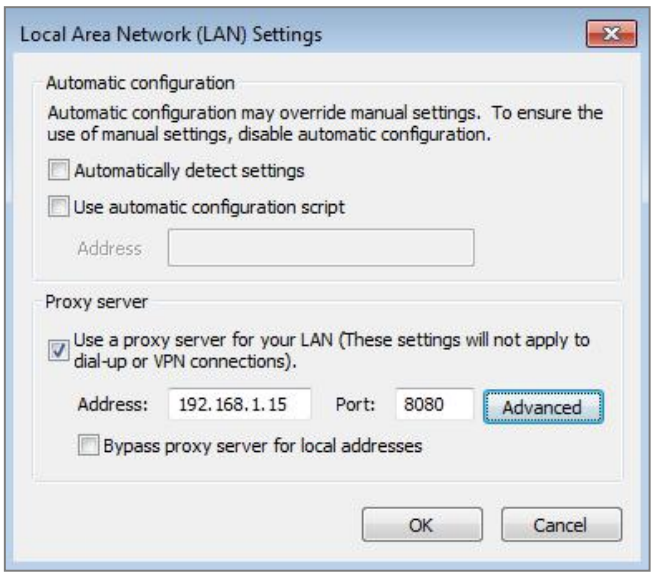

Fig. 55: Manual proxy server configuration - Windows 7

On the other hand, Unix-based and Mac OS systems can specify proxy settings by exporting variables prior to Maligno client execution from a terminal. The following example illustrates this process in Kali Linux.

```
root@demo:~/BT3-2.8/clients# export http_proxy="http://192.168.1.15:8080"
root@demo:~/BT3-2.8/clients# export https_proxy="https://192.168.1.15:8080"
root@demo:~/BT3-2.8/clients# python maligno_client_standard.py
```

Fig. 56: Proxy configuration from a Linux terminal

If the proxy server requires authentication, Maligno clients will prompt you to enter credentials.

4. BT3 Module: Pcapteller

Pcapteller 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, Pcapteller reads network packets from a PCAP file, and replays them into the network. The module allows packet manipulation (MAC addresses, IP addresses, and packet payloads) prior to replay, so it is possible to customize the traffic with specific addresses and indicators 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, Pcapteller will allow you to reuse existing PCAP files and make everything look like the attack is really happening in your own network. Pcapteller can help you improve 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. 57: Pcapteller module ready for use after invocation

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

```
BT3 ~ pcapteller > version

[*] You are running Pcapteller version 1.9
```

Fig. 58: Pcapteller version command output

- **Module help overview**
  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.
    download <pcap>         Download a given PCAP file from the Blue Team Training Toolkit cloud.
    exit                    Exit the Blue Team Training Toolkit.
    help                    Display help menu.
    info <pcap>             Display detailed information about a PCAP file.
    run                     Run the module with the given options.
    search <string>         Find PCAP files based on a given string.
    set <option> <value>    Set module option.
    show downloads          Display a history of PCAP files downloaded from the cloud.
    show interfaces         Display available network interfaces.
    show options            Display module options.
    show pcaps              Display all available PCAP files.
    show pcaps cloud        Display PCAP files available in the cloud.
    show pcaps disk         Display PCAP files available on your computer.
    show pcaps free         Display free PCAP files available in the cloud.
    show pcaps premium      Display premium PCAP files available in the cloud.
    version                 Display module version.
```

Fig. 59: List of commands supported by the module

- **Module material list**
  PCAP files available for use can be listed with “show pcaps”. If a content subscription account is already authenticated, the command will retrieve available PCAP information from the online library.

Fig. 60: Example with available PCAP files on disk

Fig. 61: Fragment of the online PCAP library

More targeted profile listing can be achieved with “show pcaps cloud”, “show pcaps disk”, “show pcaps free” and “show pcaps premium”. These commands will present all PCAP files available online, PCAP files found locally on your computer, PCAP files which can be downloaded for free, and PCAP files which can be downloaded with the use of content credits respectively.

- **Material search**
  Available PCAP files can be easily found with the “search” command. Searches use the PCAP file name or its description as criterion.

Fig. 62: Search results presented by the module

- **Material information**
  Detailed information about a given PCAP file can be shown with the “info” command. The expected command argument is the PCAP file to present. Note that PCAP files downloaded to your local disk will have a “.pcap” extension, while those online do not.

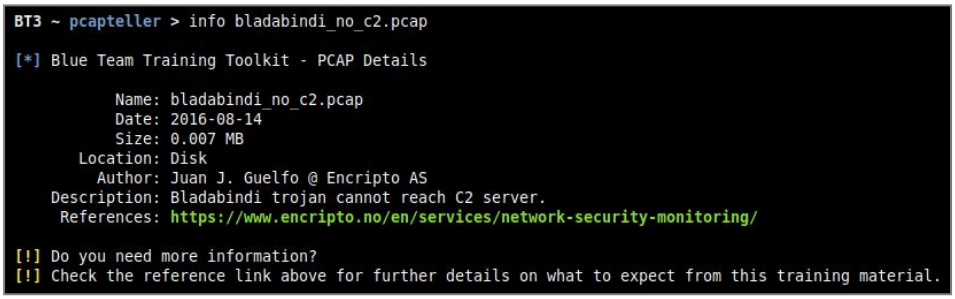

Fig. 63: Details about a PCAP file found on disk

- **Material download**
  Authenticated content subscription accounts will have access to the Blue Team Training Toolkit online library, with both free and premium training content. Such material is ready for use and offers a “plug & play” experience.

  Premium online training content has a price, which will be deducted from the user’s existing content credit balance. Premium downloads require users to have enough credit balance in order to complete the download. Free online content, on the other hand, can be downloaded without restrictions.

Downloading online resources can be done with the “download” command, and the material name provided as an argument.

```
BT3 ~ pcapteller > download bladabindi_no_c2

[!] This download will deduct 0 credit(s) from your content credit balance.
    Would you like to continue? [Y/N]: y

[*] Downloading...
[+] Congratulations! New training material is now available on your disk.
```

Fig. 64: Successful material download

- **Material download history**
  The training material download history associated with your subscription account can be retrieved with “show downloads”.

```
BT3 ~ pcapteller > show downloads

    Name                    Type         Timestamp (UTC)
    ----                    ----         ---------------
    icloader                PCAP file    2018-01-25 17:29:44
    CVE-2012-0158_payload   PCAP file    2018-01-25 18:39:16
    cybergate_rat           PCAP file    2018-01-25 18:43:16
    dreambot_trojan         PCAP file    2018-01-25 18:45:01
    bladabindi_no_c2        PCAP file    2018-01-26 11:33:34
    demo                    PCAP file    2018-01-26 11:39:04
    bladabindi_no_c2        PCAP file    2018-01-26 11:40:53


[*] Downloads: 7
```

Fig. 65: Material download history

- **Module network interfaces overview**
  Available network interfaces can be displayed with the “show interfaces” command. This is useful for checking what interfaces can be used for traffic replay, without leaving the BT3 console.

```
BT3 ~ pcapteller > show interfaces

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        ether 00:12:34:56:78:00  txqueuelen 1000  (Ethernet)
        RX packets 333  bytes 213497 (208.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1272  bytes 108925 (106.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Fig. 66: Listing available network interfaces

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

```
BT3 ~ pcapteller > show options

    Name             Setting     Required    Description
    ----             -------     --------    -----------
    FILE                         True        Pcap file to replay in libpcap format.
    FRAGMENTATION    False       True        Fragment packets during replay. Useful for networks with low MTU.
    INTERFACE                    True        Network interface to replay the packets with.
    MTU              1500        True        MTU to use with packet fragmentation.
    PCAP_IP_LIST                 False       Comma-separated list of IP addresses to replace as seen on the pcap file.
    PCAP_MAC_LIST                False       Comma-separated list of MAC addresses to replace as seen on the pcap file.
    PCAP_PLD_LIST                False       Comma-separated list of packet payloads to replace as seen on the pcap file.
    PROTOCOL_LIST                False       Comma-separated list of protocol layers, which packet payload manipulation will apply to.
    REAL_TIME        False       False       Honor inter-packet arrival time while replaying traffic.
    WIRE_IP_LIST                 False       Comma-separated list of IP addresses to replay as seen on the wire.
    WIRE_MAC_LIST                False       Comma-separated list of MAC addresses to replay as seen on the wire.
    WIRE_PLD_LIST                False       Comma-separated list of packet payloads to replay as seen on the wire.

BT3 ~ pcapteller >
```

Fig. 67: Module options and their current values

Pcapteller 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.
PCAP_PLD_LIST	Defines a comma-separated list of payload values to replace. The payload values must be defined as seen in the PCAP file. The payload values defined in the PCAP_PLD_LIST will be replaced by the same element index of the WIRE_PLD_LIST.
PROTOCOL_LIST	Defines a comma-separated list of protocol layers, which packet payload manipulation will apply to. Supported values are “DNS”, “NBNS”, “SMB”, “RAW”. When choosing “DNS”, the payload of both DNS request and response packets will be manipulated. The same behavior will also apply to “NBNS”. On the other hand, “RAW” will allow you to manipulate packets that contain raw payloads, such as HTTP, UDP or ICMP. Manipulation in this case is limited to ASCII values contained by the payload.
REAL_TIME	Defines whether inter-packet arrival timing should be honored. When this option is enabled, Pcapteller 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.

WIRE_PLD_LIST	Defines a comma-separated list of payload values to inject during the traffic manipulation phase. Such values will be visible on the wire during the traffic replay. The payload values defined in the PCAP_PLD_LIST will be replaced by the same element index of the WIRE_PLD_LIST.

- **Module option configuration**
  Module option values can be set with the "set" command, the desired option and its new value.

```
BT3 ~ pcapteller > set INTERFACE eth0

[+] INTERFACE => eth0

BT3 ~ pcapteller >
```

Fig. 68: Setting a new option value

- **Module execution**
  Once all required module options have been configured with valid values, Pcapteller can begin to replay packets with the "run" command. All module options are validated prior to execution.

```
BT3 ~ pcapteller > run

[*] Checking packet payload manipulation parameters...
[*] Pcapteller started at 23:39:22. Press [CTRL+C] to stop.

[*] Reading "/root/BT3-2.3/pcaps/demo.pcap"...
[+] 17398 packet(s) found.

[*] Processing 17398 of 17398 packet(s) | Error: 0 packet(s).
[*] Replaying packet(s) via eth0...
[+] Replay complete.

[*] Pcapteller finished at 23:39:36.
```

Fig. 69: Successful packet replay with Pcapteller

Recommendations to prevent inconsistencies during traffic replay

In order to reduce the chances of generating inconsistent packets while using packet payload manipulation, it is recommended to replace pairs of payload values with the same size or length. In other words, each pair of elements from the PCAP_PLD_LIST and WIRE_PLD_LIST should have the same size.

For example, if you want to replace an HTTP URI value defined as "/website/index.html" in a PCAP file, the recommended approach is to choose a URI with the same length as replacement. A good example could be then "/thisIsReplaced.php". As you may see, both strings have 19 ASCII characters (same size), which ensures an optimal replacement condition.

The example could be implemented with the following options:

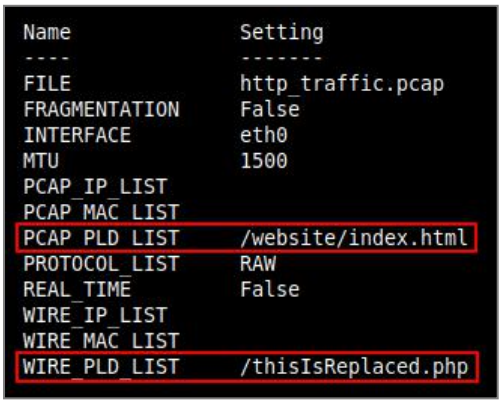

Fig. 70: Payload values with the same size

When an existing piece of data (found in the network traffic of the original PCAP) is replaced with a new value, Pcapteller will recalculate packet lengths, sizes and checksums. This is done to ensure that valid network traffic is generated during the replay phase.

You should be aware of the possibility of encountering small inconsistencies when inspecting manipulated network traffic with network protocol analyzers (e.g. Wireshark), if the pairs of payload values defined in the PCAP_PLD_LIST and the WIRE_PLD_LIST have different lengths. Pcapteller will always warn you if such type of situation is detected prior to replaying network traffic.

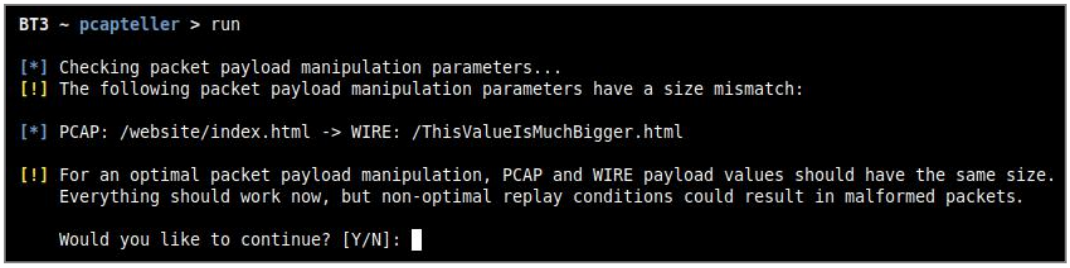

Fig. 71: Warning informing about non-optimal replay conditions

Replacing packet payloads with values that have same size is specially important when manipulating some RAW protocol layers, such as HTTP. On the other hand, UDP or DNS should not encounter issues.

4.2 PCAP Metadata Modules

In order to present metadata related to PCAP files already exist on disk, Pcapteller uses simple Python modules that are deployed together with the actual PCAP files. For example, “demo.py” will store metadata for “demo.pcap”.

Fig. 72: Metadata information presented by Pcapteller module

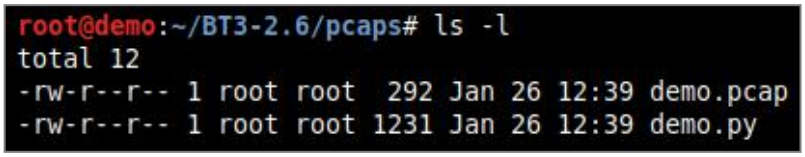

Fig. 73: PCAP file with its metadata module deployed on disk

Metadata modules are structured as follows:

Pcapteller - PCAP Metadata Module Structure		
Class	**Purpose**	**Value Visibility**
Info	Gathers general information about the PCAP file.	BT3's command-line interface.

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

Pcapteller - PCAP Metadata Module Attributes		
Class	**Attribute**	**Purpose**
Info	author	Defines who created the PCAP file.
	Date	Describes when the PCAP file was created.
	description	Provides a summary of what kind of communications PCAP file contains. This field will be visible on the BT3's "show pcaps" and "search" commands output.
	license	Describes the license model applied to the PCAP file.
	references	Includes links to threat intelligence reports or other materials that backup the behavior represented by the PCAP file.

4.3 Setting up Pcapteller

This section is going to demonstrate how BT3's Pcapteller 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/41521313735218521O 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. 74: 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 ~ pcapteller > set PCAP_IP_LIST 192.168.122.70

[+] PCAP_IP_LIST => 192.168.122.70

BT3 ~ pcapteller > set WIRE_IP_LIST 172.31.10.11

[+] WIRE_IP_LIST => 172.31.10.11

BT3 ~ pcapteller > 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       Comma-separated list of MAC addresses to replace as seen on the pcap file.
    PCAP_PLD_LIST                     False       Comma-separated list of packet payloads to replace as seen on the pcap file.
    PROTOCOL_LIST                     False       Comma-separated list of protocol layers, which packet payload manipulation will apply to.
    REAL_TIME        False            False       Honor inter-packet arrival time while replaying traffic.
    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       Comma-separated list of MAC addresses to replay as seen on the wire.
    WIRE_PLD_LIST                     False       Comma-separated list of packet payloads to replay as seen on the wire.

BT3 ~ pcapteller >
```

Fig. 75: 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 ~ pcapteller > run

[*] Checking packet payload manipulation parameters...
[*] Pcapteller started at 16:23:49. Press [CTRL+C] to stop.

[*] Reading "/root/BT3-2.3/pcaps/test.pcap"...
[+] 17398 packet(s) found.

[*] Processing 17398 of 17398 packet(s) | Error: 0 packet(s).
[*] Replaying packet(s) via eth0...
[+] Replay complete.

[*] Pcapteller finished at 16:24:03.
```

Fig. 76: Running BT3's Pcapteller 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. 77: Fragment of the manipulated PCAP file with attacker IP address and the victim (172.31.10.11)

Since Pcapteller 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 ruleset) 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 (iq3ahljcfeont3xx)
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. 78: Alerts generated by Intrusion Detection System (Snort) during the execution of the example

4.4 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 Pcapteller 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 "172.16.50.0/24", 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 premium training material "cryptxxx_ransom" will downloaded from the BT3 cloud and later used during the case.

```
BT3 ~ pcapteller > search cryptxxx

    File             Size (MB)    Location    Date          Price      Description
    ----             ---------    --------    ----          -----      -----------
    cryptxxx_ransom  21.8         Cloud       2016-08-13    1 Cred.    Cryptxxx infection through EK


[*] Search results: 1

[!] PCAP material may contain infected files.
[!] Use caution when exporting objects from the network traffic.
```

Fig. 79: Material used in this example

The original PCAP file shows host "192.168.1.4" as victim. The MAC address of the gateway used by such host is "00:1f:33:c3:43:34".

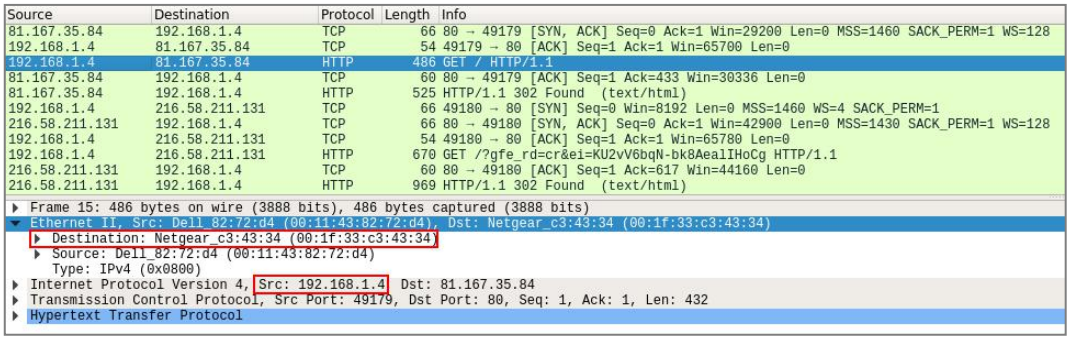

Fig. 80: Fragment of the original contents of the PCAP file

- **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 "172.16.50.111" (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             cryptxxx_ransom.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     192.168.1.4           False       Comma-separated list of IP addresses to replace as seen on the PCAP file.
    PCAP_MAC_LIST    00:1f:33:c3:43:34     False       Comma-separated list of MAC addresses to replace as seen on the PCAP file.
    PCAP_PLD_LIST                          False       Comma-separated list of packet payloads to replace as seen on the PCAP file.
    PROTOCOL_LIST                          False       Comma-separated list of protocol layers, which packet payload manipulation will apply to.
    REAL_TIME        True                  False       Honor inter-packet arrival time while replaying traffic.
    WIRE_IP_LIST     172.16.50.111         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.
    WIRE_PLD_LIST                          False       Comma-separated list of packet payloads to replay as seen on the wire.
```

Fig. 81: Module options prior to traffic manipulation

For even a more realistic look, "REAL_TIME" support will be enabled on Pcapteller. This will honor inter-packet arrival time 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".

Src IP	SPort	Dst IP	DPort	Pr	Event Message
172.16.50.111	49198	74.208.99.117	80	6	ET CURRENT_EVENTS Possible Job314/Neutrino Reboot EK Flash Exploit Jan 07 2015 M2
74.208.99.117	80	172.16.50.111	49198	6	ET CURRENT_EVENTS Job314/Neutrino Reboot EK Landing July 07 2016 M1
74.208.99.117	80	172.16.50.111	49198	6	ET CURRENT_EVENTS Job314/Neutrino Reboot EK Landing June 11 2016 M4 (with URI Primer)
195.128.174.138	80	172.16.50.111	49192	6	ET CURRENT_EVENTS Evil Redirector Leading to EK Jul 12 2016

Fig. 82: 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          Src port  Destination     Dst port  Protocol  Info
74.208.99.117         80  172.16.50.111      49198  TCP       [TCP segment of a reassembled PDU]
74.208.99.117         80  172.16.50.111      49198  HTTP      HTTP/1.1 200 OK  (text/html)
172.16.50.111      49198  74.208.99.117         80  TCP       49198→80 [ACK] Seq=576 Ack=2509 Win=65700 Len=0
172.16.50.111      49198  74.208.99.117         80  HTTP      GET /tail/sneak-ride-10426604.swf HTTP/1.1
74.208.99.117         80  172.16.50.111      49198  TCP       80→49198 [ACK] Seq=2509 Ack=1117 Win=31616 Len=0
74.208.99.117         80  172.16.50.111      49198  TCP       [TCP segment of a reassembled PDU]
74.208.99.117         80  172.16.50.111      49198  TCP       [TCP segment of a reassembled PDU]
74.208.99.117         80  172.16.50.111      49198  TCP       [TCP segment of a reassembled PDU]
▸ Frame 10: 595 bytes on wire (4760 bits), 595 bytes captured (4760 bits)
▾ Ethernet II, Src: Dell_82:72:d4 (00:11:43:82:72:d4), Dst: PaloAlto_00:00:02 (00:1b:17:00:00:02)
  ▸ Destination: PaloAlto_00:00:02 (00:1b:17:00:00:02)
  ▸ Source: Dell_82:72:d4 (00:11:43:82:72:d4)
    Type: IP (0x0800)
```

Fig. 83: Fragment of the replayed traffic (network decoy)

5. BT3 Module: Mocksum

Mocksum is a module that provides access to a collection of files that mimic malware samples via MD5 hash collisions. The files downloaded via Mocksum allow you to simulate and plant realistic artifacts during training sessions, incident response drills or red team engagements. Without the risk of handling real malware. In a nutshell, these artifacts are harmless files that produce the same MD5 checksum as real malicious files. In many cases, the harmless artifacts also get detected by anti-virus software.

5.1 Getting Started

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

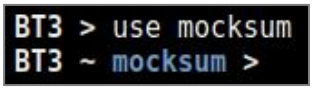

Fig. 84: Mocksum module ready for use after invocation

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

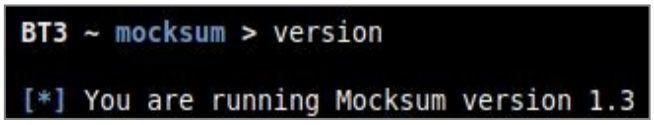

Fig. 85: Mocksum version command output

- **Module help overview**
  Mocksum supports a range of general commands, which can be displayed with “help”.

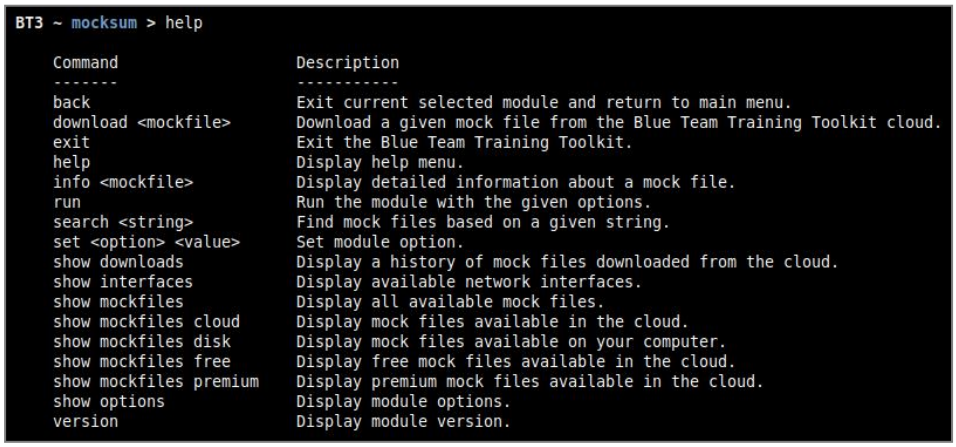

Fig. 86: List of commands supported by the module

- **Module material list**
  Mock files available for use can be listed with “show mockfiles”. If a content subscription account is already authenticated, the command will retrieve mock file information from the online library.

Fig. 87: Example with available mock files in the cloud

More targeted mock file listing can be achieved with “show mockfiles cloud”, “show mockfiles disk”, “show mockfiles free” and “show mockfiles premium”. These commands will present all mock files available online, mock files found locally on your computer, mock files which can be downloaded for free, and mock files which can be downloaded with the use of content credits respectively.

- **Material search**
  Available mock files can be easily found with the “search” command. Searches use the mock file name or its description as criterion.

```
BT3 ~ mocksum > search win

    File                    Size (MB)    Location    Date          Price      Description
    ----                    ---------    --------    ----          -----      -----------
    win_x86_met_reverse_http  0.974        Cloud       2016-11-23    1 Cred.    Windows x86 meterpreter reverse http with MD5 hash collision.
    win_x86_pwd_bind_shell    0.016        Cloud       2016-11-23    1 Cred.    Windows x86 password protected bind shell with MD5 hash collision.
    win_x86_reverse_shell     0.019        Cloud       2016-11-17    0 Cred.    Windows x86 reverse shell with MD5 hash collision.

[*] Search results: 3
```

Fig. 88: Search results presented by the module

- **Material information**
  Detailed information about a given mock file can be shown with the “info” command. The expected command argument is the mock file to present. Note that mock files downloaded to your local disk will have a “.mock” extension, while those online do not.

```
BT3 ~ mocksum > info win_x86_reverse_shell.mock

[*] Blue Team Training Toolkit - Mock File Details

          Name: win_x86_reverse_shell.mock
          Date: 2016-11-17
          Size: 0.019 MB
      Location: Disk
        Author: Juan J. Guelfo @ Encripto AS
   Description: Windows x86 reverse shell with MD5 hash collision.
    References: https://www.encripto.no/en/services/network-security-monitoring/

[!] Do you need more information?
[!] Check the reference link above for further details on what to expect from this training material.
```

Fig. 89: Details about a PCAP file found on disk

- **Material download**
  Authenticated content subscription accounts will have access to the Blue Team Training Toolkit online library, with both free and premium training content. Such material is ready for use and offers a “plug & play” experience.

  Premium online training content has a price, which will be deducted from the user’s existing content credit balance. Premium downloads require users to have enough credit balance in order to complete the download. Free online content, on the other hand, can be downloaded without restrictions.

  Downloading online resources can be done with the “download” command, and the material name provided as an argument.

```
BT3 ~ mocksum > download win_x86_reverse_shell

[!] This download will deduct 0 credit(s) from your content credit balance.
    Would you like to continue? [Y/N]: y

[*] Downloading...
[+] Congratulations! New training material is now available on your disk.
```

Fig. 90: Successful material download

- **Material download history**
  The training material download history associated with your subscription account can be retrieved with “show downloads”.

```
BT3 ~ mocksum > show downloads

    Name                          Type         Timestamp (UTC)
    ----                          ----         ---------------
    linux_x64_netcat_bind_shell   Mock file    2018-01-25 17:56:02
    linux_x64_netcat_bind_shell   Mock file    2018-01-25 17:56:06
    win_x86_reverse_shell         Mock file    2018-01-26 16:59:03


[*] Downloads: 3
```

Fig. 91: Material download history

- **Module network interfaces overview**
  Available network interfaces can be displayed with the "show interfaces" command. This has been included in order to provide a more homogeneous command list among the different BT3 modules.

```
BT3 ~ mocksum > show interfaces

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        ether 00:12:34:56:78:00  txqueuelen 1000  (Ethernet)
        RX packets 16592  bytes 24629142 (23.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 131400  bytes 129564340 (123.5 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Fig. 92: Listing available network interfaces

- **Module option list**
  This module currently provides access to the mock file library and it has no configurable options.

- **Module execution**
  This module currently provides access to the mock file library and it cannot be run.

5.2 Mock File Metadata Modules

In order to present metadata related to mock files already exist on disk, Mocksum uses simple Python modules that are deployed together with the actual mock files. For example, "win_x86_reverse_shell.py" will store metadata for "win_x86_reverse_shell.mock".

```
BT3 ~ mocksum > show mockfiles disk

    File                      Size (MB)   Location   Date         Price     Description
    ----                      ---------   --------   ----         -----     -----------
    win_x86_reverse_shell.mock  0.019       Disk       2016-11-17             Windows x86 reverse shell with MD5 hash collision.
```

Fig. 93: Metadata information presented by Mocksum module

```
root@demo:~/BT3-2.6/mockfiles# ls -l
total 28
-rw-r--r-- 1 root root   799 Jan 25 20:09 __init__.py
-rw-r--r-- 1 root root 19566 Jan 26 17:59 win_x86_reverse_shell.mock
-rw-r--r-- 1 root root  1299 Jan 26 17:59 win_x86_reverse_shell.py
```

Fig. 94: Mock file with its metadata module deployed on disk

Metadata modules are structured as follows:

Mocksum - Mock File Metadata Module Structure		
Class	**Purpose**	**Value Visibility**
Info	Gathers general information about the mock file.	BT3's command-line interface.

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

Mocksum - Mock File Metadata Module Attributes		
Class	**Attribute**	**Purpose**
Info	author	Defines who created the mock file.
	Date	Describes when the mock file was created.
	description	Provides a summary of what kind of malware or shellcode the mock file mimics. This field will be visible on the BT3's "show mockfiles" and "search" commands output.
	license	Describes the license model applied to the mock file.
	references	Includes links to threat intelligence reports or other materials that backup the behavior represented by the mock file.

5.3 Next Steps

Once a mock file has been downloaded to your disk, you may plant it in your training environment. Multiple possibilities and goals can be accomplished with mock files, such as:

- **Flags**
  Mock files could be used as flags, which let the blue team know that a (simulated) malicious file has been found.

- **Mastering log correlation and third party threat intelligence**
  Mock files have MD5 hash collisions that mimic real malware samples. By calculating their checksums, your blue team can find real information about the mimicked malware sample in different sources.

  This kind of practice can allow the blue team to master event investigation, get used to using third party threat intelligence services, or correlate in-house logs (e.g. Centralized anti-malware solution).

For a more realistic experience, mock files can be renamed to ".exe" (Windows) or ".bin" (Linux). Since the mock files are not malicious files, there is no risk if the files are accidentally executed.

6. Support

Encripto AS provides technical support according to the terms and conditions described at https://www.bt3.no/terms-conditions/

Blue Team Training Toolkit support page can be found at https://www.bt3.no/support/

7. 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 profiles using POST client requests and a "Keep-Alive" server response header, may cause errors in server responses. As a workaround, use "Keep-A1ive" as response header value.

- 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.

- Network protocol analyzers (e.g. Wireshark) can report network traffic inconsistencies when BT3's Pcapteller is used for replaying network traffic, and packet payload manipulation under non-optimal conditions is attempted.

  This can occur when existing PCAP payload data is replaced by smaller or bigger amounts of injected data. This situation has been observed with some protocols that use raw payloads (e.g. HTTP). As a workaround, you should inject data with the same size or length as the data that is about to be replaced during a traffic replay. In this case, the traffic replay will occur under optimal circumstances.

Feel free to contact support@bt3.no for feedback, bug reports or feature requests.