jamk.fi

Suitability of commercial product for Cyber Red Team operations

Jukka Kuusela

Master's thesis
May 2016
Technology
Degree Programme in Cyber Security

jamk.fi

Kuvailulehti

Tekijä(t)	Julkaisun laji	Päivämäärä
Kuusela, Jukka Tapani	Opinnäytetyö, ylempi AMK	18.5.2016
	Sivumäärä 73	Julkaisun kieli Suomi
		Verkkojulkaisulupa myönnetty: x

Työn nimi

Commercial products suitability for Cyber Red Team operations

Tutkinto-ohjelma

Degree Programme in Cyber Security

Työn ohjaaja(t)

Salmikangas, Esa, Hautamäki, Jari

Toimeksiantaja(t)

Puolustusvoimat

Tiivistelmä

Tarkoituksena oli tutkia ja arvioida kaupallisten tunkeutumis- ja uhkasimulointiohjelmistojen soveltuvuutta Red Team -toimintaan kybertoimintaympäristössä.

Käyttötarpeen, toimintatapojen ja myös historian selvittämiseksi työn alussa käytiin läpi Red Team -toimintaa yleisellä tasolla: mitä se on ja miksi sitä tarvitaan. Työn edetessä tarkentuivat kyberympäristöön liittyvän Red Team -toiminnan ominaispiirteet. Mitä on Red Team toiminta kyberympäristössä, mitkä ovat kyberoperaatioiden tyypilliset vaiheet Red Team operaatioissa ja mitä vaatimuksia nämä yhdessä asettavat käytettäville ohjelmille.

Varsinaisen arviointityön ensimmäisessä vaiheessa valittiin mukaan otettavat ohjelmat. Valitut ohjelmat valittiin ja rajattiin tarkasti, jotta työn kokonaistyömäärä pysyy halutuissa rajoissa. Arvioinnin seuraavassa vaiheessa luotiin kriteerit ohjelmien arviointiin. Kriteerit sisältävät halutut vaatimukset ja toiminnallisuudet testattavalle ohjelmalle. Kriteerit käytiin läpi ja hyväksyttiin testaukseen osallistuvien asiantuntijoiden kanssa. Varsinaisessa testausvaiheessa Puolustusvoimien asiantuntijat, testiryhmä, käyttivät arvioitavia ohjelmia sekä itsenäisesti että myös osana ryhmää. Kukin testaaja tuotti omat mielipiteensä kriteerien mukaisiin kohtiin. Testauksen lopuksi näiden pohjalta luotiin testauksen lopputulokset.

Testauksen tuloksena havaittiin, ettei yksikään arvioiduista ohjelmista täytä kaikkia kriteerien vaatimuksia. Arvoituja ohjelmia voidaan kyllä hyödyntää tehtävien operaatioiden yksittäisissä vaiheissa, kuten hyötykuorman toimittamisessa kohdeympäristöön.

Avainsanat (asiasanat)

Cobat Strike, Immunicty Canvas, Core Impact Pro, Metasploit Pro, Red Team, Kyber

Muut tiedot

jamk.fi

Description

Author(s)	Type of publication	Date
Kuusela, Jukka	Master's thesis	18.5.2016
		Language of publication: en
	Number of pages 73	Permission for web publication: x

Title of publication

Suitability of commercial products for Cyber Red Team operations

Degree programme

Master's degree programme in Cyber Security

Supervisor(s)

Salmikangas, Esa, Hautamäki, Jari

Assigned by

The Finnish Defence Forces

Abstract

The purpose of this thesis was to study, evaluate and estimate how well commercial penetration and Advanced Persistent Threat (ATP) simulation applications suit Cyber Red Team operations.

The beginning of the study clarifies the needs, ways of acting and the history of Red Teaming: what it is and why and when it is needed. Later on, more focus is set on the characteristics of Cyber Red Teaming. What are the typical steps in Cyber Red Team operations and what requirements these set to the used applications.

The actual evaluation started with a selection of applications to be included into the evaluation. The candidates were found by searching the Internet or availed by specialists working on The Finnish Defence Forces. The selection was done carefully, because the purpose was to keep the overall workload reasonable.

The next step was to create criteria for the application evaluation. This included the functional and non-functional requirements for Cyber Red Team applications. The criteria were checked and approved by specialists taking part in the evaluation. During the evaluation, the team members used applications as individuals and also as a part of a team. Each member produced their own opinions and notes of the individual requirements on the criteria. The final results were collected and combined from these individual notes.

The final result was that none of the tested applications can fulfill all the requirements of Cyber Red Teaming. However, each application can be useful in individual steps of Cyber Red Team operation. These can be, for example, exploit development or payload delivery to the target environment.

Keywords/tags (subjects)

Cobat Strike, Immunicty Canvas, Core Impact Pro, Metasploit Pro, Red Team, Cyber

Miscellaneous

Acronyms

Term	Explanation
API	Application Programming Interface
APT	Advanced Persistent Threat
ARP	Address Resolution Protocol
AV	Anti-Virus
CCDCEO	Cooperative Cyber Defence Centre of Excellence
CRT	Cyber Red Team
CVE	Common Vulnerabilities and Exposures
C2	Command and Control
DCE	Distributed Computing Environment
DLL	Dynamic Link Library
DNS	Domain Name System
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection Systems
IOC	Indicator of compromise
IPC	Integrated Pentest Environment
LAN	Local Area Network
LSASS	Local Security Authority Subsystem Service
MTTC	Mean Time to Compromise
MTTD	Mean-Time to Detect
MTTP	Mean Time to Privilege Escalation or “Pwnage”
MTTR	Mean-Time to Recovery
OSINT	Open Source Intelligence
PCAP	Packet Capture
PDF	Portable Document Format
RMI	Remote Method Invocation
RPC	Remote Procedure Call
RPT	Rapid Penetration Test
RTF	Rich Text Format
SCADA	Supervisory Control And Data Acquisition
SMB	Server Message Block
SOCS	Socket Secure
SQL	Structured Query Language
SSH	Secure Shell
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
WinRM	Windows Remote Management
WMI	Windows Management Instrumentation

Contents

Figures

Tables

1 Introduction

1.1 Research Objective

The main research objective was to study a functionality of commercial penetration testing and Advanced Persistent Threat (APT) simulation applications and estimate how well those suit to long-term Cyber Red Team operations. An important part of this study was to develop evaluation criteria that was used when selected applications were evaluated. A purpose of this criteria was to collect main features or requirements that should be provided by the Cyber Red Team application. The criteria also ensured, that all evaluated applications were handled equally.

1.2 Scope

The scope of this study was to evaluate commercial penetration testing and Advanced Persistent Threat (ATP) simulation applications and estimate how well those suit to long lasting Cyber Red Team operations. The first part of evaluation was to create criteria for the application evaluation. Criteria defines not only functional but also non-functional requirements that are important for an application used in Cyber Red Team operation. The study included also selection of applications that were included to evaluation. This was done carefully, because purpose was to keep overall workload reasonable. The actual evaluation of selected applications was done against evaluation criteria. Evaluation was also done together with other specialist in the Finnish Defence Forces. This required that test environments had to be made available for testing.

Based on this scope, some application types were omitted from this study. These included pure application vulnerability scanners, code analyzers and fuzzing tools. Also exploit development tools like compilers and debuggers, pure collaboration tools and non-commercial applications were also left out of evaluation. This kind of exclusion was done to make study work more compact and manageable.

1.3 Structure of the Thesis

The research is conducted using the following process:

Chapter 2. - Analysis of the theoretical base.

Chapter 3. - Analysis of the currently existing research, tools and methods.

Chapter 4. - Application evaluation criteria.

Chapter 5. - Application evaluation

Chapters 6. - Analysis of the results and conclusions.

Chapter two focuses on building the theoretical base. It aims to present what is Red Teaming and why and when it is used. It presents also what is Red Teaming in a context of cyber security and what are the typical steps in Cyber Red Team operations.

Chapter three presents used research tools, methods and description of test environments.

Chapter four contains created evaluation criteria. Evaluation criteria is divided in evaluation domains. Each domain has textual description and a list of key evaluation questions.

Chapter five contains application evaluation.

Chapter six contains result summary, conclusion and also presents the areas for further research.

2 Theoretical base

These chapters clarify what Red Teaming is, why it is needed and when it should be used, what is Red Teaming in a context of cyber security and what are usual steps in Cyber Red Team operations. These all help to understand requirements for Cyber Red Team applications.

2.1 Red Teaming

"If ignorant both of your enemy and yourself, you are certain to be in peril."

– Sun Tzu

Cognition is a mental process of knowing and it includes aspects such as awareness, perception, reasoning, and judgment. It also describes the acquisition, storage, retrieval and use of information or knowledge to achieve understanding, reasoning, meaning and learning. It develops through life, and the environment has a significant influence in it. Common experiences, interactions and culture affect the way persons think and may contribute to poor decisions. These biases and factors are important when assessing adversary's thinking and possible actions. Leaders and decision makers should seek out and challenge assumptions and use competing hypothesis rather than just seeking evidence to support a preferred theory. Also the amount of available information may be bigger than what brains can handle at a given time. Stress and pressure may lead decisions into a desired direction. Psychologists and analysts have proposed a variety of methods to counter the effect of these factors. These methods aim to broaden decision maker's mindset by considering more options and assessing them more objectively. Alternate analysis represents a set of methods to avoid pitfalls of poor decision making. Red Teaming is one method of alternative analysis and it can support decision making in almost any context. (Mateski, 2009, 2-10) (Decision-Making and Problem Solving: Human and Organizational Factors, 2011, 14-29)

U.S. Army Field Manual, The Operations Process defines Red Teaming as follows (The Operations Process, 2010, 19):

Red Teaming is a function that provides commanders an independent capability to fully explore alternative plans and operations in the context of the operational environment and from the perspective of partners, adversaries, and others. Red Teams assist the commander and staff with critical and creative thinking and help them avoid groupthink, mirror imaging, cultural missteps, and tunnel vision throughout the conduct of operations. Commanders use Red Teams to provide alternatives during planning, execution, and assessment to:

- *Broaden the understanding of the operational environment.*
- *Assist the commander and staff in framing problems and defining end state conditions.*
- *Challenge assumptions.*
- *Ensure the perspectives of the adversary and others are appropriately considered.*
- *Aid in identifying friendly and enemy vulnerabilities and opportunities.*
- *Assist in identifying areas for assessment.*
- *Anticipate cultural perceptions of partners, adversaries, and others.*
- *Conduct independent critical reviews and analyses of plans and concepts to identify potential weaknesses and vulnerabilities.*

The term Red Team comes from American military war gaming, where the blue team was traditionally the United States and, during the Cold War, the Red Team was the Soviet Union. Red Team is sometimes confused with Red Cell, which is intended to role-play the enemy in war-games in order to test plans and courses of actions. Red Team may also have this role; however, as a whole, Red Team's role is more significant than this. It should improve the overall effectiveness. (Mateski 2009, 31; Mulvaney, 2012, 2)

To be efficient, the Red Team should have persons with different skills, language, culture, economics and background to widen the view of the team. (Mulvaney, 2012, 3)

In 2005 the U.S. Army launched its University of Foreign Military and Cultural Studies (UFMCS). The purpose of the initiative is to "to provide the educational and training foundation to support the fielding of an Army-wide Red Team capability" and train army officers to "*look at problems differently; account for the perspective of the adversary, multinational partners, and others; and to frame alternative strategies.*". (Benson, 2007)

2.2 Cyber Red Teaming

Organizations must be prepared for the existing and the future security threats. Traditionally, organizations have focused on prevention of security breaches. A

strategy has been aimed at eliminating vulnerabilities and thereby mitigating security breaches before they happen. Threat and risk modelling, code analysis, vulnerability scanning have been tools reducing attack surface. Organizations have relied on security protections on network border with tools like firewalls, Intrusion Detection Systems (IDS) or email antivirus scanners. These methods do not eliminate risks totally; they just limit it.

IT environment is constantly changing. An organization's sensitive data can be found almost everywhere; private data centers, in the cloud, partner's premises or in employees' mobile devices. Cloud or hybrid cloud computing models also set requirements for security controls and processes. New technology inherently brings new risks and vulnerabilities. (Microsoft, 2014, 6)

In order to deal with the current sophisticated, highly targeted, constantly changing and advancing threat environment, organizations are forced to a more proactive approach to security issues. One way to test organizations' security controls and processes and increase security awareness is to use Cyber Red Teaming (CRT). Cyber Red Teams focus on threats from adversaries in the cyber space by following assumed actions, techniques, tactics and procedures of the attacker. The main goal of Cyber Red Teaming is to improve the overall security of the target organization by exposing vulnerabilities in the implemented security controls, strategies, postures, plans and concepts. Cyber Red Team operations are more technical in nature compared to the common concept of Red Teaming. Operations consists mainly of penetration testing techniques; however, the may also include unconventional tests and techniques as well as intrusion testing on physical facilities and real-life cyber-attacks in military context.

NATO Cooperative Cyber Defence Centre of Excellence (CCDCEO) "Cyber Red Teaming" document defines Cyber Red Teaming as:

"(…) an element that conducts vulnerability assessment in a realistic threat environment and with an adversarial point of view, on specified information systems, in order to enhance an organization's level of security."

(Brangetto, Çalışan, Rõigas, 2015, 11)

The article “Toward a Red Teaming Taxonomy 2.0” (Mateski, 2004) presents purposes and approaches for Red Team operations as illustrated in Table 1.

Table 1 The Red Teaming purposes and approaches (Mateski, 2004)

	Purpose	Functions	Examples
Passive	Understand	• Help BLUE better understand RED, BLUE, and how RED and BLUE view each other • Clarify BLUE assumptions and expose biases	• Various intelligence, military, and commercial planning efforts (implicit)
	Anticipate	• Anticipate possible RED courses of action • Avoid surprise • Better shape BLUEs courses of action	• Threat, risk, and vulnerability assessments (implicit) • The military decision making process (MDMP)
Active	Test	• Probe or penetrate BLUE systems or security • Identify and explore vulnerabilities • Explore and test RED courses of action and BLUE countermeasures interactively	• Penetration testing (physical and IT) • Some military exercises and experiments
	Train	• Teach BLUE how RED thinks and operates • Prepare BLUE to respond to possible RED courses of action	• National Training Center opposition force (OPFOR), Top Gun, etc. • TOPOFF exercises

Today, large or compelling organizations must assume that either a security breach has already occurred or it is a matter of time until it will. Red teaming exercises help organizations not only to strengthen current security controls and procedures but also **improves the detection** of the security incidents and **trains** security incident response strategies and processes. For example, Microsoft Corporation has “Assume Breach” security strategy in cloud services and it utilizes their own Red Team to strengthen threat detection, response and defense for its enterprise cloud services

(see Figure 1). The basic idea in this strategy is to assume that attackers have already exploited vulnerabilities and gained privileged access. (Microsoft, 2014, 7-8)

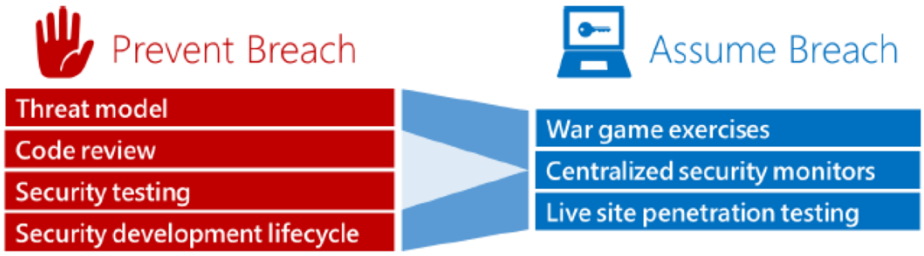


Figure 1 Microsoft Assume Breach strategy

2.3 Red Team operation and cyber kill chain

Advanced cyber-attack does not involve a single discrete event; rather it usually takes several steps to accomplish the purposed mission. These steps are usually shared among missions and are called cyber kill chain. The cyber kill chain contains the following steps (Gaining The Advantage, Applying Cyber Kill Chain Methodology to Network Defense, 2015, 3-10):

1. **Reconnaissance**. The first step contains active and passive operations to seek and collect information about a target. Passive operations include mainly usage of Open Source Intelligence (OSINT). Attacker can find plenty of public information about technologies used in a target and a list of persons and their roles for the social engineering attacks. Active operations contain, for example, online web page scanning or DNS queries. The purpose of this step is to identify and select targets for the next step, which consists of scanning the network, mapping the machines and Local Area Networks (LANs), listing the IP addresses of available machines, and finding out which services are up and available to attacker. If an attacker selects to use social engineering in attack, this step may produce a list of users and their roles in organization or a list of subcontractors' employees.

2. **Weaponization** is a step when an attacker obtains or develops a suitable exploit code for the target system's vulnerability. Deliverable payload code is then created by coupling exploit code with a selected shellcode.

3. **Delivery**. A specially crafted payload from the previous step is delivered into the target system. Possible delivery methods are, for example Microsoft Office document macro or another email attachment file. If the vulnerability is in a public service, e.g. in an organization's wiki pages, the attacker may exploit it with a direct online attack.

4. **Exploitation**. After successful delivery, the payload is executed in the target system, which may take place automatically in a target system or by actions of a careless user.

5. **Installation**. In this phase the shellcode is injected into a suitable process in the target system or a backdoor is created for the attacker. The exploited process or application may crash during exploitation, therefore the installation usually includes migration into another process.

6. **Command and control (C2) channel** gives attacker a communication channel to the target system, which is required in order to continue to operate in the target system remotely. The attacker sends commands to the system and receives command responses and downloaded data. The attacker may also send more malicious code into the target system. Communication may occur, for example over HTTP or DNS traffic. The intention is to mimic target network's normal traffic pattern and in this way keep out of sight. At the end of this process the attacker has gained an initial foothold in the target system.

7. **Actions on objectives**. In this step the attacker performs the actions to achieve his initial goals inside the target network. This phase may take months and require thousands of small steps planned in advance. These steps may include the following sub-steps:

- Attacker tries to make foothold **persistent**. The reached command and control channel should also be system boot resistant. This requires that the payload code must be stored into a disk and somehow executed during boot sequence or by user's assumed actions.
- **Privilege escalation**. Initial foothold may result in normal or local user account privileges. In order to continue the operation, the attacker most probably needs privileged account permissions. If the attacker is lucky, those may be easily available. System cache or configuration file may contain the

administrator's password in clear text or the attacker may steal administrator's Kerberos token from existing process.

- **Lateral movement**. In this step the attacker moves to other systems inside organization's network. The attacker may want to establish more C2 channels so that if some are lost there is still at least one to continue with. There might also be a need for data channel if the estimated data transfer amount is large enough. The initial goal may be, for example, to get access to classified data or defacement of an organization's web pages. Lateral movement takes the attacker towards initial target. The intent is to reach the most protected computers, servers and data.
- **Data Exfiltration**. This is a step when an attacker steals data from the target system and it requires tools for data discovery and good data channel.

After-action analysis. This is not a part of cyber kill chain; however, it is an important factor in Red Teaming. After-attack analysis and after-attack report give the target organization a possibility to calculate key metrics for security incident handling and a possibility to improve cyber defense and incident handling procedures. Microsoft calls this phase with a name "Red Team Breach Post-Mortem". (Microsoft Enterprise Cloud Red Teaming, 2014, 15)

By comparing Red Team actions to indications of compromise (IOC) reports and counter actions taken, it is possible to calculate the following metric values (Microsoft Enterprise Cloud Red Teaming, 2014, 15):

- **Mean-Time to Detect (MTTD)** refers to the amount of time it takes to detect a security incident.
- **Mean-Time to Recovery (MTTR)** refers to how long it takes to recover from a breach once detected.

In a similar way the Red Team can calculate key metric values for their own actions (Microsoft Enterprise Cloud Red Teaming, 2014, 12):

- **Mean Time to Compromise (MTTC)** measures the time from a start of the exercises to the point in time when the Red Team has successfully compromised an asset.

- **Mean Time to Privilege Escalation or "Pwnage" (MTTP)** is the time from the start of the exercise to full compromise of the target system.

3 Current research, tools and methods

3.1 Current research

The Cyber Red Teaming is a quite new way of measuring organizations' security controls. There are not many tools available, that are purely focused on this and no current scientific research was found about matter. There are some published articles, mainly from organizations using Cyber Red Teaming, but these publications handle issue in a general level. They do not present much details about tools used in Cyber Red Teaming or what are requirements for these tools. The best document found about Cyber Red Teaming is the one published by NATO Cooperative Cyber Defence Centre of Excellence (Brangetto, Rõigas, Çalışan, 2015). Also, Microsoft's own thoughts about network security and how Red Teaming can strengthen it is worth reading. (Microsoft Enterprise Cloud Red Teaming, 2014)

3.2 Tools and methods used

The First part of product evaluation was to select tools to be evaluated. Selection was based on publically available information either found from Internet or introduced by employees in The Finnish Defence Forces. Cyber Red Teaming has many similarities with penetration testing and most of programs are created for penetration testing. Only one program, Cobalt Strike, is built for Cyber Red Teaming. There are some free applications, like PowerShell Empire, available but there omitted from this study.

Second part of evaluation contained an evaluation criteria creation. Evaluation domains, main functional areas, were first identified and agreed. After this, more detailed evaluation questions were added. A purpose of these questions was to provide more specific requirements for applications to be evaluated. Criteria was accepted after it was checked and commented by specialists in The Finnish Defence Forces.

The last part of study contained an actual application evaluation. It was done by specialist in The Finnish Defence Forces and was based on:

- Material received from vendor. These ware mainly user guides or video material available in vendors' public web pages. These provides answers to some evaluation criteria questions. Provided material was also part of evaluation.
- Real tests done in virtual test environments. Team members used programs both themselves and as a part of team exercises. Evaluation team contained beginners as well as more advanced users. Each team member had evaluation criteria in hand and used that as a base for evaluation. Final evaluation results presented in this thesis, are combination of these individual opinions.

Tested applications were tested in different virtual environments. Both purchased and demo licenses provided by vendors were used. Core Security was not able to provide demo licenses in a provided timeframe but they provided own virtual environment for tests. Each virtual environment contained at least two hosts with applications to be tested so that features for team work could be evaluated. There were also target hosts for attacking and team servers or similar if needed.

Program versions tested were available in beginning of the year 2016. No main version updates were installed during test period.

4 Evaluation criteria

Following chapters define evaluation criteria domains presented in an evaluation criteria table (Appendix A, Evaluation Criteria).

4.1 Documentation and training material.

In military Cyber Red Team operations, some team members may not be familiar with the tools used. For this reason, it is important that there is some documentation and training material available so that those persons get familiar with a used tool.

These tools are usually complicated and full of features so even more advanced users may need reference material in order to check some functionality details.

The provided documentation can be either books or electronic documents, e.g. pdf or help files shipped with the product or online help pages on a company's web pages. The advantage with electronic documents is that they make keyword based searching easy. The problem with online help web pages is that the internet connection may not be always available during Cyber Red Team operation. Shipped documents, on the other hand, are always available.

Nowadays some tool vendors provide online training videos which can teach a product's basic usage for new users so that they get familiar with a product user interface and basic operations. More advanced users may find new features from online videos.

4.2 User interfaces

The Cyber Red Team application may provide different user interfaces for different usages. More advanced users may prefer command line like user interfaces while novice users rather work with a graphical user interface. A command line user interface may provide a possibility to use product specific batch files or operating system specific shell scripts. These files allow user to automate some actions and make it possible to repeat actions against multiple targets or multiple times against single target. Later on these files can be used as a part of attack campaign documentation.

Because cyber-attack operations follow the same steps (cyber kill chain), used tools usually share similar functionalities. Advanced users may change the used program during campaign or even use multiple programs simultaneously and for this reason the user interface should be intuitive so that it is easy to learn and change the tool used. User interface, either graphical or command line, should provide situational awareness information. These requirements are presented in chapter 4.16 Situation awareness.

4.3 Implementation

The used programming language is important because sometimes there is a need to check, change or extend the program's functionality. Some Red Team tools are rather development platforms designed to allow easy new feature development or integration of other security products.

Security tools are usually made by using Python or Java programming languages. The advantage of **Python** is that it works in different operation systems, and many of new exploits provide a proof of a context code written in Python. Python is a scripting language which offer an advantage to an attacker because it provides a layer of abstraction that an anti-virus protection may not know how to interpret. **Java** implementation can also be run in different operation systems. Java source code is typically compiled to Java bytecode. This bytecode can then be run in Java runtime environment in different platforms. It is also possible to compile to native machine code for a specific platform.

Some tools provide **PowerShell** attack modules for Windows environment. These modules typically provide tools for post exploitation actions. PowerShell is a task automation and remote system management platform for Windows environments and a scripting language for Windows operation systems. PowerShell offers a distinct advantage to attacker due to its tight integration with the Windows operation systems and a fact that it can be run entirely from memory. Antivirus evasion is much easier when there is no need to drop files on disk. Because it is meant to be used to automate Windows management task, it has also a rich set of features also benefitting attacker or penetration tester. Windows 8 and Windows Server 2012 or later Windows OS versions have PowerShell installed. (Using Windows PowerShell, 2013)

Modularity refers to the design of a system composed of separate, relative autonomous components that can be connected together. Modular applications are composed of smaller, separated code blocks that are well isolated and have own life cycles. Modular design is an advantage also in Cyber Red Team tools:

- Software structure and functionality is easier to learn if user wants to check how software works in a specific situation.
- Software functionality is easier to modify or extend.
- Some applications are designed so that 3th party modules can be easily added in to software. Exploits packs are the most typical examples of these modules.

4.4 Software updates

Red Team applications are under ongoing development. New features are created and new exploits or other new features are constantly added, which means that software must be updated often. If the software is difficult to update (for example, the software must be reinstalled), users do not do that so often. That is why the software should be easy to update. There should be also an option for offline updates if online update is not possible.

4.5 Database

Red Team tools usually utilize some sort of database to store configuration information and also intelligence information collected during attack campaign. This information must be a permanent storage if applications are closed during operations or crashed because of opponent's countermeasures. This information must also be shared between team members. New members may join the team, team members work in different shifts or operate in different locations. All members must have access to gathered information. This requires database replication between team members' application databases or a usage of centralized team server with own database.

4.6 Reconnaissance

Reconnaissance is the first step in the cyber kill chain. The attacker must first gather information about the attacked systems or users in the target organization. The purpose is to find a suitable system to be exploited or target persons for phishing mails. At least a part of the gathered information, like IP addresses, operation system

versions, services etc., must be stored into a database for future use. If external tools are used, there must be a way to import this information to tools database. It is the worst option to enter information manually.

Network scanning is an important step that may identify active hosts or free IP addresses on a target network. The scan may also identify operation system versions, open services on found hosts, service vendors and versions and authentication requirements. All this kind of information is vital for attacker, and thus, network scanning functionality should be a part of the Cyber Red Team application either built-in or the data can be imported from an external tool.

Open-Source Intelligence (OSINT) refers to any public, un-classified intelligence and includes anything freely available on the Web. OSINT helps an attacker to find information from available public data sources, e.g. an organizations' websites, forums, blogs, social networks, videos, and news sources. This information may reveal possible targets for social engineering campaign or used techniques inside target organization. OSINT mining tools are usually separate commercial or freeware applications; however, it should be possible to import this information to the Cyber Red Team application for future use and also for reporting purposes.

4.7 Exploit library

The target system exploitation requires some kind of vulnerability and specially developed exploit for that vulnerability. Exploit code execution, exploitation, makes it possible to execute selected payload code in the target. The attacker finds through network scanning or by using OSINT intelligence what operation systems and services are found from target networks. Based on that information, the attacker must be able to find a target specific exploit that fulfills other requirements like code execution option. It is important, that used Cyber Red Team application provides a comprehensive set of ready-made exploits, a possibility to use 3rd party exploits and a possibility to use custom exploits. Also, versatile search functionality must be available.

Windows servers, workstations and applications are the most common systems found in the target environment. There are still old operation systems available, at

least in cyber range environments, thus, exploits for all client Windows operation systems since Windows XP to Windows 10 should be available. Exploits for server operation systems since Windows 2000 Server to 2012 R2 server should also be available.

There is a huge amount of commercial and freeware **Windows applications** with versatile amount of software versions. These applications provide their own vulnerabilities. The Cyber Red Team application should provide exploits for Windows applications and provide a possibility to use custom made exploits.

Linux and Unix exploits should be available for popular Linux/Unix distributions, Linux/Unix base components and Linux/Unix services.

The market share of **OS X** in client operation systems is small compared to one of Windows; however, OS X computers may still be important targets in Cyber Red Team operations and exercises and thus, there should be ready made exploits for OS X computers or possibility to add ones into Cyber Red Team application.

A modern society uses a heavily **embedded system** which controls (Securing SCADA Infrastructure, 2010, 2):

- Industrial processes like manufacturing, production and power generation.
- Infrastructure processes like water works and water distribution, wastewater collection and treatment, power transmission and distribution.
- Facility processes in public and private facilities.

For this reason, Cyber Red Team application should contain exploits at least for the most popular embedded platforms.

4.8 Privilege escalation

Privileges define what a user can do in a system. After initial foothold, the attacker may have normal user's privileges and may require other privileges to continue operation. In vertical privilege escalation, the attacker gains himself higher privileges. In horizontal privilege escalation, the attacker assumes the identity of another user with similar privileges.

There are a number of privilege escalation methods. Clear text password may be recoverable from Windows LSASS memory dump or available in configuration file in a shared folder. Administrator password or password hash may be possible to crack by doing dictionary or brute-force attack. The attacker may steal a password hash from running process and use the pass-the-hash technique or use DLL preloading vulnerability in one of the applications in the target system. The Cyber Red Team application should have support for various privilege escalation methods.

4.9 Data exfiltration

Data exfiltration is the unauthorized transfer of data from a computer or system. It is sometimes referred to as data extrusion, data exportation, or data theft. It can be a manual operation carried out by someone with physical access to a computer or it may be automated and carried out through command and control channel by malicious code running on the target system. The Cyber Red Team application should have tools for data exfiltration. These may include, for example, password harvesting tools, key loggers, screenshot tools and tools allowing data mining from databases. It is good if the tool used allows addition of custom exfiltration modules.

Sometimes the attacker must transfer large files, and thus it must be possible to download files from an exploited host, which also sets requirements for used command and control channel (Chapter 4.12 Payload)

4.10 Persistence

For the attacker to maintain a foothold inside the target network, the malware needs to be installed persistently. This means that it will be reactivated in the case of a host reboot or user logon. There are many ways to do persistence. Methods used in Microsoft Windows platforms are the use the system registry, scheduled tasks, Windows Management Instrumentation (WMI) event subscriptions or use of custom Windows services. In Linux or Unix systems, persistence may be based on run command (rc) and stored executable files. The Cyber Red Team application should support persistence at least in Windows operation systems.

4.11 Lateral movement

After the initial foothold, the attacker may start moving into other computers in the target organization. The primary target may be organization's domain controller, data storage or code repository. The techniques used in this lateral movement are for example:

- Attacker uses revealed username and password to open SSH connections to other hosts in organization.
- Attacker uses stolen password hash to access shared disks in other computers.
- Attacker uses remove management frameworks like WMI or RMI for lateral movement.

The tool should contain tools for lateral movement.

4.12 Payload

Payload refers to the part of malware code delivered to the target computer and which performs malicious actions. The payload may open Control and Communication (C2) channel back to the attacker which allows interactive communication between the attacker and target system.

Control and Communication channel enables the attacker to send commands to remote payload and also get answers and requested data back from payload. Each payload may have different roles in the payload communication. One payload may pivot traffic between the attacker and the other payloads inside same network. Some other payload may be used to keep persistence and communicate only once a day.

When one Red Team member needs to carry out actions in one of the target hosts, he may ask C2 channel from other team member or he may command persistence payload to spawn a new session for him/her; therefore, the Red Team members must be able to share the current C2 channels and also the spawn new C2 channels.

The payload communication should mimic normal network traffic and must be able to pass target organizations security controls like firewall and security proxies. One

communication channel may not work in a different organization; thus, the payload should support multiple communications channels. These use for example HTTP/HTTPS protocol, DNS protocol or IRC protocol. There can be their own protocol, like SMB that is used for communication between payloads. It should be possible to change the used protocol during operation. It should also be possible to define communication interval and working hours.

4.13 Evasion

Because most organizations nowadays use host based antivirus software and at least network based IDS product, the attacker must try to hide attack by using different evasion methods. There are many different evasion methods most of which try to get past the signature based IDS/AV system. **Obfuscation** is the process where transmitted data is manipulated in such a way that the signature will no longer match but the receiving device with still interpret it properly. Obfuscation may be done just by using different character encoding or adding extraneous harmless characters. **Fragmentation** is a method where the sent data is broken into multiple packets and these packets are possible sent in the wrong order. In **encryption** the data is encrypted before sent to the receiving end. A detecting system cannot decrypt traffic without encryption key and so the signature cannot match. Encryption is a powerful evasion method because each encryption key produces a different signature.

The tool should contain built-in support for evasion methods or it should be possible to use external evasion tools like Veil Framework.

4.14 Compatibility with other tools

In long lasting Cyber Red Team operations, the team may use different tools in different operation phases or purposes. For example:

- External tool is needed for some attack phases and C2 session must be passed to external tool.
- Evasion features of external application are used to modify payload of RT application.

- Gathered password hashes must be sent to password cracking software.
- Initial exploit must be done with the external tool and the payload must be generated and exported from the used tool.

Cyber Red Team applications should provide data import/export functionality for gathered intelligence. It should also provide a way to pass sessions at least to Metasploit Framework and a way to export payload in different format (raw binary, Python, Java, hex encoded …). It is also important that it has a possibility to use payloads exported from external tools.

Red Team may use different tools to gather information from the target, which may contain information gathered with open source intelligence tools, network scanning tools, manually gathered information, information received from third parties etc. In order to make this information available and easily usable to team members, it must be imported into used Cyber Red Team application.

4.15 Teamwork

Red Team operations are, like name says, operations that are done in a team. The team members may change during the operation and the team members may be situated in different locations. In a major operation there may even be more than one Red Team working in different shifts. For this reason, the tools for teamwork are needed.

Communication is important in Red Teaming because it is supposed to be controlled and organized. The team leaders must be able to communicate with team members so that operation instructions and commands can be delivered. Also, the team members must be able to share information and opinions during Red Team operations. Some actions may require strict synchronization between members and this can be achieved with communication messages.

4.16 Situation awareness

Red Team members must share be able to get and share the same situation awareness information. Information shared among members is more technical containing detailed information of vulnerabilities, targets, passwords etc. Situation

awareness information must also be presented to superiors in an organization. The higher in the organization the viewer is, the more generic information must be presented. There might be separate management systems, at least in military organizations, and it may be necessary to transfer high level situation awareness information automatically to the management system.

4.17 Reporting options

Reporting is the last and very important step of Red Team operations. The report should contain:

- Evidence of penetration to different systems
- List of gathered intelligence like systems in target network, usernames and passwords, etc.
- Found vulnerabilities.
- Timeline of taken actions.

4.18 Evaluation criteria table

Appendix A illustrates the evaluation criteria in a table format. It contains a collected list of evaluation criteria questions for each of domains described in the previous chapters. This table was used during a product evaluation as a reference material to ensure that products were equally evaluated and that each evaluation team member used similar evaluation criteria.

5 Product evaluations

5.1 Cobalt Strike 3.2

Cobalt Strike is a software for adversary simulations in a Cyber Red Team operations. It gives a post-exploitation agent and covert channels to emulate a quiet, long-term embedded actor in a target organization’s network. It provides tools for social engineering, collaboration capabilities, post exploitation tools and after-action reports.

Cobalt Strike uses a client-server model. The server component, team server, is a listener and a controller for the beacon payloads and it also hosts Cobalt Strike's social engineering attacks. The team server also stores collected data and manages logging. (Mudge, 2016, 8)

Cobalt Strike is used in many Cyber Red Team exercises like Locked Shields which is real-time network defence exercise organized by the NATO Cooperative Cyber Defence Centre of Excellence (CCDCEO).

The first versions of Cobalt Strike were tightly integrated with the Metasploit Framework. Online attacks available through Cobalt Strike's module browser were from Metasploit Framework. Also Nmap -tool integration was provided so that network scan results were automatically imported into Cobalt Strike database. Starting from version 3.0, integration with Metasploit Framework and Nmap was removed.

Cobalt Strike is created by Raphael Mudge from a Strategic Cyber LLC. Tested version was Cobalt Strike 3.2.

CR-01, Documentation. Product manual, a PDF file, can be downloaded from product's home pages. It describes the basic application functionality and is a good reference for new users. Various Cobalt Strike features are presented in a separate online web pages. Many of those contain a video clip that explains the feature in question and shows how the feature is used. An aggressor script tutorial and a reference manual are also available as online web pages. Aggressor Script is the scripting language built into Cobalt Strike, and it allows users to modify and extend the Cobalt Strike client.

The best way to learn Cobalt Strike usage is to watch publicly available, free and good quality Advanced Threat Tactics video course. This course not only shows how Cobalt Strike is used, but it is also a good starting point to Cyber Red Team operations. In-house training is available from Strategic Cyber LLC and also from some third party companies.

CR-02, User interface. Cobalt Strike is used through graphical user interface. Command line user interface is not provided but Aggressor scripts can be executed

with separate Agscript program. This program is included only in the Cobalt Strike Linux package.

The main views are the beacon view and a target view. The beacon view shows the planted beacons in the target systems either in a data table or in a graphical representation. The graphical presentation shows the communication routes between beacons and between beacons and the team server (Figure 2).

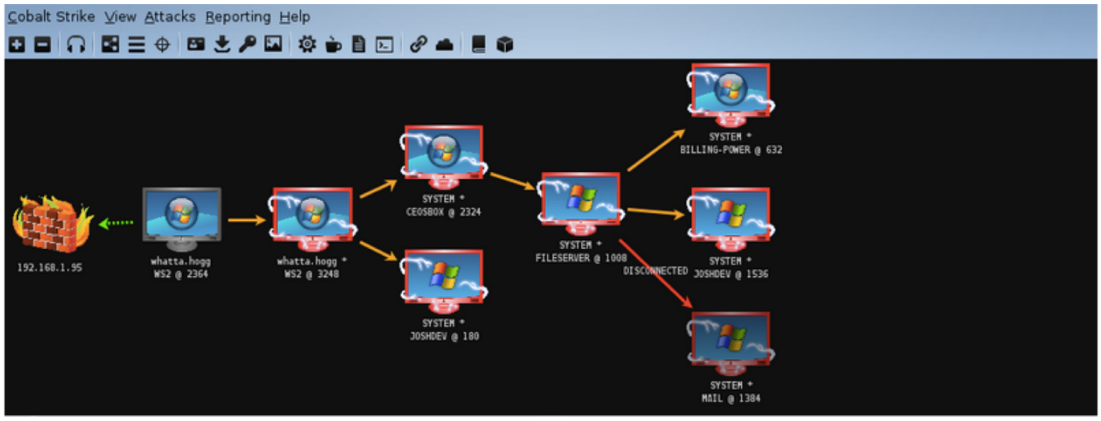


Figure 2 Cobalt Strike beacons

The beacon operation can be accessed through both views (Figure 3).

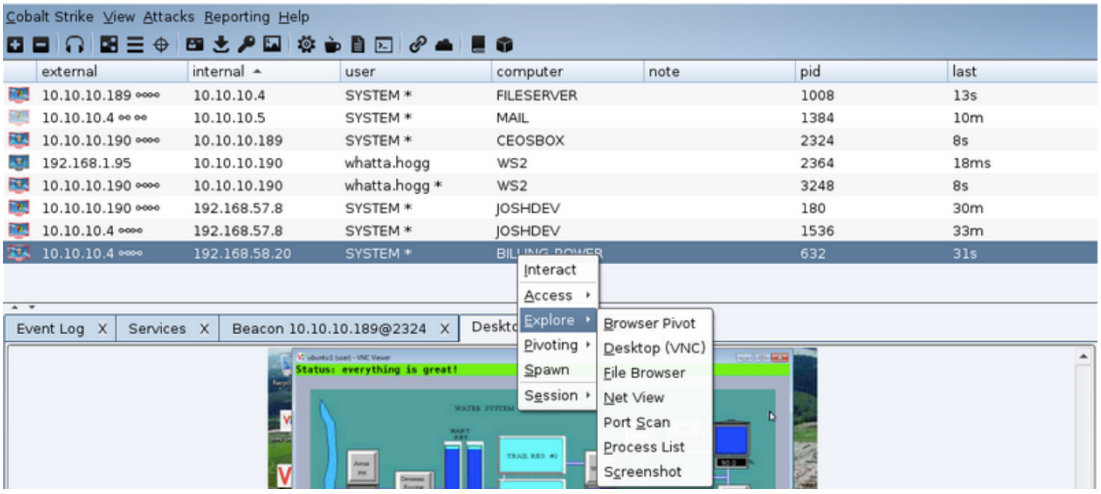


Figure 3 Operating beacon

The target view shows information of found and stored targets, i.e. servers and workstations.

The bottom of the GUI contains tabs for various purposes. These tabs include beacon consoles (command line shells for beacons), event and web logs, list of services in targets etc. The main menu and toolbar provide access to the main functionalities and information such as log files. Through the mouse menu it is possible to operate specific beacons and targets.

The user interface is logical and easy to use. After some use it is easy to find the desired functionality or information from menus or toolbar. The graphical view of current beacons is particularly good.

CR-03, Implementation. Cobalt Strike is a Java program and runs on Oracle's Java 1.7 or later. The supported environments for client are Windows 7, Mac OS X, Kali Linux and Ubuntu Linux. The supported environments for team server are Kali Linux and Ubuntu Linux. The user interface features and default reports are defined with Aggressor scripts. (Mudge, 2016, 7)

Cobalt Strike user interface and team server are located in one jar –file. It contains compiled Java classes, binaries and required Aggressor scripts. It is not supported or recommended to modify this package; however, it is possible to use Java decompiler and check the application functionality from the source code. Java source code is modular and quite easy to read.

Aggressor Script is a scripting language built into Cobalt Strike. It is based on Sleep scripting language and allows users to modify and extended the client. Most Cobalt Strike dialogs and features are written as stand-alone modules that expose some interface to the Aggressor Script engine. It is possible, for example, to integrate external tools like Nmap to Cobalt Strike or create custom data import or export operations. (Mudge, 2016, 11)

CR-04, System Updates. System updates are easy with a provided command line script. When executed, the script goes online, it checks that the license is valid and downloads and installs recent updates. Offline updates are possible manually because the update process replaces only cobaltstrike.jar file with a new one.

CR-05, Database. Cobalt Strike 3.0 does not include any SQL database. The persistent data is stored into Java serialized object files. These files can be read and modified with Aggressor scripts. Data model is introduced in Aggressor Script Tutorial and Reference.

CR-06, Reconnaissance. Cobalt Strike 3.0 contains a web system profiler for client-side attacks. This tool starts a local web-server and fingerprints anyone who visits it. The system profiler provides a list of applications and plugins it discovers through the user's browser.

After the initial foothold has been acquired, the beacon's port scanner can be used to discover new hosts from the target network. The port scanner provides two methods for scanning. The ARP method uses an ARP protocol request to discover hosts from the target network. The ICMP method sends an ICMP echo request to find alive hosts. In addition to the port scanner, the beacon's net module provides tools to interrogate and discover targets from a Windows active directory network. Beacon contains also a Net module. It implements some of the functionalities of net command available in windows operation system. Net module can also be used for reconnaissance. (Mudge, 2016, 47)

It is possible to create Aggressor scripts that import network or web application vulnerability scanning results into Cobalt Strike data files.

CR-07, Exploits. Cobalt Strike 3.0 does not provide any online exploits.

CR-08, Privilege escalation. Cobalt Strike 3.0 provides tools for privilege escalation; however, only for Windows hosts. The following methods are provided in beacon payload functionality (Mudge, 2016, 45-46):

- Elevate command, which utilizes CVE-2014-4113 vulnerability in win32k.sys kernel-mode drivers.
- Elevate with known credentials (username and password)
- User Account Control bypass privilege escalation technique takes advantage of a loophole in the UAC default.
- Mimikatz, which is integrated into the beacon, can be used to to recover plaintext passwords and password hashes for users who are logged on to the

current system. It can also be used to pull a password hash for a user from the domain controller.

CR-09, Data exfiltration. Only data exfiltration method provided is a file download functionality in beacon. Big data files can be downloaded in small chunks which helps to hide beacon network traffic. The size of the chunk depends on beacon's current data channel. Beacon works only in Windows environments. For other operation systems, beacon provides pivoting options (Mudge, 2016, 54-55):

- A SOCKS4a proxy server created on team server tunnels traffic into target network. Some tools, like Metasploit, can be configured to use SOCKS proxy. The Proxychains -tool can be used to force other tools to use a SOCKS proxy server.
- The rportfwd command setups a reverse pivot through beacon.
- A covert VPN creates a network interface on the Cobalt Strike system and bridges this interface into the target's network.

These pivoting options allow users to download files from the target network and to use external tools to harvest data from file system and databases.

CR-10, Persistence. Beacon does not have any built-in persistence mechanism so persistence must be done manually. A fully staged beacon can be exported into Windows service executable which is then uploaded and configured into target host.

CR-11, Lateral movement. **Beacon** has some built in options for lateral movement. Beacon's **psexec** command executes a payload on a remote host. Command generates a Windows service executable, copies it to the specified administrative file share (like c$ or ADMIN$), creates and starts a service, and cleans up after itself. Beacon's **psexec_psh** command executes a payload on a remote host with PowerShell. This command creates a service to run a PowerShell one-liner, starts it, and cleans up after itself. This method does not write anything to disk. Beacon's **winrm** command uses WinRM to execute a payload on a remote host. Beacon's **wmi** command delivers a payload via Windows Management Instrumentation (WMI). (Mudge, 2016, 48-49)

Beacon works only in Windows environments. In other environments beacons pivot options allows attacker to access other hosts in target network.

CR-12, Payload. Cobalt Strike's payload is called beacon and it is provided only for Windows environments.

Beacon's communication back to team server (C2 channel) can be implemented by using HTTP, HTTPS or DNS protocol. Beacons inside target network (peer-to-peer) can communicate through Windows named pipes. Beacon's communication is either asynchronous or interactive. Asynchronous communication is low and slow. Beacon calls home, downloads its tasks, and goes to sleep. Interactive communication happens in real-time.

Beacon console is a command line user interface for beacon management. It allows user to enter command for beacon to be executed and it also shows responses to these commands. Beacon menu contains the most often used commands for beacon management (Figure 4).

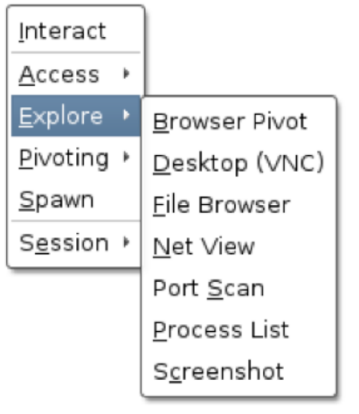


Figure 4 Beacon menu

One of the powerful features of the beacon is the possibility to import and execute PowerShell on the target host, which allows a user to extend the beacon's functionality. Another powerful feature is the session management which allows the user to spawn new C2 sessions to other team servers and also to external tools' listeners like Metasploit HTTP listener. It is also possible to inject the beacon to new processes and thus get new C2 sessions from other process.

Other beacon features (Mudge, 2016, 41-45):

- Keystroke logger captures keystrokes from a selected process.

- File upload and download commands allow user to copy files between the target and the team server.
- Screenshot command takes screenshot(s) from target host.
- Privilege escalation features described in criteria domain CR-08.
- Pivoting options that are described in criteria domain CR-09.
- Lateral movement support described in a criteria domain CR-11.

CR-13, Evasion. Cobalt Strike does not provide any built-in support for evasion. Beacon can be exported in different formats which allows external evasion tool usage.

CR-14, Compatibility with other tools. Cobalt Strike contains a support for Metasploit Framework. It is possible to pass a session from Cobalt Strike to Metasploit. To do this, the user can set up a foreign listener. These are aliases for x86 payload handlers running in the Metasploit Framework. User can also use a Metasploit Framework exploit to deliver a beacon. Cobalt Strike's beacon is compatible with the Metasploit Framework's staging protocol. User can tunnel Metasploit Framework exploits and modules through beacon by using a SOCKS proxy server. (Mudge, 2016, 54-55)

CR-15, Teamwork. Cobalt Strike supports a team work with many features. Cobalt Strike client contains a simple chat system. Team members connected to the same team server can send messages to each other. Messages can be either private or public. The private message is visible only to the specified target team member. The public message is visible to all members.

Because collected data is stored into a currently used team server, all users using the same team server share same information and can use current beacon sessions. A beacon can also be tasked to spawn a new session to other team servers. In this way one team member can for example use beacon session in a shared team server and ask beacon to spawn new session to the private team server.

CR-16, Situation awareness. Cobalt Strike users connected to a team server can see all data from the team server's data containers. User see for example found targets and beacon communications chains.

It is possible to export data from a team server and import that information into another system. Because data is stored into files, online database connection is not possible. However, it is possible to automate the data export functionality with an Aggressor script.

CR-17, Reporting. Cobalt Strike has several report options for after-action analysis. Following reports are available (Mudge, 2016, 67-70):

- The activity report provides a timeline of Red Team activities.
- The hosts report shows collected information on a host-by-host basis.
- Indicators of compromise (IOC) report shows various indicators of compromise including which domains were used and MD5 hashes for uploaded files. It also provides a "sample" of beacon traffic based on a used C2 profile.
- Sessions report shows indicators and activity on a session-by-session basis. This report includes the communication path each session, MD5 hashes of files put on disk during that session, miscellaneous indicators like service names, and a timeline of post-exploitation activity.
- The social engineering report shows spear phishing emails and user and data collection statistics. This report also shows applications discovered by the system profiler.
- Custom reports can be defined with an Aggressor script language.

A data for reports is collected from all of those team servers that user is currently connected. Reports are available as an MS Word or PDF document. Report details, such as a logo, a title, a description, and hosts displayed can be configured for in most reports.

5.2 Core Impact Pro 2015 R1

Core Impact Pro is a penetration testing software for assessing security controls and testing security vulnerabilities in an organization. It allows user to test a wide range of targets including endpoint systems, mobile devices, wired and wireless network systems, web applications and web based services.

Core Impact Pro not a Cyber Red Teaming software but it includes some features for a teamwork. Intention in product usage is to use the same techniques that are employed by today's cyber-criminals.

Because of export restrictions and a strict time schedule, Core Security was not able to provide any demo licenses for testing. Instead they provided ready virtual environment with two client hosts running Core Impact Pro and some target hosts for attacks. This set some limits for the product evaluation. For example, it was not possible to monitor network traffic and check what kind of indicators agent (Core Impact Pro's payload) communication provides.

Core Impact Pro is a product from Core Security Technologies.

CR-01, Documentation. Documentation provided for Core Impact Pro is fairly good. Almost 400 pages long user guide is provided as a pdf file. It goes through all usual features for program use starting from program installation and ending to a chapter describing the underlying architecture and technologies used in the application implementation. Available is also a set of public videos. These demonstrate some basic features of program usage and even the quality of some vides is poor, these videos are good starting point for new users.

Core Security and third party companies provide online and onsite trainings for penetration testing and product usage. By taking part in trainings, it is possible to get and maintain Core Impact Certified Professional certificate.

CR-02, User interface. User interface for the product is good and logical. All basic tasks are easy to find and windows are arranged logically. One part of the user interface, which is unclear and confusing, is the entity view (see CR-16). Windows show currently running modules and also modules that have already been executed. Entity view shows gathered information based on selected domain; network entities, client side exploit entities or web application entities.

The Rapid Penetration Test (RPT) feature, presented in the version 4.0 of Core Impact, provides step-by-step automation of the penetration testing process. These packages run automatically through a predefined set of attack modules defined for each attack phase. For example, provided phases for network operations can be seen

in Figure 5. This kind of automation helps penetration testers but is not useful in Cyber Red Teaming campaigns. These are not mandatory for the program usage because user can execute individual modules one-by-one.

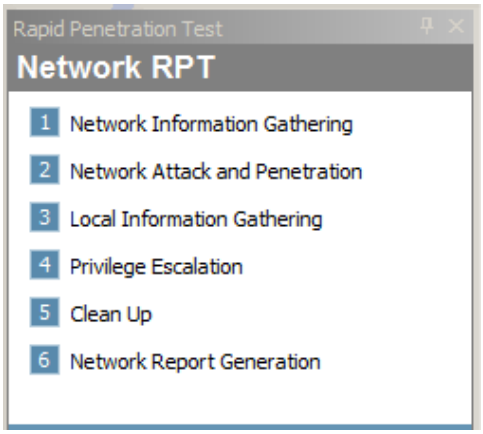


Figure 5 Network RPT wizards

CR-03, Implementation. Core Impact Pro implementation is modular. Individual modules are implemented by using Ruby, Python and C# programing languages. User can extend program usage with own custom Python modules. User interface contains a wizard for module creation making module integration easier of user.

CR-04, System Updates. System updates are easy to do. The user interface will display an alert when there is a new software release available. In this case, a user can download and install the latest version directly from user interface. User can also update only the latest module updates either online from the update server or offline by using a IMP file. (Core Impact Pro, User Guide, 2015, 41)

CR-05, Database. Core Impact Pro uses a Microsoft SQL server which is a relational database server. The database schema is not available but it's is possible to find out directly from database.

CR-06, Reconnaissance. The Core Impact Pro application provides a rich set of reconnaissance features. The easiest way to do reconnaissance is to use Rapid Penetration Test wizards. The **Network Information Gathering wizard** uses host discovery, port scanning, OS identification and service identification modules to

gather information from target network. The **Local Information Gathering wizard** can be used to get more precise information from each host. Module usage requires that the user has an access (agent) to target hosts. Information gathered includes network information, email addresses, credentials, installed applications and more precise operating system version. The **Client-side Information Gathering wizard** harvests email address from the Internet or from the specified target host, typically target organization's intranet service. Executed modules will also look for downloadable documents and search within them for email addresses and sensitive data such as social security and credit card numbers. The **Web Applications Information Gathering wizard** scans the domain of a known web-based application and identifies pages or web services that may be vulnerable to potential attacks. The **Access Point Discovery wizard** is reports any detectable wireless networks as well as any devices connected to them. (Core Impact Pro, User Guide, 2015, 52-106)

These wizards may not suit for the Cyber Red Team usage as such but it is possible to execute required modules one-by-one manually. If user prefers to use external reconnaissance tools, Core Impact Pro provides a set of data import modules. With these, user can import data files created by popular network and vulnerability scanners like Nmap, Nessus, Nexpose, SAINT or Retina. All gathered information is stored into the SQL database. If a teaming workspace is in use, team members are able to share this data. This is a very essential feature in Cyber Red Team operations.

CR-07, Exploits. Core Impact Pro provides a comprehensive exploit library. Available are exploits for all Windows versions, AIX, FreeBSD, Linux, OS X, OpenBSD and Solaris operation systems. There are also exploits for various applications, network devices, surveillance cameras, Web applications, authentication vulnerabilities.

Exploits are available in Python code. If needed, the user can modify these exploits or create and install new custom exploits.

CR-08, Privilege escalation. The Privilege Escalation RPT step executes local privilege escalation attacks on connected agents not running as the super user or the administrator. Again, Red Team member may not want to use automated and noisy operations, but rather use only selected modules. (Core Impact Pro, User Guide, 2015, 88-90)

The Core Impact Pro agent has also Mimikatz module integrated. This can be used to harvest usernames and plaintext passwords and password hashes from target hosts memory. (Core Impact Pro, User Guide, 2015, 302)

CR-09, Data exfiltration. Data exfiltration method available in Core Impact Pro is a file browse and download functionalities. User cannot change command and control channel features so file upload occurs as fast as is possible by used communication channel. If user wants to download file in smaller chunks, it must be done manually.

User can also create a VPN connection with the targeted host. With this tunnel in place, user can then run 3rd party tools from the local system and have them interface with the host. This feature requires that pcap –plug-in has been installed on the agent. This plug-in on agent enables a faster scanning and an add support for packet capture and packet injection to a pivoted agent. TCP Proxy plugin allows user to create tunnel from client host to agent host. This makes it possible to use external tools through tunnel. (Core Impact Pro, User Guide, 2015, 301)

CR-10, Persistence. The agent menu in the user interface contains an option to make agent persistent. When executed, an agent is stored in the filesystem of the compromised host so that it can be used across system. Persistence methods provided for Windows hosts are Windows executable file that is started from created service. In Linux hosts, persistence is created with an executable file which is started from run command (rc) files.

Core Impact Pro does not support staging process. Thus, in case of persistence, whole agent is written in the disk instead of smaller staging executable code.

CR-11, Lateral movement. In addition to the exploitation process normally used to get initial foothold, Core Impact Pro can use a valid username and password to deploy an agent on a remote host. There are variety of different protocols provided by utility modules that can be used for doing this (Core Impact Pro, User Guide, 2015, 312):

- SMB: Installs an agent by connecting to a network share.
- WMI: Installs an agent using Windows Management Instrumentation.
- SSH: Installs an agent by connecting through SSH.

- Rlogin: Installs an agent by connecting through rlogin.
- Telnet: Installs an agent using the telnet service.
- Unix-portshell: Installs an agent on a Unix target using a shell bound to a port.
- VNC Protocol: Installs an agent using the VNC protocol.
- Win-portshell: Installs an agent on a Windows target using a shell bound to a port.

These modules provide good selection of options for lateral movement.

CR-12, Payload. The payload in Core Impact Pro is called an agent. There are good variety of different agents for different target platforms. The most common operation systems are supported. These include Windows systems since Windows 7, popular Linux and Unix distributions and Mac OS X versions. There are also agents for Android mobile devices and IOS network devices. Web applications agents represent the information of how to exploit a web application vulnerability. User can use these agents to perform activities, like SQL Injection or Cross Site Scripting, on the web application's server.

Agent's command and control channel is defined when agent is deployed to the target. Possible protocols for the agent communication are TCP, reverse TCP, HTTP or HTTPS. Online attacks have also options to reuse the same TCP connection that was used to deliver the attack. Defined channel properties, like used protocol, cannot be changed after channel has been created. Only exceptions are session keep-alive settings. It is not possible to define working hours or transmission interval. Pivoting (chaining) features are good. User can pivot traffic through any deployed agents. These pivot chains can be as long as is needed.

Agent is operated from user interface. Agent menu offers functions to be performed (Figure 6). Menu content varies according current agent; the android agent contains commands relevant to mobile phone whereas Windows or Linux agents provide commands relevant to servers or workstations.

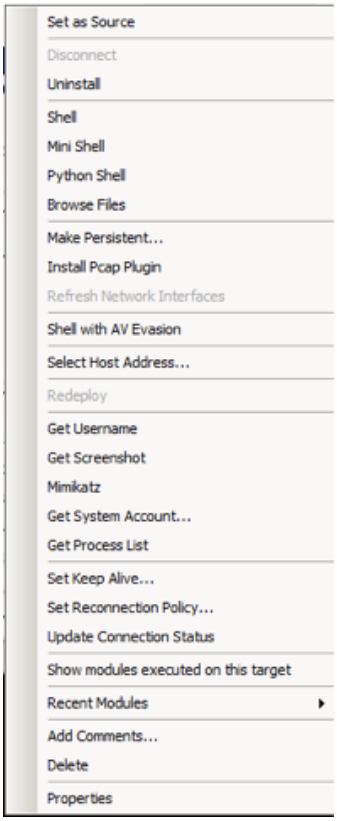


Figure 6 Agent menu for Windows agent

Agents can use plug-ins to add functionality to a deployed agent. The following plug-ins are available in product (Core Impact Pro, User Guide, 2015, 316):

- PCAP plug-in provides packet-capture capabilities for the agent.
- TCP Proxy allows to create TCP tunnels from console to the agent.
- HTTP Proxy over TCP Proxy allows to browse a web server that is visible from the agent's host machine.

CR-13, Evasion. Core Impact Pro has some built-in evasion features. From agent modules, it is possible to select a shell with antivirus evasion features. This shell contains antivirus evasion qualities that will reduce the chances that it will be detected by the host machine's antivirus processes.

When using online attacks, it is possible to select evasion options. These are accomplished via encryption, fragmentation, and/or limiting the number of packets that will be sent as part of an exploit over a specific period of time.

CR-14. Compatibility with other tools. Core Impact Pro supports multiple products that may be needed during the attack campaign. Metasploit Framework can be integrated to product so that user, when executing RPT attacks, can select to include Metasploit Framework exploits to the attack. In this case Core Impact Pro will select which exploits from the Metasploit Framework to run. User can also install a Core Impact Pro agent from a Metasploit console. This requires that user has already Meterpreter session to the target host. Requirement for Meterpreter is troublesome because it is easily detected by antivirus products.

Core Impact Pro has a great data import support for various penetration testing tools. It is possible to import data many of the most common vulnerability scanners like Nessus, Nexpose, Acunetix, Nmap, Retina or Saint. Imported data appears to current workspace and is available to all joined members. (Core Impact Pro, User Guide, 2015, 346)

Agents' TCP Proxy plug-in allows to use external tools through agents C2 channel.

CR-15. Teamwork. Core Impact provides sufficient teamwork support. Provided Teaming features allows more than one user to collaborate on a single workspace giving team members the ability to share data and delegate tasks. To user teaming feature, at least one teaming session must be created. Then, team members can join that session and use the shared the workspace. All joined users will see discovered entities, planted agents, executed modules and their output. In addition, with entity assignment feature team members can assign entities to specific users. Entity filtering feature shows targets that have been assigned to current user. (Core Impact Pro, User Guide, 2015, 213-216)

CR-16, Situation awareness. Core Impact Pro does not provide satisfactory situational awareness information. The entity view of the user interface shows among other things discovered entities from target network and possible agents running in those hosts. If there are lots of hosts, it is hard to find, for example, what is current situation with agents. What agents are still alive and in which hosts we have agent running. To view how each agent communicates back to home, user must use the agent chaining route module. An example entity view can be seen in Figure 7.

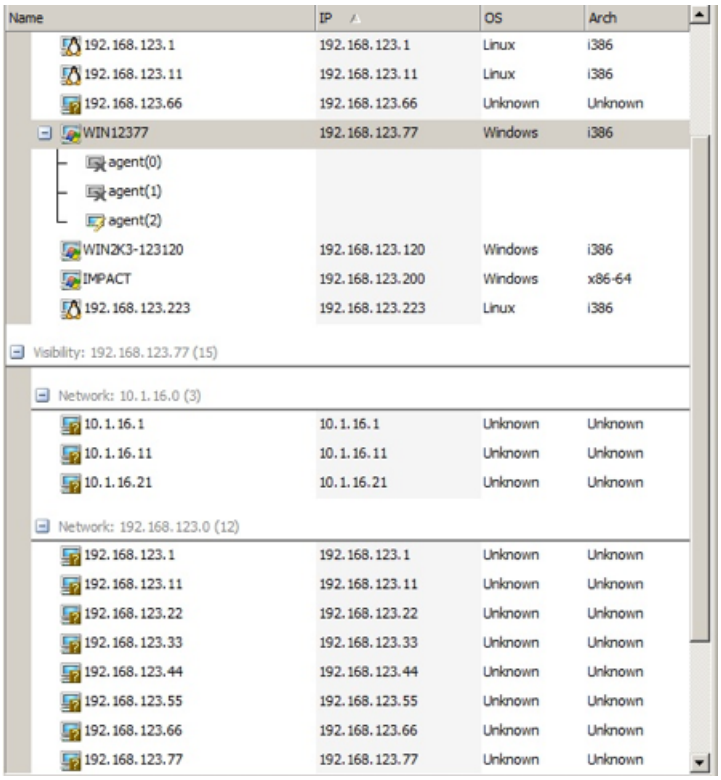


Figure 7 Host entity view with one active agent

CR-17, Reporting. Core Impact Pro provides an excellent and comprehensive reporting features. Each of the Rapid Penetration Tests provides a set of reports. Most reports are made by using Crystal Reports but there are also some reports that use the Microsoft Excel as the reporting engine. Most reports have options that allows user to customize reports layout or other features. Microsoft Excel spreadsheet reports can be further modified and customized if default versions are not suitable. Some of the offered reports are (Core Impact Pro, User Guide, 182-189):

- Host Report reports detected hosts and discovered details of each hosts.
- Host Based Activity Report shows all modules run for each detected host.
- Executive Report shows completed penetration test activities and their results.
- Activity Report is a detailed report of all modules executed
- Attack Graph Report which shows successful attack vectors in a form of graphical representation (Figure 8).

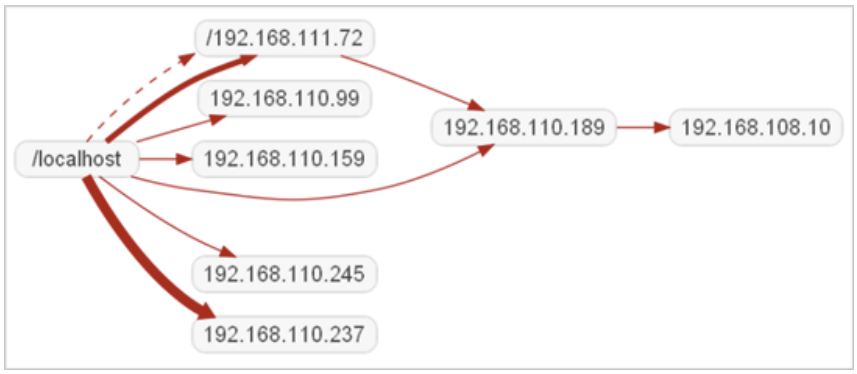


Figure 8 Attack Graph Report

5.3 Immunity Canvas 7.09

Immunity Canvas is a defence and a penetration testing software Immunity Inc. The software architecture is open and flexible by design and software is more an exploit development platform rather than as a full-featured penetration-testing application. It includes hundreds of online exploits and uses MOSDEF, a custom C compiler, for payload construction.

CR-01, Documentation. Canvas documentation is delivered in the form of demonstration movies. There are only some short pdf tutorial documents that describe basic Canvas usage. There are lots of features that are not explained either in tutorial documents or video tutorials so many features must be learned just by testing or by investigating from supplied source code. Most exploit modules have exploit specific information.

Software vendor provides two day virtual online trainings that covers basic application usage.

CR-02, User interface. User interface included in Canvas is illogical and disorganized. For example, “Current Target(s)” control in toolbar is used by some modules but some other modules use selected nodes in Node Management view. The user interface contains a command line view for a node and listener management. This view does not have a text buffer which makes it difficult to read longer text outputs. Poor user interface makes Canvas a little bit harder to use, at least for new users. However, the user interface provides access to required functions. User can browse,

search and executed exploit or utility modules and execution status can be followed from status and log views. Planted payloads, nodes, are shown in a node management view (Figure 9). Information gathered through each node is available through a node's mouse menu.

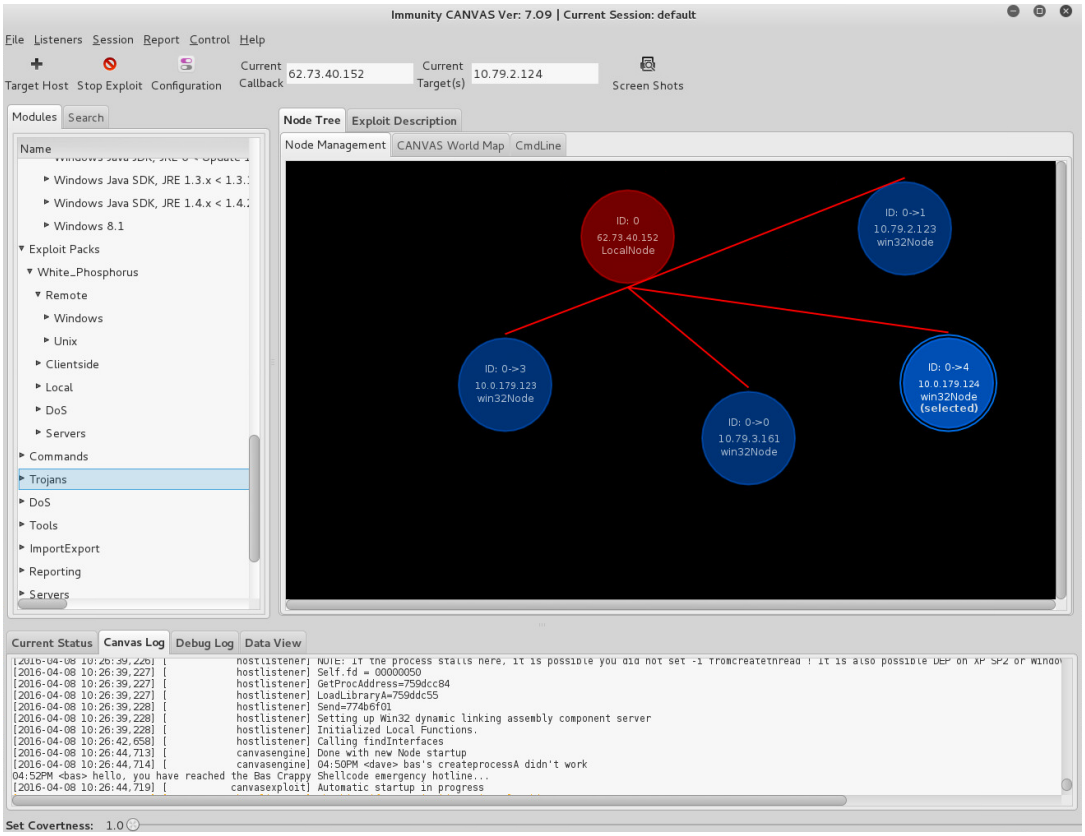


Figure 9 Immunity Canvas user interface

Listener shell provides common actions, like file management or screenshot, with target host (Figure 10). Again, it takes little time to find out how this works. For example, text control in top of is used in many purposes.

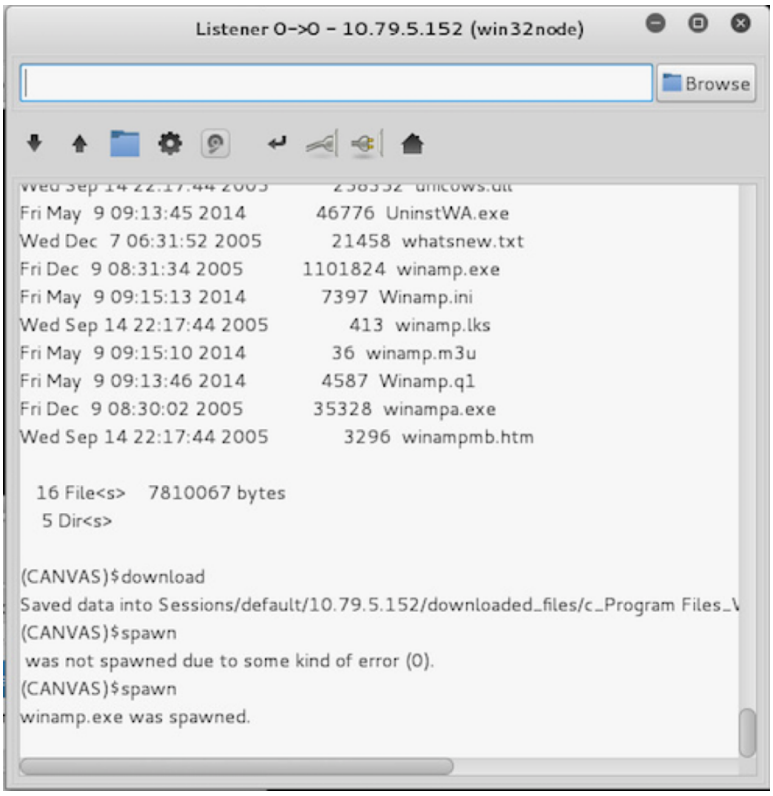


Figure 10 Listener shell

Canvas World Map view reconciles IP addresses with their geographic locations. This feature requires third party GeoIP libraries and was not tested because all tests were done in virtual environment. Example Word Map view can be seen in Figure 11. (Immunity CANVAS, 2016)

Figure 11 Node geographic locations

Because all modules are implemented on Python, these can also be executed from command line. This allows users to create shell or Python scripts to automate and repeat some attack phases.

CR-03, Implementation. Canvas is built on Python and the user interface is built on top of pyGTK. An architecture is fully open meaning that a full source code is included in the program delivery. Canvas payloads utilize MOSDEF which is a custom C compiler that supports dynamic remote code linking written in Python. It allows users to inject code in exploited processes and thus extend nodes' functionality. MOSDEF also allows attacker to stay on memory without touching the disk. (McGeorge, 2013)

It is possible to extend or change the user interface functionality by modifying provided source code.

CR-04, System Updates. Software updates are easy to make online because the user interface provides an option for this. Offline update functionality is not available but can be done manually just by copying all files from updated version.

CR-05, Database. Canvas does not utilize any database. User's session data can be manually stored into clear-text files and can then restored when needed. Tests showed that the restore operation does not always work as expected. Sometimes a part of the session data was lost. This stored session data does neither contain all discovered data. When user executes modules, a module output is available only in a log view.

CR-06, Reconnaissance. When Canvas is launched, it starts automatically to sniff IP addresses from the connected network. This feature is not very useful in Cyber Red Team operations and can be disable from settings. There are over 40 reconnaissance modules available including modules for PING sweeping, ARP scanning and TCP/UDP port scanning. Also some OSINT modules, like DNS discovery, are available. Canvas does not have any tools for web application vulnerability scanning. Only option is to execute exploits against web application but this seldom is an option in Red Team operations.

It is possible to import data from external tools like Nmap, Nessus or Nexpose. The absence of database makes it harder to utilize imported data.

CR-07, Exploits. Tested version of Canvas contains over 800 exploits for Windows, Linux, Unix and OS X platforms and for web applications. In addition to these exploits, some 3rd party companies provide exploit packs for Canvas. These provide exploits to more specific targets like web, SCADA or healthcare systems. All exploits are available with a source code which allows malleability and extensibility.

Canvas is a good framework for creating custom exploits because it provides ready payload creation and protocol structures while leaving the full flexibility of a Python script. The easiest way to write custom modules is by modifying existing modules. To integrate it in the GUI, exploit must be just placed in correct subfolder. Also MOSDEF environment can be utilized in custom exploits.

User can search suitable exploits (modules) from search tab (Figure 12). Search can be targeted to specific data field in module metadata like Microsoft Security Bulletin number, module name or CVE name. More complicated searches must be done with regular expressions.

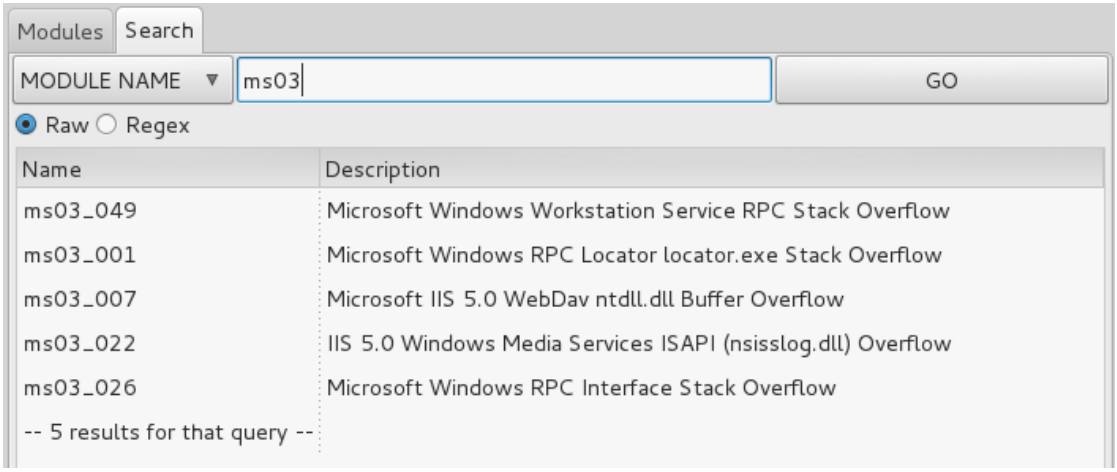


Figure 12 Module search tab

CR-08, Privilege escalation. Canvas provides over twenty modules for privilege escalation. These are available for Windows, Linux, Unix and Mac OS X hosts.

CR-09, Data exfiltration. Canvas provides manual file management functions through a listener shell. A disk spider module searches files from exploited host that match a file pattern and optionally downloads all matching files. Files are downloaded through used command and control as fast as used channel allows. User cannot change channel properties.

Canvas modules can dig useful data from the target environment. If target system is part of Windows domain, modules can find domain's hosts and usernames. Other modules search clear text password, password hashes or Kerberos tickets from memory.

TCP forwarding module allows to forward TCP connections through a MOSDEF node. Tunneling can either be from local host to remote host or reverse tunnel from remote host to local host. SSH reverse tunnel module create a reverse SSH tunnel between you and the target.

CR-10, Persistence. Only persistence module provided is WMI based persistence for Windows hosts.

CR-11, Lateral movement. The psexec module provides option to execute commands on another computer inside Windows network. This can be utilized during lateral movement phase. In addition to this, Canvas supports lateral movement through pivoting and TCP tunneling. Every planted agent, a node, can be used for pivoting. Together with online attacks allows user to move forward in target network. The wp_tcpforward module enables to forward TCP connections through MOSDEF nodes.

CR-12, Payload. Canvas supports multiple platforms including Windows, Linux, Solaris, OS X and Android. Each of these platforms have multiple shellcodes. These shellcodes may provide basic node functionality like a file management or screenshots while other shellcode enables user to upload and execute PowerShell cmdlets or execute PowerShell modules. Canvas payloads utilize MOSDEF which is a custom C compiler that supports dynamic remote code linking written in Python. It allows users to inject code in exploited processes and thus execute own code. MOSDEF also allows attacker to stay on memory without touching the disk. ()

C2 channels available are HTTP and DNS with an option to encrypt. User cannot change set working hours or other communication parameter for any C2 channels. Canvas includes Universal listener and platform specific listeners. Universal listener has been developed to accommodate all cases and work independently with whatever platform the user is working on.

CR-13, Evasion. Canvas does not provide any evasion modules. Instead, there is a covertness bar at the bottom of the user interface. User can use this control and set a covertness value. Higher the value is, more effort Canvas set to evasion. The user doesn't pick the evasion technique himself; it is selected automatically depending on the used protocol and setting of the covertness bar. The covertness feature is implemented in the protocol level and each protocol has different methods of evasion.

CR-14, Compatibility with other tools. Canvas provides import modules that allows user to import data from external tools. Evaluated version contained import modules for Nmap, Nessus, Nexpose and QualysGuard vulnerability scanners.

CR-15, Teamwork. Canvas does not provide any useful support for a teamwork. Only teamwork feature provided is a commander/operator mode which is provided by Canvas Strategic modules. When this mode is used, person working as a commander can check what is data is available in each operator’s Canvas desktop. So operator can figure out team’s current situation awareness information only by going through team member’s desktop and then combining all this information. Team members cannot share information except by doing it manually with a chat that is also provided by this mode.

CR-16, Situation awareness. Canvas provides decent situational awareness view but only for individual team members. A shared and combined team view is not possible to have.

A node management view shows current, planted nodes in target environment (plus a local node) and a command and control channel route to local node. Different node colors reveal exploited host’s types (Figure 13).

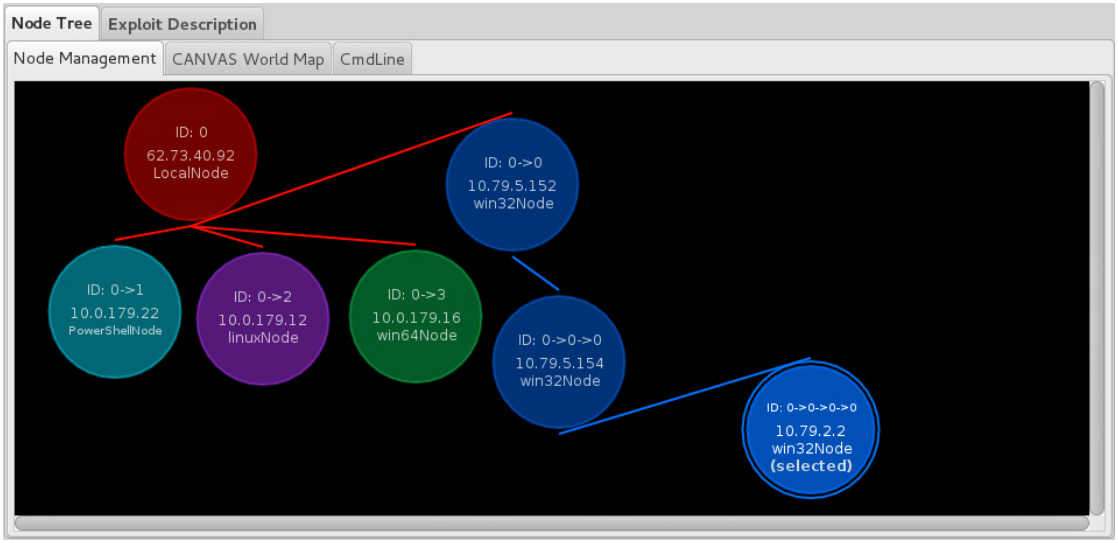


Figure 13 Node Management view

Each node has own data; information about the host in which the node is running and knowledge about target environment collected through that node. This data is accessed with mouse menu.

CR-17, Reporting. Canvas provides two reports; a canvas report and a clientd report. The first one is simple activity report showing mainly attempted exploits and successful attacks. The clientd report is more detailed client-side vulnerability report. Both reports are generated from session data read from stored text file.

Generated reports are OpenOffice documents. User can modify these documents and change for example organization or header information. For regular use, it is possible to modify report template files.

These reports are done mainly for penetration testers. A bigger problem with the Canvas reporting is that each report is user specific; built on user's session data. It is not possible to get combined report that collects session data from all team members.

5.4 Metasploit Pro

Metasploit Pro is a penetration testing platform that enables to find, exploit, and validate vulnerabilities. The platform includes the Metasploit Framework and its commercial counterparts like Metasploit Pro or Nexpose Ultimate. The Metasploit

Framework is the foundation on which the commercial products are built. It is an open source project that provides the infrastructure, content, and tools to perform penetration tests and security auditing. The commercial editions provide features that are unavailable in the Metasploit Framework like web-based user interface, automation of common penetration testing tasks and reporting.

Metasploit Pro is a product of Rapid7.

CR-01, Documentation. Documentation for Metasploit Pro is publicly available in Rapid7's web pages. Good quality documents include installation documents for different platforms, a user guide, and a getting started guide. There are also some special tutorials for issues like SSH key testing or passing the hash. In addition to these documents provided for Pro version, there are a vast amount of documents created for free community version. Thus information is available, but is scattered around which is often the situation with big and long open source projects.

CR-02, User interface. Metasploit Pro includes a web based user interface. It provides a simple to use automation for basic penetration testing tasks, such as scanning and exploiting targets, building social engineering attacks or phishing campaigns, scanning and exploiting web applications, and validating vulnerabilities. These wizard like tasks may be useful for penetration testers, but not very useful in Cyber Red Team operations. Nevertheless, the web interface also allows to search and execute individual modules.

For users, that prefer to operate on command line, the Metasploit Pro includes own, msfconsole like command line user interface, which gives an access to most of the features in Metasploit Pro. This console supports also resource files which allows to automate or re-execute individual attack steps.

CR-03, Implementation. Metasploit Pro is implemented on top of Metasploit Framework. Framework, in turn, is built by Ruby scripting language. Implementation is modular, each module extending the functionality of the Metasploit Framework. Module can be an exploit, auxiliary, payload, no operation payload (NOP), or post-exploitation module. The web user interface of Metasploit Pro utilizes Ruby on Rails, which is an open source web application framework also written in Ruby. (Metasploit Pro, User Guide, 2015, 3-4)

Because Metasploit Framework is implemented as an open source project, implementation quality is not constant. Some modules present very scrappy coding conventions.

CR-04, System Updates. New releases of Metasploit Pro can be checked and installed through Web user interface. It is also possible to download offline update file and use that to update installations without internet access.

CR-05, Database. Metasploit Pro uses PostgreSQL database, the one that is provided by a Metasploit Framework. Because Metasploit Framework is implemented in open source project, the database schema is publicly available. Custom implementations can utilize database and for example retrieve information directly from the database.

CR-06, Reconnaissance. Metasploit Pro provides an automated discovery scan, a Quick PenTest wizard (Figure 14). It uses Nmap for basic TCP port scanning and runs additional scanner modules to gather more information about the target hosts. Scan contains four steps. The first step is a ping scan to detect what hosts are online. After that, Metasploit Pro tries to identify the ports that are open and the services available on those ports. Third step is a version detection. Metasploit Pro sends a variety of probes to the open ports and detects the service version numbers and operating system based on how the system responds to the probes. During the last step, gathered data is stored into database. This kind of automated scan, as being too noisy, may not suit to Cyber Red Team operations. (Metasploit Pro, User Guide, 2015, 41-46)

It is possible to execute individual auxiliary modules one by one on selected targets. User may also create and add own scanning modules on top of features provided by Metasploit Framework. These include things like built in support for proxies, SSL, reporting, and built in threading. (The ultimate guide to the Metasploit Framework)

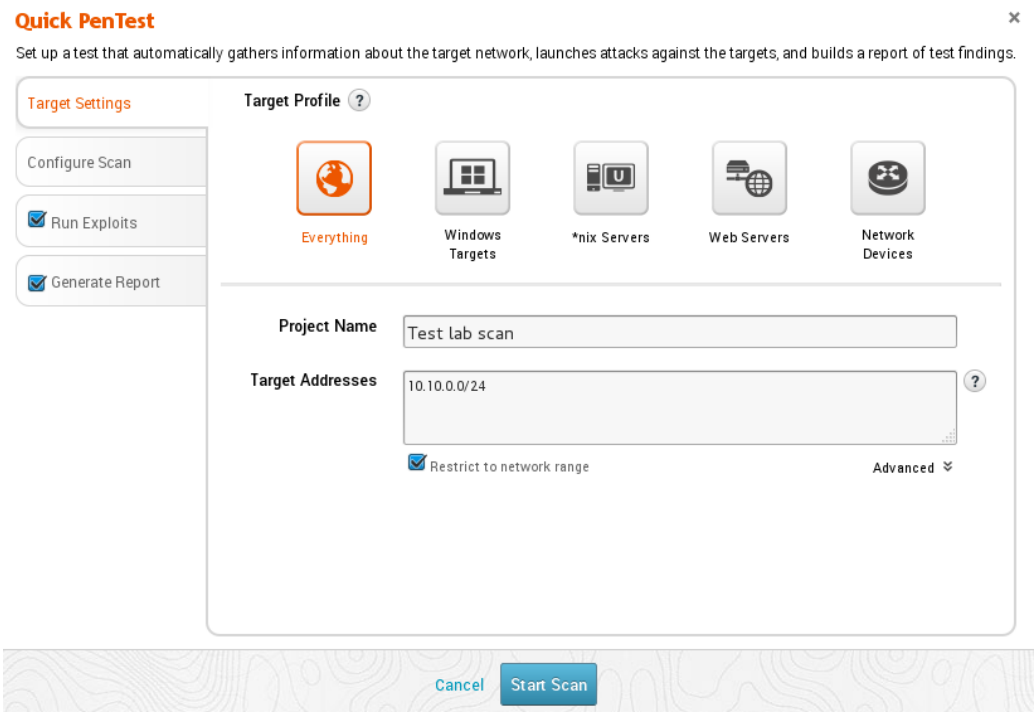


Figure 14 Quick PenTest wizard

CR-07, Exploits. Metasploit Pro is built on top of the Metasploit Framework which provides an exploit database. Metasploit Framework contains over 1500 exploits for different platforms including Windows, Unix, Linux and Android systems. There are also exploits for applications like databases, content management systems or web applications. Users can also create custom exploit modules for new vulnerabilities.

CR-08, Privilege escalation. Metasploit provides multiple tools for privilege escalation. The Meterpreter shell, one of provided payloads, contains post-exploitation capabilities that enable user to do things like escalate privileges. Meterpreter script, getsystem, uses a number of different techniques to attempt to gain system level privileges on the attacked host. There are also various local exploits that can also be used to also escalate privileges. If these does not work, one option is to run brute force attacks to escalate account privileges and to gain access to exploited machines. (Mittal, 2011, 11)

CR-09, Data exfiltration. A successful exploit results in an open session which allows user to extract information from a target. For this, Metasploit Pro provides an automated evidence collection wizard. This tries to collect information, such as system details, passwords, installed applications, logged-on users or joined domain,

from target. This functionality is nice in exercises and also for penetration testers but are not useful in Cyber Red Team operations. Again, user may execute individual modules to get desired information. (Metasploit Pro, User Guide, 2015, 19)

Metasploit provides also file management features. From the Meterpreter console it is possible to download individual files using the "download" command. The file_collector script allows the user to search and download files that match a specific pattern.

CR-10, Persistence. Metasploit Pro does not provide any tools for a persistence so user must use ones provided by the Metasploit Framework. It provides a Meterpreter script, persistence.rb, that creates a service based backdoor. This will be configured to start automatically either when the user logs on or system boots and it tries to open reverse TCP connection to defined listener. Another backdoor, one that allows direct bind, can be done with metsvc script. Both of these scripts allows connections without any authentication and are easy to find. In addition to these, some Metasploit modules provide persistence options. There are, for example, persistence modules for Windows that built persistence through registry or scheduled task. (The ultimate guide to the Metasploit Framework)

CR-11, Lateral movement. The portfwd command in Meterpreter shell can be used as a pivoting technique. It allows to forward TCP connections through the exploited host making it a pivot point. Through this pivot, user can execute online attacks to other hosts in target network or use for example use file upload functionality together with psexec commands for lateral movement. (Miller, 2004, 37)

CR-12, Payload. Payload types that Metasploit Pro provides are Meterpreter, Command shell and Powershell (Figure 15). **Meterpreter**, or Meta-Interpreter, is an advanced, multi-function payload that operates via reflective DLL injection. It provides an advanced interactive shell that provides post-exploitation capabilities that allows user to escalate privileges, dump password hashes, take screenshots, launch and migrate processes, and upload files to the target. It resides completely in the memory, which makes it difficult to detect with conventional forensic techniques. New features can be accomplished through the Meterpreter scripting environment. These can be loaded and unloaded dynamically when needed. Apart

from these, Meterpreter can be further extend with extensions. For example, Incognito extension allows to impersonate user tokens from compromised system. (Mittal, 2011, 1; (The ultimate guide to the Metasploit Framework))

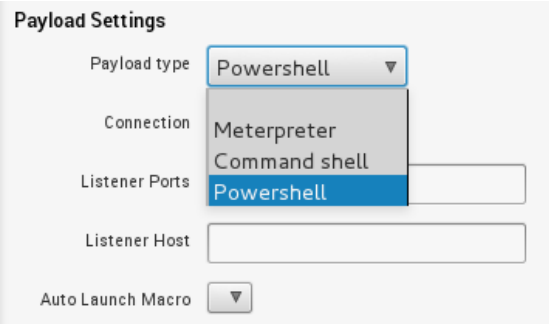


Figure 15 Payload type selection

Command shell payload provides a command shell that can be used to run single commands on an exploited host. It provides limited capabilities and can start a new process that can be easily detected by IDS or anti-virus systems.

Powershell payload provides an ability to execute PowerShell cmdlets on exploited host. These are hard to find by anti-virus products because they usually do not touch disk and do not involve risky process injections.

With Metasploit Pro, user can select between two possible C2 channels; bind or reverse. Bind connection is useful when the targets are behind a firewall or a NAT gateway. Reverse connection is useful if firewall blocks direct connections to the target. (Metasploit Pro, User Guide, 2015, 150)

CR-13, Evasion. Metasploit Pro allows to set transport layer evasion level as a part of auto-exploitation options. Low level inserts delays between TCP packets, medium sends small TCP packets and high sends small TCP packets and inserts delays between them. User can also set evasion level for exploits involving DCE/RPC, SMB and HTTP. The higher the application evasion level, more evasion techniques are applied. (Metasploit Pro, User Guide, 2015, 100)

Metasploit Pro's Generate Payload module provides dynamic payload generation for AV evasion (Figure 16). It creates dynamic payloads by making the stage of the payload dynamic. This is done with randomized C code. (Maloney, 2013, 11)

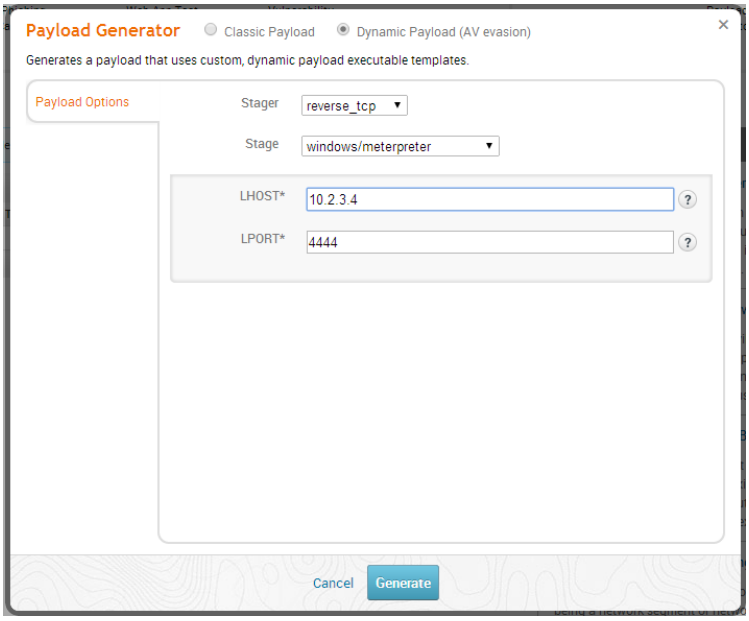


Figure 16 Dynamic payload creation

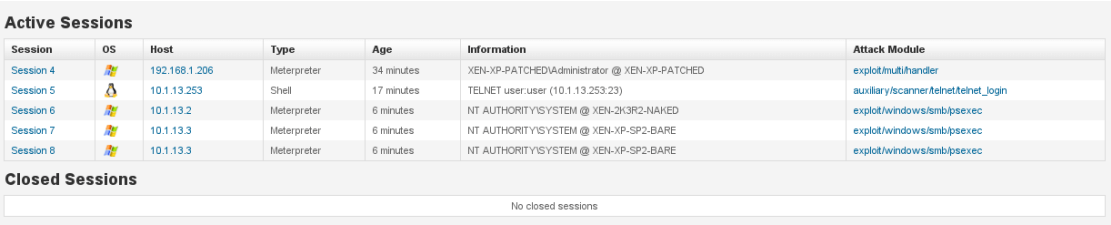
CR-14, Compatibility with other tools. Metasploit allows to import vulnerability scan reports from third party scanners, such as Nessus, Core Impact, Qualys, Burp or Retina. Imported data, such as each host's operating system, services, and discovered vulnerabilities, is imported into the project. (Metasploit Pro, User Guide, 2015, 13)

Metasploit Pro can be integrated with Nexpose. Integration allows to execute Nexpose vulnerability scans directly from Metasploit Pro's web interface. If Metasploit is used for vulnerability validation, validation results can be sent back to Nexpose. (Metasploit Pro, User Guide, 2015, 50-53, 70)

CR-15, Teamwork. The multi-user support of Metasploit Pro provides the collaboration features for team members. Team members can log into the same instance of Metasploit Pro to perform tasks, review data, and share projects. Members with administrative privileges can set network boundaries and add tags to hosts. Tags can be used to assign certain hosts to a specific team member to attack

or test. Teaming support is available through Metasploit web interface. (Metasploit Pro, User Guide, 2015, 29-33)

CR-16, Situation awareness. Metasploit Pro does not provide much for situational awareness. Team members connected to same Metasploit Pro instance can see current active sessions in one table (Figure 17).



Active Sessions

Session	OS	Host	Type	Age	Information	Attack Module
Session 4		192.168.1.206	Meterpreter	34 minutes	XEN-XP-PATCHED\Administrator @ XEN-XP-PATCHED	exploit/multi/handler
Session 5		10.1.13.253	Shell	17 minutes	TELNET user:user (10.1.13.253:23)	auxiliary/scanner/telnet/telnet_login
Session 6		10.1.13.2	Meterpreter	6 minutes	NT AUTHORITY\SYSTEM @ XEN-2K3R2-NAKED	exploit/windows/smb/psexec
Session 7		10.1.13.3	Meterpreter	6 minutes	NT AUTHORITY\SYSTEM @ XEN-XP-SP2-BARE	exploit/windows/smb/psexec
Session 8		10.1.13.3	Meterpreter	6 minutes	NT AUTHORITY\SYSTEM @ XEN-XP-SP2-BARE	exploit/windows/smb/psexec

Closed Sessions

No closed sessions

Figure 17 Active sessions

Because all project information, including exploited hosts, are in database, it is possible to transfer session information to other systems.

CR-17, Reporting. Metasploit Pro provides good reporting features. Reporting tasks, generating, downloading, e-mailing, and deleting reports, can be done from the web interface. Reporting options allow to define which hosts are included to or excluded from report. Reports are available in HTML, PDF, RTF or Microsoft Word formats.

Reports that are most useful in Cyber Red Team operations are Activity report and Compromised and Vulnerable hosts. The first one shows exact exploits and modules run at the technical level. The second one lists all hosts on which user was able to open a session, successfully run a module, or record a vulnerability.

A custom reports can be created using Jasper report templates. (Metasploit Pro, User Guide, 2015, 234-259)

5.5 Faraday 1.0.15

Faraday is different kind of software in this evaluation. It is not suited to be evaluated against evaluation criteria but deserves to be included to this thesis.

Faraday is a product that integrates other security and penetration testing tools below Integrated Pentest Environment (IPE). Faraday is purely designed for collaborative penetration testing and secure data sharing among team members. Idea is to re-use the available tools in the community to take advantage of them in a multiuser way. (Faraday, 2016)

Faraday provides some user interface options to choose from. Graphical GUI includes command line console for entering commands, host tree showing discovered hosts, log console showing execution logs and item info view which shows detailed host information. Faraday's dashboard, web GUI, contains a summary and visualizations of the data in a workspace (Figure 18). It contains also simple chat. The ZSH terminal provides command line user interface. (Faraday, 2016)

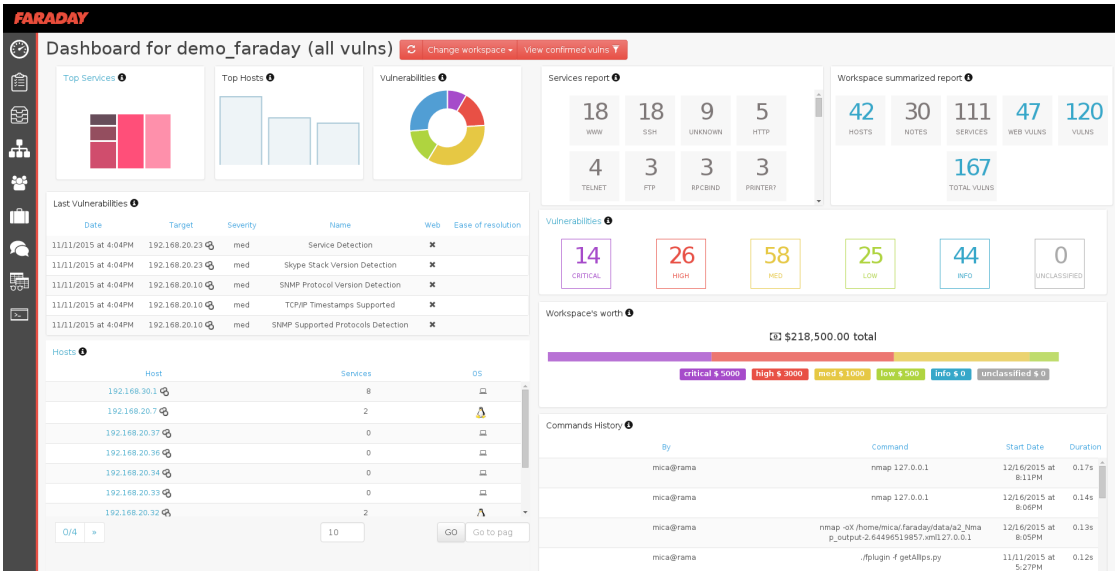


Figure 18 Faraday's dashboard

External products are integrated with plugins. There are three kinds of plugins available for Faraday. Console plugins intercept commands entered in the console. Fired plugin executes external tool and reads and interpreters tool’s output. Report plugins import files created by external tools. API plugins connect tools via external APIs or databases. Current version contains 51 plugins for products like Acunetix, Beef, Burp, Immunity Canvas, Metasploit, Nexpose, Nikto, Nmap, and Retina.

Provided documentation and existing plugins provides information to write own plugins. (Faraday, 2016)

Received information is stored into current workspace. Workspaces are used to divide information into smaller units and to integrate all the results from team members in current project.

Faraday uses CouchDB database as a data storage. Database is also used for data replication and sharing between multiple Faraday instances and for data transformations for reporting. Two team usage topologies are admitted in the configuration, Server Centric and Replicated. In server centric topology, all team members use the same CouchDB database. In Replicated topology every user has an own local CouchDB and replicate with a centralize CouchDB Server. In either way, team members are able to share gathered information. (Faraday, 2016)

Implemented reporting allows to create reports using the results obtained in each workspace. When the report is generated, all the data is processed and placed in a Microsoft Word compatible document that can then be downloaded. (Faraday, 2016)

Faraday is developed by Infobyte Security.

5.6 Results

All the evaluated programs have own strengths and weak points. Following contains a short summary of each of the evaluated programs.

Cobalt Strike is the only one of these evaluated applications that is purely made for Adversary Simulations and Red Team Operations. It has many of those features that are required from Red Team application. Provided documentation is good and video material contains lots of useful hints and information for any new Red Team member. The user interface is logical and situational awareness information is clearly presented and available to all users. It supports team operations and and created reports combine data from all connected team servers. Reconnaissance tools are not the best ones but external tools can be used through a pivoting.

The biggest weak points are the missing database and the fact that a beacon, and thus most of post exploitation features, works only in Windows environments. It

does not either have any support for online attacks. However, a beacon can be easily exported and delivered by third party products. Beacon's staging process is compatible with Metasploit Framework.

Core Impact Pro provides also good documentation and video training material. Application feels and looks professional and logical. Rapid Penetration Tests automate many phases of penetration testing including reconnaissance and exploitation. These automated functions are not very useful in Red Team operations but it is possible to execute individual modules as needed. It has a comprehensive exploit library and good post exploitation features. These covers also network devices and surveillance cameras. Provided reporting is good.

Team work support is sufficient but situational awareness information is the biggest weak point. Simple table does not provide suitable view for this.

Immunity Canvas has decent set of exploits for the most popular target systems. In addition to these, third parties provide additional exploit libraries. Available are agents for mostly needed targets, including Android devices. Agents are based on MOSDEF which makes it possible to extend agent's functionality and also creation of custom agents. Pivoting is possible through all agents regardless of agent's type. Canvas is implemented with Python language and can be modified and extended. Canvas provides a good framework for exploit development.

Provided documentation level is not adequate. There are lots of features that are not covered either in tutorials or in video material. Also user interface is illogical and disorganized and thus requires little more work go get familiar with it. Node view shows planted nodes but this view is only user specific. It does not either provide suitable support for team work, provided commander/operator functionality is not very useful. Session data is stored into text files and makes impossible to transfer data to external systems.

Metasploit Pro provides good documentation. User interface is also logical and provides automated task for common penetration testing phases including reconnaissance and exploitation. These may not be useful in Red Team operations but it is possible to execute individual modules. Custom automated tasks can be created with resource scripts. Metasploit uses a PostgreSQL database and thus it can

be integrated to external systems. Exploit database is from free Metasploit Framework. Post exploitation and reporting features are good. Team work and information sharing is possible just by using same Metasploit Pro instance.

Metasploit Pro does not provide suitable situational awareness view. Simple session table does not provide enough information. Most of features offered in Metasploit Pro are from Metasploit Framework modules.

Faraday has their own approach to penetration testing. It is not a new penetration testing tool but rather it tries to combine existing tools under one user interface. Tested version contained over 50 plugins for different applications. Documentation provides instructions for custom plugin creation. Gathered data is stored into a database and information sharing is possible either by using a common database or by database replication. Product is still quite new and feels still unfinished. Idea is good because this kind of solution may solve common problem for getting combined action report of Red Team operations.

6 Analysis of the results and conclusions

6.1 Summary of the thesis

The purpose of this was to study, evaluate and estimate how well commercial penetration and Advanced Persistent Threat simulation applications suit to Cyber Red Team operations. Because almost all of the evaluated programs were meant to be used in penetration testing appointments, this study purpose can also be seen as task to evaluate how well current commercial penetration testing applications suit to Cyber Red Team operations.

A selection of applications to be included in an evaluation was quite straightforward, because there are not so many commercial products available in the work's scope. Some of the considered applications were given by specialist working on The Finnish Defence Forces and other were find by browsing the Internet. Included applications were agreed together with same specialists. The scope, commercial applications, left out some interesting open-source applications. Examples of these are PowerShell Empire and Metasploit Community edition. One commercial application, Immunity

Innuendo, was not available for testing. Based on publicly available information, it appears to be a promising application. (Immunity INNUENDO, 2016)

An evaluation of this kind of applications is not easy. Some issues, like user interface usability or documentation quality share peoples' opinions. Beginners prefer step-by-step style instructions whereas more advanced users like more reference guide style manuals. In a similar way beginners like to use graphical user interfaces with step-by-step wizards and more experienced users like more command line user interfaces and a script based automation. The evaluation tried to take account all personal opinions and tried to put those opinions into perspective of persons' professional level.

Tested applications were complicated and full of features. Available time for testing was not sufficient to discover all the details of evaluated applications. It would require much more time and resources as well as more usage in Cyber Red Team operations to get all out of these applications. In this way there might be found more positive or negative issues of tested applications. However, time was still sufficient to get enough information of each application so that the evaluation goals could be achieved.

One may also ask if evaluation criteria was comprehensive enough. Should it have contained also more detailed questions or technical issues like how easily command and control channel traffic can be found and identified from network traffic? Why were social engineering features not included? These all are questions that could have been included in criteria but were either left out knowingly or were not anymore added during evaluation. It is also matter of available time and resources. Evaluation criteria did not either include any weight values for the different evaluation criteria domains. If these were added, it would have been possible to get numerical rating and order for the evaluated applications. However, this was not done, because it would have done evaluation little more employing and was not seen beneficial in evaluation. Evaluation was done against agreed evaluation criteria and it takes into account only those issues included in Appendix 1. Thus, evaluation can be seen universally applicable only if the reader agrees with the evaluation criteria.

6.2 Conclusions on the research results

In the end, evaluation can be seen as successful. Evaluation team received enough information and testing time to find out the most important features of the tested applications. Important in this content means issues included into the evaluation criteria. Good thing in this kind of evaluation is that when multiple applications are tested either simultaneously or one after the other, it is easier to compare applications' features not only against the evaluation criteria but also against other applications.

None of the tested applications could fulfil all the requirements in criteria. However, each one of these may still be used or needed in Cyber Red Team operations. Each exploit, that one application provides, may be just the one required and a key to success in Cyber Red Team operation. The exploit delivery or initial foothold gaining, through social engineering or by direct exploit, may be done with an another application than will be used in post exploitation phases. In a similar way, exploit development and verification may be done with a different tool that is used for delivery. The usage of multiple tools brings a new challenge; how taken actions are tracked and reported. This may be solved with an application like Faraday.

6.3 Areas for further research

In the future, the mandatory may use information from this evaluation to further study and test similar applications. One interesting application, already mentioned at the chapter 6.1, is Immunity Innuendo. It would be useful to evaluate also this application and test its suitability against the evaluation criteria. Also freeware applications that were left out from the scope, could be tested against the criteria.

Tested applications are constantly developed and at least after main releases, it is worth of checking if there are any such new features that makes application more interesting. It this case application may be re-evaluated again by using evaluation criteria from this study.

Also, if there is a possibility to affect to development of this kind of application, criteria and issues from this study can be used as a foundation for a requirement specification.

Evaluation criteria may not be complete and comprehensive in the future. If there are any new evaluations, it may be worth to check if evaluation criteria need to be updated. There might arise new technical features or the way of working changes. Criteria can also be further developed into more technical direction. Even the work with criteria is productive, at least I found it to be.

References

Aitel, D. 2004. *And introduction to MOSDEF*. Accessed 27.4.2016. Retrieved from http://www.blackhat.com/presentations/win-usa-04/bh-win-04-aitel.pdf

Benson, K. 2007. *Military Adaptation of Red Teaming*. Accessed 11.2.2016. Retrieved from http://web.mit.edu/ssp/seminars/wed_archives07fall/benson.htm

Brangetto, P., Rõigas H., Çalışan E., CCDCEO. 2015. *Cyber Red Teaming. Organizational, technical and legal implications in a military context*. Accessed 11.2.2016. Retrieved from https:/ccdcoe.org/sites/default/files/multimedia/pdf/Cyber_Red_Team.pdf

Core Impact Pro, User Guide. 2015. Core Security.

Decision-Making and Problem Solving: Human and Organizational Factors. 2011. Accessed 11.2.2016. Retrieved from https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/66069/20110706JDN311_FINAL_WEBAWB.pdf

Faraday. 2016. Accessed 3.5.2016. Retrieved from https://github.com/infobyte/faraday/wiki

Field Manual 5-0, The Operations Process. 2010. Accessed 12.2.2016. Retrieved from https://fas.org/irp/doddir/army/fm5-0.pdf

Gaining the advantage. Applying Cyber Kill Chain® Methodology to Network Defense. 2015. Accessed 11.2.2016. Retrieved from http://cyber.lockheedmartin.com/gaining-the-advantage-applying-cyber-kill-chain-methodology-to-network-defense.

Immunity CANVAS. 2016. Accessed 27.4.2016. Retrieved from http://www.immunityinc.com/products/canvas/index.html

Immunity INNUENDO. 2016. Accessed 15.5.2016. Retrieved from https://immunityinc.com/products/innuendo/

Maloney, D. 2013. *Evading Anti-virus Detection with Metasploit*. Accessed 3.5.2016. Retrieved from http://information.rapid7.com/rs/rapid7/images/Evading%20 Antivirus %20Metasploit%20Webcast.pdf

Mateski, M. 2004. Toward a Red Teaming Taxonomy, 2.0. Accessed 14.2.2016. Retrieved from http://redteamjournal.com/2008/09/toward-a-red-teaming-taxonomy-20/

Mateski, M. 2009. *Red teaming, A short introduction*. Accessed 11.2.2016. Retrieved from http://redteamjournal.com/papers/A Short Introduction to Red Teaming (1dot0).pdf

McGeorge, A. 2013. *MOSDEF-C for you and me*. Accessed 27.4.2016. Retrieved from http://immunityproducts.blogspot.fi/2013/02/mosdef-c-for-you-and-me.html

Metasploit Pro, User Guide. 2015. Accessed 1.5.2016. Retrieved from https://community.rapid7.com/servlet/JiveServlet/downloadBody/1567-102-19-7189/pro-user-guide.pdf

Microsoft Enterprise Cloud Red Teaming. 2014. Accessed 10.2.2016. Retrieved from http://download.microsoft.com/download/C/1/9/C1990DBA-502F-4C2A-848D-392B93D9B9C3/Microsoft_Enterprise_Cloud_Red_Teaming.pdf

Miller, M. 2004. Metasploit's Meterpreter. Accessed 3.5.1016. Retrieved from https://dev.metasploit.com/documents/meterpreter.pdf

Mittal, S. 2011. *Post Exploitation Using Meterpreter*. Accessed 3.5.1016. Retrieved from https://www.exploit-db.com/docs/18229.pdf.

Mudge, R. 2016. Cobalt Strike. Accessed 30.3.2016. Retrieved from https://www.cobaltstrike.com.

Mulvaney, B. 2012. Red Teams, Strengthening through challenges. Accessed 12.2.2016. Retrieved from https://www.hqmc.marines.mil/Portals/138/Docs/PL/PLU/Mulvaney.pdf

Pietroforte, M. 2014. *PowerShell versions and their Windows version*. Accessed 21.2.2016. Retrieved from https://4sysops.com/archives/powershell-versions-and-their-windows-version/

Rains, T. 2015. *Penetration Testing, Red Teaming, & Forensics*. Accessed 10.2.2016. Retrieved from https://blogs.microsoft.com/cybertrust/2015/10/08/cloud-security-controls-series-penetration-testing-red-teaming-forensics/

Securing SCADA Infrastructure. 2010. Accessed 10.2.2016. Retrieved from http://www.web-switchpoint.com/files/docs/SCADA_infrastructure.pdf

The ultimate guide to the Metasploit Framework. Accessed 3.5.2016. Retrieved from https://www.offensive-security.com/metasploit-unleashed/

Using Windows PowerShell. 2013. Accessed 10.2.1016. Retrieved from https://technet.microsoft.com/en-us/library/dn425048.aspx

APPENDICES

Appendix 1. Evaluation Criteria

Evaluation criteria are presented in the table below.

ID	Domain	Evaluation criteria
CR-01	Documentation and training material	Does the product vendor provide an end user manual? Online or separate documents?
		Is there any training material available for beginners?
		Does the product vendor or 3th party provide product training?
CR-02	User interfaces	Is graphical user interface (GUI) usable and intuitive?
		Does the product provide a command line user interface?
		Does the product provide a scripting interface for automation?
CR-03	Implementation	Programming language(s) used to make the product?
		Is the program's implementation modular?
		Is it possible to modify the product?
		Is it possible to extend the product?
		Is it possible to integrate other tools to product?
CR-04	System updates	Are software updates easy to install?
		Is it possible to do offline updates?
CR-05	Database	Does the product contain database or other data storage?
		Is database schema available?
CR-06	Reconnaissance	Does the product provide tools for network scanning?
		Does the product provide tools for OSINT data mining?
		Does the product provide tools for Web application vulnerability scanning?

		Is it possible to import reconnaissance data from the external tools?
CR-07	Exploit library	Does the product contain support for online exploits?
		Does the product contain exploits for Windows systems?
		Does the product contain exploits for Linux/Unix operations systems?
		Does the product contain exploits for OS X operations systems?
		Does the product contain exploits for mobile systems?
		Does the product contain exploits for embedded systems?
CR-08	Privilege escalation	Does the product support privilege escalation in Windows systems?
		Does the product support privilege escalation in Linux/Unix systems?
		Does the product support privilege escalation in OS X systems?
		Does the product support privilege escalation in mobile systems?
		Does the product support privilege escalation in embedded systems?
CR-09	Data exfiltration	Does the product support data exfiltration from Windows systems?
		Does the product support data exfiltration from Linux/Unix systems?
		Does the product support data exfiltration from OS X systems?
		Does the product support data exfiltration from mobile systems?
		Does the product support data exfiltration from embedded systems?
CR-10	Persistence	Does the product provide support for persistence in Windows systems?
		Does the product provide support for persistence in Linux/Unix systems?

		Does the product provide support for persistence in OS X systems?
CR-11	Lateral movement	Does the product provide support for lateral movement in Windows systems?
		Does the product provide support for lateral movement in Linux/Unix systems?
		Does the product provide support for lateral movement in OS X systems?
CR-12	Payload	Does the product provide payload for Windows systems?
		Does the product provide payload for Linux/Unix systems?
		Does the product provide payload for OS X systems?
		Does the product provide payload for mobile systems?
		Are there multiple control and communication (C2) channel options?
		Is it possible to change C2 channel for existing payload?
		Is it possible to define working hours for payload communication?
		Is it possible to define C2 communication interval?
		Does C2 channel provide enough capacity to upload/download bigger files?
		Does C2 channel provide tunneling option?
		Does C2 channel provide option for pivoting?
CR-13	Evasion	Does product provide built in support for AV evasion?
		Is it possible to use external tool for AV evasion?
CR-14	Compatibility with other tools	Is it possible to pass a C2 session to external tool like Metasploit?
		Is it possible to export gathered intelligence to external tools?
		Is it possible to export payload to different formats like raw binary, Python, Java, etc.?

		Is it possible to use payload from external tool?
CR-15	Teamwork	Does the product provide any communication capabilities for team members?
		Do all team members share same intelligence information?
		Is it possible to share C2 channels?
		Is it possible to spawn new C2 sessions to other team members?
CR-16	Situation awareness	Do all team members see same situation awareness information?
		Does the product provide graphical presentation of situation awareness information?
		Is it possible to export situation awareness information to management system?
CR-17	Reporting	Does the product provide reporting?
		Is there a report for action timeline?
		Is there a report for found vulnerabilities?
		Is there a report for gathered intelligence?