

CASO DI STUDIO:
Attacco alla Lohacker

GRUPPO FORMATO DA: Maria Grazia Miccoli e Gabriele Marrano

INTRODUZIONE	3
FASE DI ATTACCO	4
Reconnaissance	4
Weaponization	6
Delivery	8
Exploitation	10
Installation	10
Command and Control	10
Actions on Objectives	11
FASE DI DIFESA	11
Triade CIA	11
MITRE	12
ALTRI CONCETTI	13
SERIOUS GAME	13
CICLO DI VITA DELLA SICUREZZA	15
METRICA CVSS	16

INTRODUZIONE

In questo documento verrà presentato il caso di studio relativo al corso di Cyber Security che riguarda la simulazione di un attacco e la difesa dell'obiettivo scelto.

In particolare, l'attacco consiste nel ritrovamento e nella manipolazione del file contenente le informazioni in merito ai clienti e agli ordini aperti di una nota azienda alimentare, la Loacker, che, tramite la sua modifica, ci permetterà di dirottare un ordine online già eseguito e pagato per l'ottenimento dei prodotti in modo gratuito. Di conseguenza, si andrà a cercare uno dei dipendenti dell'azienda, in particolare un dipendente del servizio clienti, per prendere il controllo del dispositivo così da poter cercare il file in questione e modificare l'indirizzo di consegna di un ordine.

Come già detto, il progetto si svilupperà in due fasi distinte:

- **Fase di attacco:** condotta dal *red team*, si svilupperà seguendo le sette fasi della **Cyber Kill Chain**, ovvero una tecnica militare usata dall'attaccante per colpire l'**asset** scelto, ovvero l'elemento "sensibile" che ha bisogno di essere protetto, in questo caso il presunto file con le informazioni degli ordini effettuati. Si farà uso di una reverse shell inviata tramite email di phishing.
- **Fase di difesa:** condotta dal *blue team*, è la fase in cui si cerca di mitigare l'attacco. La mitigazione avviene tramite una combinazione di misure preventive, difensive e correttive per ridurre la probabilità che un attacco si verifichi e per limitare i danni nel caso in cui si verifichi. Nel nostro caso, l'obiettivo sarà quello di insegnare ai dipendenti dell'azienda cosa fare per prevenire l'attacco tramite la progettazione di un serious game, ovvero un gioco formativo.

FASE DI ATTACCO

(attraverso l'uso della cyber kill chain)

In questa fase, l'asset può essere definito come l'elemento che l'attaccante vuole raggiungere per recare danni all'obiettivo. L'attacco è quindi l'evento che danneggia o intende danneggiare l'asset. Come già detto in precedenza, quest'ultimo verrà condotto seguendo le sette fasi della Kill Chain ed utilizzando sia tool semi-automatizzati che strategie manuali. Inoltre, in questo caso, si tratterà di un penetration testing di tipo **gray box**, in quanto non possediamo informazioni dettagliate sull'organizzazione dell'azienda presa in considerazione, ma parziali.



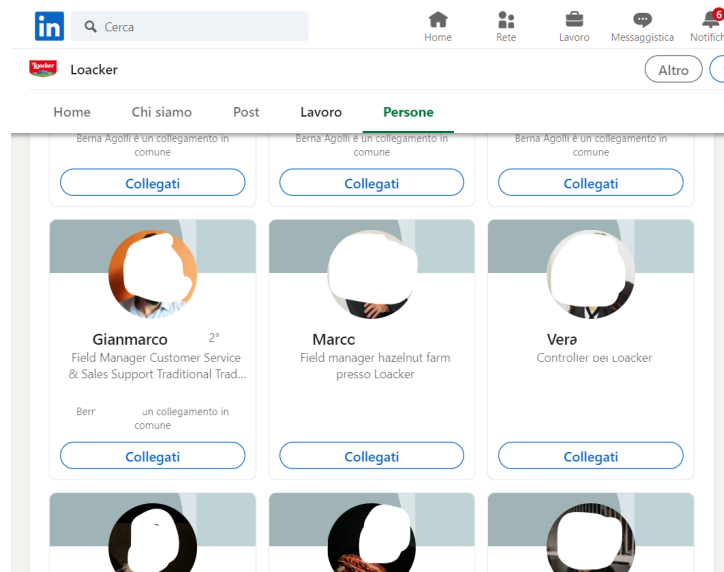
Reconnaissance

La fase di **reconnaissance**, o **ricognizione**, è la prima fase della kill chain e consiste nella raccolta di informazioni sull'obiettivo da attaccare. Inoltre, è una fase molto delicata, in quanto determina la buona riuscita (o il fallimento) delle fasi successive.

All'interno della *Reconnaissance*, è stata usata principalmente l' **OSINT (Open Source Intelligence)**, ovvero delle tecniche che si concentrano sulla raccolta e l'analisi di informazioni provenienti da fonti pubblicamente accessibili.

L'OSINT può svilupparsi in quattro fasi:

- **Discovery:** ovvero *ricerca*. In questa prima fase, si vanno a raccogliere tutte le informazioni che riguardano o interessano l'oggetto di indagine. Siccome lo scopo di questo attacco è accedere al pc di un dipendente Loacker per prenderne il controllo e vedere come effettua il cambio di indirizzo dopo averne ricevuto richiesta dal cliente stesso, abbiamo sfruttato la pagina LinkedIn di Loacker per ricercare i dipendenti, in particolare coloro che si interfacciano con il cliente stesso, coloro che lavorano per il servizio clienti. Come risultato di questa fase di ricerca, abbiamo ottenuto il nome e cognome di quattro dipendenti (oltre che alla loro carriera lavorativa).



- **Discrimination:** ovvero la *selezione*. In questa seconda fase, si va a fare una selezione dei dati raccolti nella fase precedente. Siccome per raggiungere il nostro scopo abbiamo bisogno di un solo dipendente, utilizzando dei tool, come **Skymem**, per il ritrovamento delle email aziendali, o il plugin di Google **Clearbit**, utilizzato per lo stesso scopo e avere la certezza che esistono, abbiamo selezionato un solo dipendente di cui siamo riusciti a trovare sia l'email aziendale che quella personale con dominio gmail.

A screenshot of a web application interface showing a search for 'davide' and a list of related email addresses. The search bar at the top contains 'davide' and a 'Find email addresses' button. Below the search bar, there is a section titled 'Related emails' with a table of results. The table has columns for '#', 'Email', 'Date', and 'Domain'. The results list 10 email addresses with their corresponding dates and domains.

#	Email	Date	Domain
1.	davide.tammaro@gmail.com	2019-07-19	gmail.com
2.	d.tammaro@recow.it	2022-07-02	recow.it
3.	davidet@me.com	2019-08-18	me.com
4.	davide.t@lagnascogroup.it	2018-09-24	lagnascogroup.it
5.	dt@free.fr	2016-11-18	free.fr
6.	d.t@yandex.ru	2018-08-20	yandex.ru
7.	tdavide@yahoo.it	2018-09-27	yahoo.it
8.	t.davide@silexdue.it	2019-02-10	silexdue.it
9.	td@mixmap.com	2018-06-19	mixmap.com
10.	t.d@live.com.au	2018-06-05	live.com.au

- **Distillation:** ovvero l'*analisi*. In questa terza fase, si procede con l'analisi più dettagliata dell'obiettivo scelto. Nel nostro caso, per capire come applicare il *social engineering* nelle fasi successive, abbiamo deciso di analizzare i social media più utilizzati al momento, ovvero **Facebook** e **Instagram**.
- **Dissemination:** ovvero la *diffusione*. In quest'ultima fase, viene prodotta una documentazione finale e riassuntiva del materiale raccolto e analizzato. In particolare, queste sono le informazioni trovate e che possiamo considerare

sfruttabili (il nostro obiettivo prenderà il nome di Davide, per privacy omettiamo il cognome):

- Laureato in mediazione linguistica
- Tifoso milanista frequentante lo stadio (quindi presumibilmente è abbonato)
- Ha partecipato ad una sfilata di moda locale

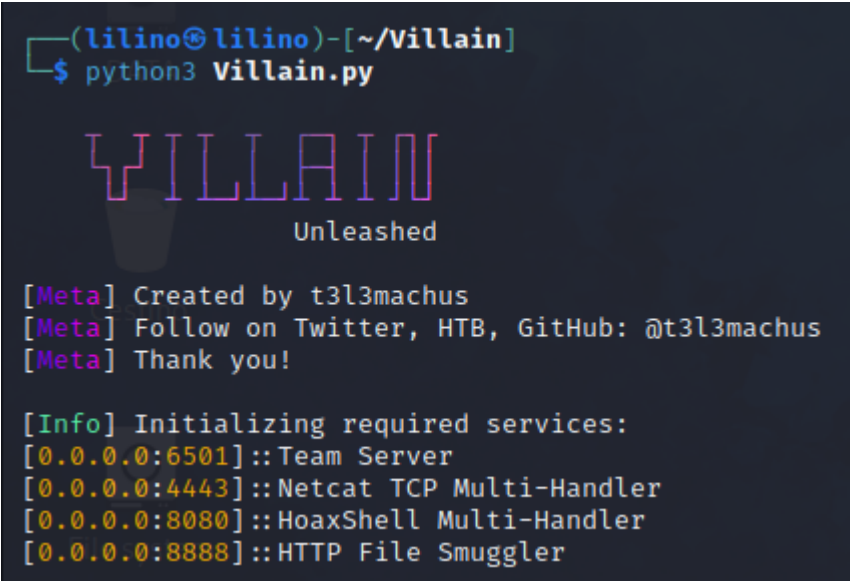
In particolar modo, abbiamo deciso di sfruttare il terzo punto nella fase di Delivery in quanto, svolgendo un'analisi anche sulla sfilata a cui ha partecipato, abbiamo scoperto che ne è stata svolta una prima edizione nell' Aprile 2022 ed un'edizione straordinaria, per la collezione autunno-inverno, ad Ottobre 2022 in seguito al successo tra gli abitanti della prima. Di conseguenza, visto che nell'anno 2023 è stata riorganizzata l'edizione primavera-estate, allora lo scopo è quello di contattare Davide con il pretesto di voler riorganizzare anche l'edizione straordinaria e di voler la sua partecipazione (verrà contattato a nome dell'organizzatore, chiamato Enrico, di cui si omette, per privacy, il cognome).

Weaponization

La seconda fase si chiama **Weaponization (armamento)** e consiste nel creare o identificare un malware utilizzabile per l'attacco, di solito si tratta di un accoppiamento di un software per l'accesso remoto (cavallo di troia) e di un exploit (software che sfrutta una vulnerabilità del sistema). Spesso per guadagnare l'accesso ad un sistema, vengono sfruttati gli zero day da cui ancora non c'è difesa poiché sono delle nuovissime vulnerabilità che devono ancora essere "patchate" proprio perché appena scoperte.

Nel caso di studio in questione si è utilizzato il tool [Villain](#) per la creazione di un payload finalizzato all'apertura di una reverse shell, e [Netcat](#), un tool a riga di comando, responsabile della scrittura e della lettura dei file in rete. Per lo scambio di dati, Netcat utilizza i protocolli di rete **TCP/IP** e **UDP**. Lo strumento ha origine dal mondo Unix; ora è diventato disponibile per tutte le piattaforme.

In particolare Villain dà la possibilità di utilizzare payload generati da template messi già a disposizione. Nel caso di studio è stato deciso di utilizzare il payload "`powershell_reverse_tcp`". Dopo aver condotto alcune prove è stato notato che l'antivirus Windows Defender, presente presumibilmente sulla macchina della vittima, avrebbe bloccato l'esecuzione del payload, perciò a seguito di alcune ricerche e approfondimenti è



```
(lilino@lilino)-[~/Villain]
$ python3 Villain.py

VILLAIN
1.0.0
Unleashed

[Meta] Created by t3l3machus
[Meta] Follow on Twitter, HTB, GitHub: @t3l3machus
[Meta] Thank you!

[Info] Initializing required services:
[0.0.0.0:6501] ::Team Server
[0.0.0.0:4443] ::Netcat TCP Multi-Handler
[0.0.0.0:8080] ::HoaxShell Multi-Handler
[0.0.0.0:8888] ::HTTP File Smuggler
```

stata eseguita una fase di **analisi statica** del codice del payload al fine di studiarne le caratteristiche che lo rendono un possibile malware per l'AV, così da effettuare opportune modifiche, e riuscire a baypassare l'antivirus in questione.

Un indicatore importante per l'analisi statica effettuata dall'AV è l'**entropia** che, nel caso di un file, indica il livello di "casualità" cioè quanto disordinatamente sono disposti i byte al suo interno o, detta diversamente, la probabilità che un determinato valore si ripeta, conoscendo i valori precedenti, questa misura indica il livello di compressione o offuscamento del codice. Se un file non è compresso/offuscato allora avrà entropia bassa e questo significa che una delle metriche utilizzate dall'AV farà risultare il payload come "non malevolo", se invece il file è compresso/offuscato allora la sua entropia è alta.

Per ridurre l'entropia nel codice del payload sono stati aggiunti quindi lunghi commenti del tipo `<#aaaa...aaaaa#>` così da abbassare il valore di entropia, più nel specificato il valore di entropia è passato da 4.93 a 2.03, riducendola così di circa il 60%

(questo calcolo è stato effettuato attraverso l'uso del teorema dell'entropia di Shannon)

E' possibile verificare tali valori attraverso l'applicazione del seguente teorema o mediante il calcolatore online presente [qui](#), e utilizzando il contenuto dei seguenti file: [payload originale](#), [payload modificato](#).

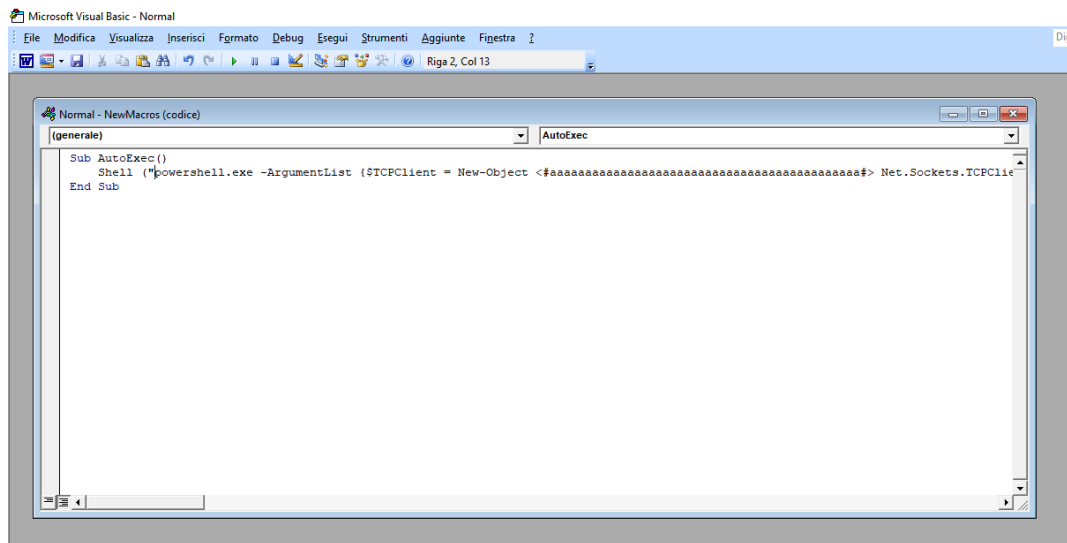
[illegible]

A questo punto sono state ripetute le prove iniziali, ed i risultati sono stati decisamente positivi, infatti con queste modifiche al payload l'AV non lo catalogava più come "possibilmente malevolo".

Il payload è stato quindi rinominato in *“custom powershell reverse tcp”*.

Infine il codice del payload, ormai pronto, è stato inserito con una macro auto eseguibile all'apertura, in un file word che verrà poi consegnato alla vittima.

L'auto esecuzione è stata effettuata attraverso il comando "*AutoExec*".



Il file word è stato creato utilizzando tecniche di **social engineering** per giustificare l'apertura e la consegna di questo, infatti il file word farà riferimento ad una sfilata di moda che si terrà ad ottobre 2023 e presenterà quindi numerose informazioni a riguardo.

Delivery

La terza fase si chiama **Delivery (consegna)** e consiste nella trasmissione della arma cyber (definita nella fase di weaponization) all'obiettivo. Normalmente si impiegano email con link a siti fasulli o documenti allegati contenenti malware per la distribuzione alla vittima. Ma anche chiavette USB, infrarossi, bluetooth, supporti ottici, tastiere o mouse con un malware "nidificato" nel firmware o altri metodi sono possibili.

Nel caso di studio è stato scelto lo [spearphishing](#) come modalità di consegna del file word. Lo spearphishing è una forma di phishing mirato, ovvero un attacco che consiste nell'inviare file o link malevoli tramite email o sms ad una vittima nello specificato fingendosi qualcun'altro.

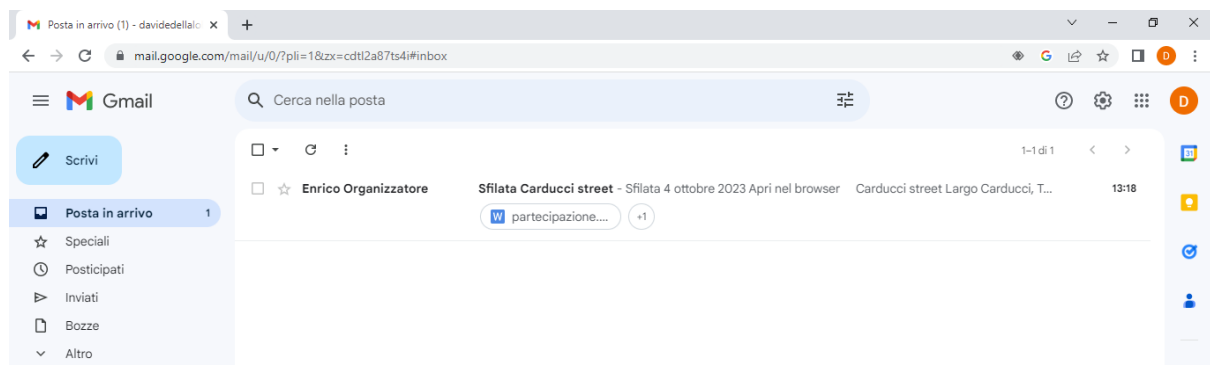
Nello specifico si è deciso di voler curare nei minimi dettagli questa fase, di conseguenza sono stati provati e poi abbandonati strumenti e piattaforme come **Temp Email**, **Set toolkit**, **Emkei** ed altre in quanto non davano la possibilità di una personalizzazione completa dell'intestazione dell'email, o l'email inviata sarebbe stata contrassegnata come spam, riducendo così il fondamentale concetto di **trust** che regge l'intero caso di studio, indi per cui, si è deciso di usufruire di un servizio di affitto di un dominio attraverso [register.it](#) e l'uso di [Brevo](#) (ex Sendinblue) per la creazione e l'utilizzo di un template per l'email, e il conseguente invio, così da aumentare la professionalità dell'email e il conseguente aumento di trust nei confronti di questa. Di seguito l'email in ricezione alla vittima:



Il dominio scelto ed affittato è stato
la casella di posta associata è stata

→ carduccistreet.it
→ enricoorganizzatore@carduccistreet.it

come si nota l'email arriva nella posta ordinaria, non è contrassegnata come spam e il mittente risulta "Enrico Organizzatore", ovvero il reale organizzatore della Carducci Street ma del quale non abbiamo inserito il cognome per ovvie motivazioni.



Exploitation

In questa quarta fase, si sfruttano le vulnerabilità. Nel nostro caso, presupponendo che Davide si sia fidato dell'email inviatagli a nome di Enrico, dopo aver scaricato l'allegato ed averci cliccato sopra per visualizzare il contenuto, si avvierà in modo automatico la macro di Office con il codice malevolo.

MODULO DI PARTECIPAZIONE

Spett.le
Enrico *****

Il/La sottoscritt
Nat... a il C.F.
residente in Via/P.zza n.
CONFERMA la sua partecipazione alla sfilata Carducci Street 2023, prevista in data 4 Ottobre 2023, in veste di modello.

Luogo e data

Firma

I dati inseriti saranno trattati unicamente per finalità connesse al procedimento per il quale la presente dichiarazione viene resa (Reg. UE 2016/679).

Installation

La quinta fase si chiama **Installation** (installazione) e consiste nell'installare, all'interno del sistema obiettivo, allo scopo di consentire all'attaccante di poter restare all'interno del sistema a suo piacimento, la cosiddetta persistenza. Di solito si usano a tale scopo dei Malware Trojan (RAT Remote Access Trojan), vengono aperte delle porte nella rete, o vengono create delle reverse shell. In questa fase il sistema viene silenziosamente, ma pesantemente modificato (possono essere modificate chiavi di registro, files di sistema, anche le partizioni di avvio). Questo è uno dei motivi per cui l'esito del ripristino dei "sistemi compromessi" non è mai scontato.

Nel nostro caso, l'installazione della reverse shell viene considerata come l'apertura della reverse shell stessa fatta nella fase precedente. La reverse shell si avvia in modo automatico e nascosto.

```
[Shell] Backdoor session established on 172.29.40.193
Villain > sessions

Session ID      IP Address      OS Type  User      Owner  Status
-----
a1aef8-07b6e6-37e42d  172.29.40.193  Windows  WINDOWS\root  Self   Active

Villain > shell a1
Failed to interpret session_id.
Villain > shell a1aef8-07b6e6-37e42d

This session is unstable. Consider running a socket-based rshell process in it.
Interactive pseudo-shell activated.
Press Ctrl + C or type "exit" to deactivate.

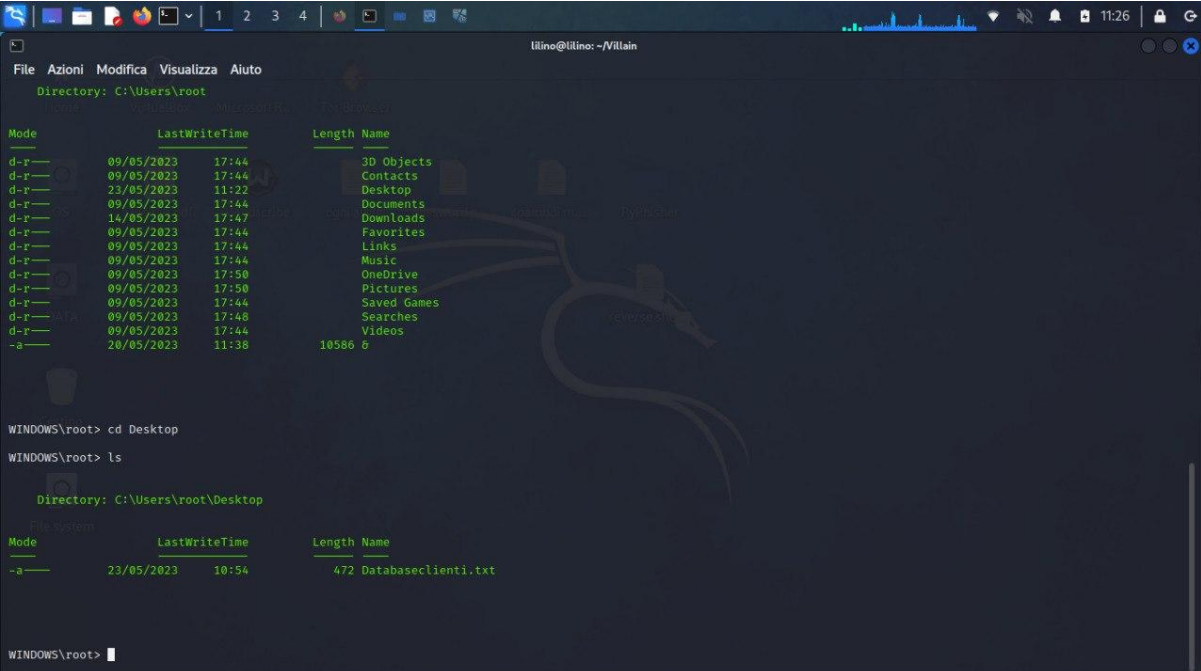
WINDOWS\root> 
```

(dispositivo dell'attaccante in cui si è aperta la sessione con la vittima)

Command and Control

Nella penultima fase, si accede al dispositivo della vittima e si prende il controllo. Nel nostro caso, una volta attivata la reverse shell in modo automatico, il listener, avviato invece sulla macchina attaccante, avrà aperto una connessione con netcat tra se stessa e la vittima.

Da questo momento si ha il controllo della vittima, è possibile spostarsi all'interno della memoria e cercare i file d'interesse.



The screenshot shows a Windows command prompt window with a dark background. The user is logged in as 'ilino@ilino: ~\Villain'. The prompt shows the current directory as 'C:\Users\root'. The user enters 'cd Desktop' and then 'ls'. The output shows a list of files and folders in the Desktop directory, including '3D Objects', 'Contacts', 'Desktop', 'Documents', 'Downloads', 'Favorites', 'Links', 'Music', 'OneDrive', 'Pictures', 'Saved Games', 'Searches', 'Videos', and a file named 'Databaseclienti.txt' with a size of 472 bytes. The prompt is currently at 'WINDOWS\root>'.

```
ilino@ilino: ~\Villain
File Azioni Modifica Visualizza Aiuto
Directory: C:\Users\root
Mode                LastWriteTime         Length Name
----                -
d-r--             09/05/2023   17:44           3D Objects
d-r--             09/05/2023   17:44           Contacts
d-r--             23/05/2023   11:22           Desktop
d-r--             09/05/2023   17:44           Documents
d-r--             14/05/2023   17:47           Downloads
d-r--             09/05/2023   17:44           Favorites
d-r--             09/05/2023   17:44           Links
d-r--             09/05/2023   17:44           Music
d-r--             09/05/2023   17:50           OneDrive
d-r--             09/05/2023   17:50           Pictures
d-r--             09/05/2023   17:44           Saved Games
d-r--             09/05/2023   17:48           Searches
d-r--             09/05/2023   17:44           Videos
-a-              20/05/2023   11:38        10586 Databaseclienti.txt

WINDOWS\root> cd Desktop
WINDOWS\root> ls

Directory: C:\Users\root\Desktop
Mode                LastWriteTime         Length Name
----                -
-a-              23/05/2023   10:54         472 Databaseclienti.txt

WINDOWS\root>
```

Actions on Objectives

L'ultima fase consiste nel vero e proprio attacco al sistema obiettivo. Nel nostro caso, andremo effettivamente a trovare l'eventuale file in cui vengono descritti gli ordini e a cambiare l'indirizzo di destinazione di un ordine già fatto e pagato.

FASE DI DIFESA

Triade CIA

L'obiettivo di questa fase è quella di proteggere gli asset, ovvero qualsiasi risorsa di valore per un'organizzazione che richiede protezione. Questa fase è svolta dal blue team, il quale non subentra solo davanti ad un attacco in corso, ma protegge il sistema in modo continuo, garantendo la riservatezza, l'integrità e la disponibilità dei dati.

Queste tre ultime proprietà da garantire vengono rinchiusi all'interno della cosiddetta **triade CIA**:

- **Confidentiality**: si riferisce alla protezione delle informazioni da accessi o divulgazioni non autorizzati. Essa può essere garantita attraverso l'uso di controlli di accesso, crittografia e misure di protezione dei dati.

- **Integrity:** riguarda la protezione delle informazioni da modifiche non autorizzate. Per garantire l'integrità, vengono utilizzati controlli come firme digitali, hash crittografici e registri delle modifiche.
- **Availability:** si riferisce alla garanzia che le informazioni siano accessibili ed utilizzabili quando necessario. Per garantire questo, vengono adottate misure come il backup dei dati, la ridondanza dei sistemi e le politiche di gestione delle emergenze.

Nel caso di studio in particolare sono state violate la confidenzialità delle informazioni riferite ai clienti, come: nome, cognome, indirizzo di residenza e numero di ordine; e anche l'integrità delle informazioni, infatti l'obiettivo finale è stato proprio cambiare l'indirizzo di destinazione di un pacco, manomettendo così le informazioni originali.

Quindi la triade CIA fornisce una base solida per la progettazione e l'implementazione di misure di sicurezza efficaci per proteggere le informazioni.

MITRE

Un'azienda, può anche fare riferimento alla difesa MITRE, un framework di cyber security che fornisce una guida per affrontare gli attacchi informatici. Nell'immagine sottostante, è presente il catalogo di tecniche di attacco e contromisure corrispondenti.

MITRE | ATT&CK®

Matrices

Contribute

Tactics

Techniques

Data Sources

Mitigations

Groups

Software

Campaigns

Resources

Blog

MATRICES

Enterprise

PRE

Windows

macOS

Linux

Cloud

Network

Containers

Mobile

ICS

Reconnaissance

10 techniques

Active Scanning (3)

Gather Victim Host Information (4)

Gather Victim Identity Information (3)

Gather Victim Network Information (6)

Gather Victim Org Information (4)

Phishing for Information (3)

Search Closed Sources (2)

Search Open Technical Databases (5)

Search Open Websites/Domains (3)

Search Victim-Owned Websites

Resource Development

8 techniques

Acquire Access

Acquire Infrastructure (8)

Compromise Accounts (3)

Compromise Infrastructure (7)

Develop Capabilities (4)

Establish Accounts (3)

Obtain Capabilities (6)

Stage Capabilities (6)

Valid Accounts (4)

Initial Access

9 techniques

Drive-by Compromise

Exploit Public-Facing Application

External Remote Services

Hardware Additions

Phishing (3)

Replication Through Removable Media

Supply Chain Compromise (3)

Trusted Relationship

Valid Accounts (4)

Execution

14 techniques

Cloud Administration Command

Command and Scripting Interpreter (9)

Container Administration Command

Deploy Container

Exploitation for Client Execution

Inter-Process Communication (3)

Native API

Scheduled Task/Job (5)

Serverless Execution

Shared Modules

Software Deployment Tools

System Services (2)

User Execution (3)

Persistence

19 techniques

Account Manipulation (5)

BITS Jobs

Boot or Logon Autostart Execution (14)

Boot or Logon Initialization Scripts (5)

Browser Extensions

Compromise Client Software Binary

Create Account (3)

Create or Modify System Process (4)

Event Triggered Execution (16)

External Remote Services

Hijack Execution Flow (12)

Privilege Escalation

13 techniques

Abuse Elevation Control Mechanism (4)

Access Token Manipulation (5)

Boot or Logon Autostart Execution (14)

Boot or Logon Initialization Scripts (5)

Create or Modify System Process (4)

Domain Policy Modification (2)

Escape to Host

Event Triggered Execution (16)

Exploitation for Privilege Escalation

Hijack Execution Flow (12)

Process

Defense Evasion

42 techniques

Abuse Elevation Control Mechanism (4)

Access Token Manipulation (5)

BITS Jobs

Build Image on Host

Debugger Evasion

Deobfuscate/Decode Files or Information

Deploy Container

Direct Volume Access

Domain Policy Modification (2)

Execution Guardrails (1)

Exploitation for Defense Evasion

File and Directory Permissions Modification (2)

Hide Artifacts (10)

Hijack Execution Flow (12)

Credential Access

17 techniques

Adversary-in-the-Middle (3)

Brute Force (4)

Credentials from Password Stores (5)

Exploitation for Credential Access

Forced Authentication

Forge Web Credentials (2)

Input Capture (4)

Modify Authentication Process (6)

Multi-Factor Authentication Interception

Multi-Factor Authentication Request Generation

Network

Nel nostro caso di studio, la vittima potrebbe usare questo framework per cercare di rilevare e mitigare un attacco di phishing, presente al seguente [link](#).

Tra le mitigazioni consigliate dal framework in questione abbiamo:

- **Antivirus/malware** → L'antivirus può automaticamente mettere in quarantena i file sospetti.
- **Prevenzione delle intrusioni di rete** → Per bloccare l'attività è possibile utilizzare sistemi di prevenzione delle intrusioni di rete e sistemi progettati per scansionare e rimuovere allegati o collegamenti e-mail dannosi.
- **Configurazione software** → Utilizza meccanismi anti-spoofing e di autenticazione

della posta elettronica per filtrare i messaggi in base ai controlli di validità del dominio del mittente.

E' possibile rilevare una mail di phishing grazie al controllo, da parte di appositi software come gli AV, delle seguenti procedure e parametri.

- **Registro applicazioni** → L'ispezione degli URL all'interno delle e-mail (inclusa l'espansione dei collegamenti abbreviati) può aiutare a rilevare i collegamenti che conducono a siti dannosi noti.
- **Creazione di file** → Monitora la creazione di nuovi file a seguito di un messaggio/email.

ALTRI CONCETTI

Altri concetti utilizzati nella fase di difesa sono i seguenti:

- **CIS (Centro di Sicurezza Informatica):** Il CIS è un'organizzazione o entità che si occupa della gestione e del coordinamento delle attività di sicurezza informatica. Il suo scopo principale è garantire la sicurezza delle informazioni e mitigare i rischi della sicurezza in un'organizzazione. Un CIS può essere responsabile dell'identificazione delle vulnerabilità, dell'implementazione di controlli di sicurezza, dell'incident response e di altre attività di sicurezza.
- **SOC (Security Operation Center):** Un SOC è un'unità operativa che monitora e risponde agli eventi di sicurezza informatica in tempo reale. Solitamente, un SOC è composto da un team di esperti di sicurezza che utilizzano strumenti e tecnologie avanzate per rilevare e rispondere alle minacce di sicurezza. Il SOC è responsabile del monitoraggio delle reti, dell'analisi dei log, dell'incident response e della gestione delle emergenze.
- **CSIRT (Computer Security Incident Response Team):** Un CSIRT è un team dedicato alla risposta agli incidenti di sicurezza informatica. Il suo ruolo principale è gestire e rispondere agli incidenti di sicurezza, investigare sulle violazioni, coordinare le attività di risposta e mitigare gli effetti degli attacchi. Un CSIRT può essere interno all'organizzazione o esterno, come un'entità di terze parti specializzata nella gestione degli incidenti di sicurezza.

SERIOUS GAME

E' importante anche prevenire gli attacchi istruendo i dipendenti di un'azienda a prestare attenzione a ciò che fanno, in questa fase ci siamo concentrati proprio su questo aspetto tramite la progettazione di un serious game.

Il vantaggio maggiore di un serious game è quello di poter rendere l'apprendimento di temi solitamente etichettati come "noiosi" più piacevoli



attraverso un'esperienza ludica coinvolgente e stimolante.

Nel caso particolare di questo progetto è stato scelto un **educational game** con lo scopo di migliorare le conoscenze dei dipendenti della Lohacker nell'ambito della cyber security.

Nel nostro caso, il gioco prende il nome di **The Red Man**.

Visto che l'obiettivo è quello di formare i dipendenti della Loacker, il gioco è ambientato nel mondo degli gnometti Loacker. In particolare, il gioco inizia con l'arrivo in azienda di un nuovo dipendente, Blue, il quale, appena arrivato davanti alla sua scrivania, trova gli Gnometti buoni, ovvero gli gnomi Loacker.



Questi soprastanti sono solo quattro degli gnomi buoni. Loro avvisano Blue che il loro acerrimo nemico, Red Man, ha saputo del suo arrivo in azienda e che quindi vuole sfruttare la sua inesperienza per arrivare ai dati sensibili dell'azienda stessa. Di conseguenza, Red Man, affiancato dagli gnometti Lohacker, gli gnomi cattivi, lo invita immediatamente ad una sfida.



Il prototipo del gioco è stato creato tramite il software Figma che è presente al seguente link: [Prototipo Figma](#).

CICLO DI VITA DELLA SICUREZZA

Come la fase di attacco, anche la difesa ha il suo ciclo di vita che è composto principalmente da cinque fasi:

- **IDENTIFY (Identificare):** In questa fase, vengono identificati e valutati gli asset critici, i rischi e le vulnerabilità del sistema. Si tratta di comprendere ciò che deve essere protetto e quali potenziali minacce possono metterlo a rischio. In questa fase vengono condotte analisi dei rischi e valutazioni della sicurezza.
Per il nostro caso, l'asset individuato è il file degli ordini già effettuati..
- **PROTECT (Proteggere):** In questa fase, vengono adottate misure di protezione per mitigare i rischi individuati nella fase precedente. Ciò può essere fatto includendo l'implementazione di controlli di sicurezza, sistemi di autenticazione o politiche di accesso. Nel nostro caso, abbiamo dato più rilevanza alla formazione dei dipendenti Loacker, presupponendo che le "protezioni tecniche" vengano già considerate dall'azienda.
- **DETECT (Rilevare):** in questa fase, vengono messi in atto sistemi e processi per rilevare e monitorare le attività sospette o gli incidenti di sicurezza. L'obiettivo è individuare tempestivamente le violazioni della sicurezza e le attività anomale.
- **RESPOND (Rispondere):**
In questa fase, viene attuato un piano di risposta agli incidenti per affrontare le minacce e le violazioni della sicurezza rilevate. Ciò può includere la gestione delle emergenze, l'isolamento delle violazioni, la raccolta di prove e la collaborazione con le autorità competenti.
L'obiettivo è rispondere prontamente agli incidenti per mitigare i danni e ripristinare la sicurezza.
- **RECOVER (Recuperare):** In questa fase, vengono adottate misure per ripristinare le normali operazioni dopo un incidente di sicurezza. Ciò può includere il ripristino dei dati e dei sistemi da backup, l'implementazione di miglioramenti basati sull'esperienza acquisita e l'analisi post-incidente per evitare future violazioni.
L'obiettivo è tornare alla piena funzionalità operativa in modo sicuro.

METRICA CVSS

Lo standard CVSS (Common Vulnerability Scoring System) è utilizzato per valutare e classificare la gravità delle vulnerabilità e fornisce una misura numerica che consente agli esperti di sicurezza di valutare e comparare i rischi associati a diverse vulnerabilità. Esso è composto da tre componenti:

- **Punteggio base o Base Score:** rappresenta la gravità intrinseca di una vulnerabilità senza considerare il contesto operativo specifico. Il punteggio base tiene conto di diversi fattori:
 - **Attack Vector:** rappresenta la modalità attraverso cui un attaccante può sfruttare la vulnerabilità. Nel nostro caso, l'attack vector sarà *Network*, perché sfruttiamo le vulnerabilità senza avere accesso diretto al sistema bersaglio.
 - **Attack Complexity:** indica la difficoltà nello svolgere un attacco. Nel nostro caso, questo parametro sarà *Low*, perché l'attacco è semplice e richiede poche risorse
 - **Privileges Required:** indica il livello di accesso o i privilegi che un attaccante deve possedere per sfruttare le vulnerabilità. Nel nostro caso, sarà *None*.
 - **User Interaction:** indica se l'attacco richiede l'interazione attiva dell'utente. Nel nostro caso, questo parametro sarà impostato a *Required*, perché abbiamo bisogno che l'utente scarichi l'allegato e apra il documento.
 - **Scope:** rappresenta l'estensione dell'impatto dell'attacco, cioè cosa coinvolge il danno. Nel nostro caso, questo parametro sarà impostato su *Changed*.
 - **Confidentiality Impact:** valuta il potenziale impatto sulla riservatezza dei dati nel sistema. Nel nostro caso sarà impostato su *High*.
 - **Integrity Impact:** valuta il potenziale impatto sull'integrità dei dati nel sistema. Nel nostro caso sarà impostato su *High*.
 - **Availability Impact:** valuta il potenziale impatto sulla disponibilità del sistema o dei servizi. Nel nostro caso sarà impostato su *none*.

Base Score Metrics

Exploitability Metrics

Attack Vector (AV)*

Network (AV:N) Adjacent Network (AV:A) Local (AV:L) Physical (AV:P)

Attack Complexity (AC)*

Low (AC:L) High (AC:H)

Privileges Required (PR)*

None (PR:N) Low (PR:L) High (PR:H)

User Interaction (UI)*

None (UI:N) **Required (UI:R)**

Scope (S)*

Unchanged (S:U) **Changed (S:C)**

Impact Metrics

Confidentiality Impact (C)*

None (C:N) Low (C:L) **High (C:H)**

Integrity Impact (I)*

None (I:N) Low (I:L) **High (I:H)**

Availability Impact (A)*

None (A:N) Low (A:L) High (A:H)

- **Punteggio temporale o Temporal Score:** considera vulnerabilità che possono variare nel tempo. Esso è caratterizzato dai seguenti parametri:
 - **Exploit code maturity:** valuta quanto è sviluppato e disponibile un exploit per sfruttare la vulnerabilità. Nel nostro caso, il parametro sarà impostato su *Functional exploit exist*.
 - **Remediation level:** indica il grado di disponibilità di soluzioni per risolvere la vulnerabilità. Nel nostro caso, questo parametro sarà impostato su *Temporary Fix*, se consideriamo che si può usare un antivirus esterno oltre a Windows Defender.
 - **Report Confidence:** indica il livello di fiducia nelle informazioni sulla vulnerabilità. Nel nostro caso, sarà impostato su *Unknow*.

Temporal Score Metrics

Exploit Code Maturity (E)

Not Defined (E:X) Unproven that exploit exists (E:U) Proof of concept code (E:P) **Functional exploit exists (E:F)** High (E:H)

Remediation Level (RL)

Not Defined (RL:X) Official fix (RL:O) **Temporary fix (RL:T)** Workaround (RL:W) Unavailable (RL:U)

Report Confidence (RC)

Not Defined (RC:X) **Unknown (RC:U)** Reasonable (RC:R) Confirmed (RC:C)

- **Punteggio ambientale o Environmental score:** valuta la specifica situazione e l'importanza del sistema interessato all'attacco. Esso è caratterizzato dai seguenti fattori:
 - **Attack Vector:** rappresenta la modalità attraverso cui un attaccante può sfruttare la vulnerabilità nel contesto specifico dell'ambiente. Anche in questo caso, sarà impostato a *Network*.
 - **Attack Complexity:** indica la difficoltà che un attaccante deve affrontare per sfruttare la vulnerabilità nel contesto specifico dell'ambiente. Anche in questo caso sarà impostato a *Non defined*.
 - **Privileges Required:** rappresenta il livello di accesso o i privilegi che un attaccante deve possedere per sfruttare la vulnerabilità nel contesto specifico dell'ambiente. Questo parametro sarà impostato a *None*.
 - **User Interaction:** indica se l'attacco richiede l'interazione attiva dell'utente bersaglio nel contesto specifico dell'ambiente. Questo parametro sarà impostato su *Required*.
 - **Scope:** rappresenta l'estensione dell'impatto della vulnerabilità nel contesto specifico dell'ambiente. Questo parametro sarà impostato su *Non defined*.
 - **Confidentiality Impact:** valuta l'importanza della riservatezza dei dati nel contesto specifico dell'ambiente. Questo parametro sarà impostato su *Non defined*.
 - **Integrity Impact:** Valuta l'importanza dell'integrità dei dati nel contesto specifico dell'ambiente. Questo parametro sarà impostato su *Non defined*.

- **Availability Impact:** valuta l'importanza della disponibilità dei servizi nel contesto specifico dell'ambiente. Questo parametro sarà impostato su *Non defined*.
- **Confidentiality Requirement:** Rappresenta l'importanza della riservatezza dei dati richiesta nel contesto specifico dell'ambiente. Questo parametro sarà impostato su *Non defined*.
- **Integrity Requirement:** Rappresenta l'importanza dell'integrità dei dati richiesta nel contesto specifico dell'ambiente. Questo parametro sarà impostato su *Non defined*.
- **Availability Requirement:** Rappresenta l'importanza della disponibilità dei servizi richiesta nel contesto specifico dell'ambiente. Questo parametro sarà impostato su *Non defined*.

Environmental Score Metrics		
Exploitability Metrics		
Attack Vector (MAV)		
Not Defined (MAV:X)	Network (MAV:N)	Adjacent Network (MAV:A)
Local (MAV:L)	Physical (MAV:P)	
Attack Complexity (MAC)		
Not Defined (MAC:X)	Low (MAC:L)	High (MAC:H)
Privileges Required (MPR)		
Not Defined (MPR:X)	None (MPR:N)	Low (MPR:L) High (MPR:H)
User Interaction (MUI)		
Not Defined (MUI:X)	None (MUI:N)	Required (MUI:R)
Scope (MS)		
Not Defined (MS:X)	Unchanged (MS:U)	Changed (MS:C)
Impact Metrics		
Confidentiality Impact (MC)		
Not Defined (MC:X)	None (MC:N)	Low (MC:L)
High (MC:H)		
Integrity Impact (MI)		
Not Defined (MI:X)	None (MI:N)	Low (MI:L)
High (MI:H)		
Availability Impact (MA)		
Not Defined (MA:X)	None (MA:N)	Low (MA:L)
High (MA:H)		
Impact Subscore Modifiers		
Confidentiality Requirement (CR)		
Not Defined (CR:X)	Low (CR:L)	
Medium (CR:M)	High (CR:H)	
Integrity Requirement (IR)		
Not Defined (IR:X)	Low (IR:L)	Medium (IR:M)
High (IR:H)		
Availability Requirement (AR)		
Not Defined (AR:X)	Low (AR:L)	
Medium (AR:M)	High (AR:H)	

Il punteggio va da 0.0 a 10.0, nel nostro caso il risultato è il seguente:



Quindi, il nostro punteggio CVSS è 8.0.

CVSS Base Score: 9.3

Impact Subscore: 5.8

Exploitability Subscore: 2.8

CVSS Temporal Score: 8.0

CVSS Environmental Score: 8.0

Modified Impact Subscore: 5.7

Overall CVSS Score: 8.0