

EXPLOIT FARM

Road to “1.0.0” Release



DOMINGO DIRUTIGLIANO

CONTENTS

01

Introduzione

- Il progetto
- Stato attuale
- Cosa manca

02

Requisiti e analisi

- Documento di specifica dei requisiti
- Analisi dei rischi
- COCOMO Analysis

03

Management

- Kanban & Backlog
- Ri-Progettazione DB
- Scheduling GANTT

INTRODUZIONE

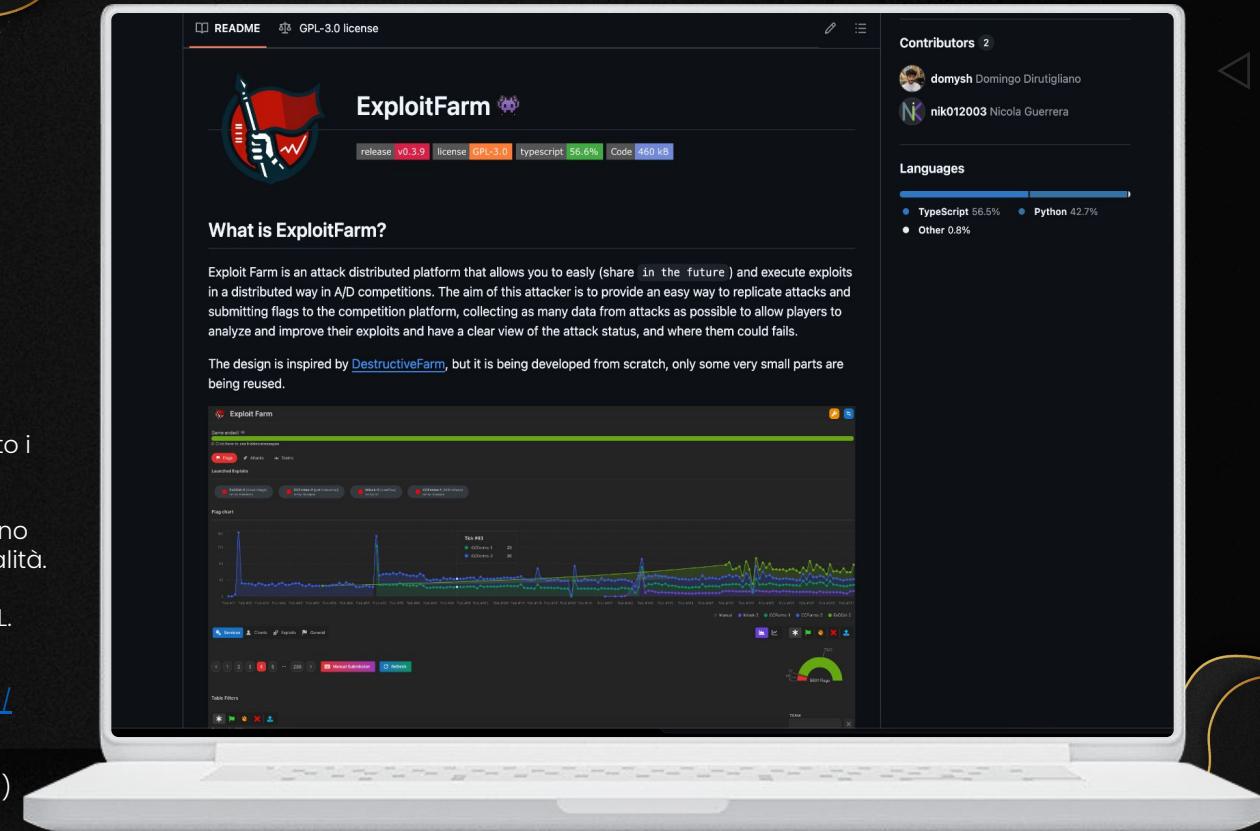
IL PROGETTO

Exploit farm è una piattaforma per la gestione di attacchi distribuiti e di submission di flag per competizioni CTF Attack Defence. Il suo target sono appunto i team CTF in tutto il mondo, e si propone come drop-replacement di moltissime soluzioni già presenti, che tuttavia peccano di alcune o quasi la totalità delle funzionalità.

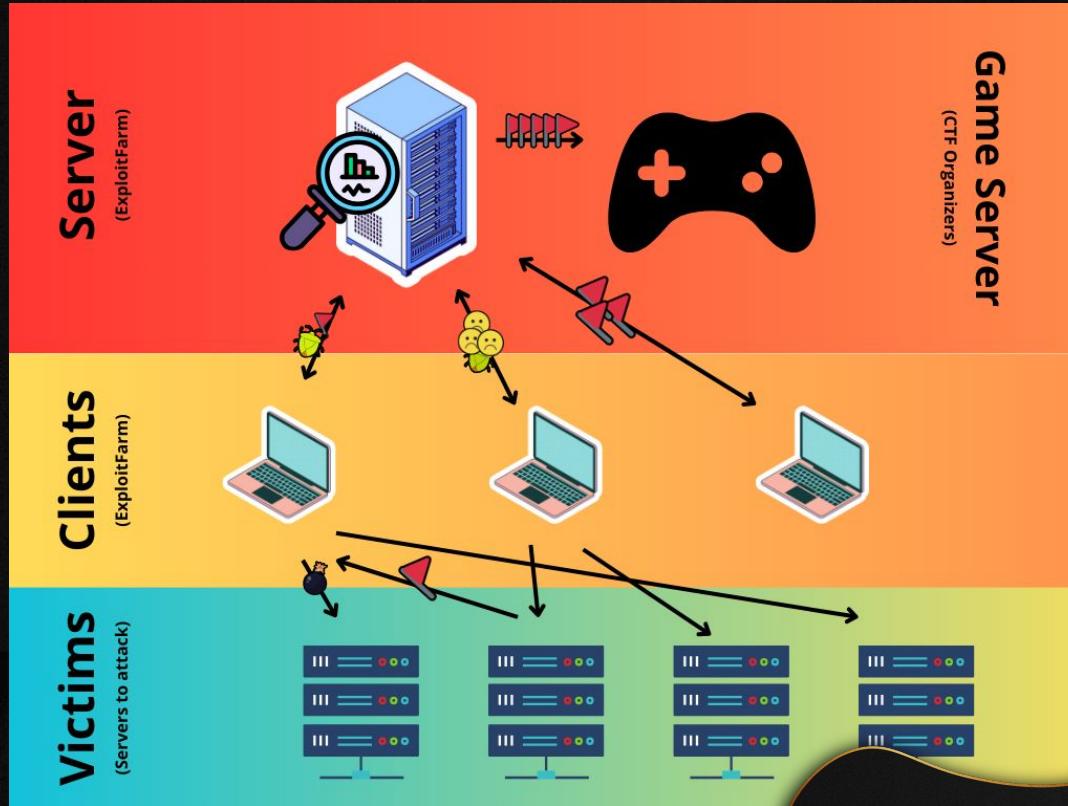
Il progetto è open-source con licenza GPL.

<https://github.com/Pwnzer0ttl/exploitfarm/>

DEMO: <https://exploit.domy.sh>
(protetta da password)



STATO ATTUALE



STATO ATTUALE



ATTACCHI E SUBMISSION

La gestione dell'avvio degli attacchi parallelo e il sistema che consegna le flag è funzionante, utilizzato anche a CC2024.



STATISTICHE

Il sistema colleziona informazioni su tempo di attacco, stdout, stderr, client, orario di esecuzione e ovviamente le flag collezionate



EASY CONFIG

Dispone di interfaccia web, [libreria python](#), con funzionalità semplici per configurare, e modificare anche a runtime le configurazione.



XFARM (TUI)

Interfaccia Terminal UI per l'avvio, test degli attacchi con configurazione semplice e intuitiva, con visualizzazione dello stato degli attacchi

COSA MANCA (OBIETTIVI DEL PROGETTO)



GESTIONE DEI SOURCE CODE DEGLI EXPLOIT + VERSIONING

- Non viene presa traccia dello stato del source code degli exploit
- Non è possibile condividere degli exploit tramite la piattaforma
- Non viene tenuto traccia sull'exploit utilizzato per ogni attacco submittarlo



GESTIONE DEGLI ATTACCHI CONDIVISI

- Non è possibile distribuire il carico di un singolo attacco su più client
- Manca un sistema di orchestrazione di attacchi condivisi

L'OBIETTIVO DEL PROGETTO È COMPLETARE I REQUISITI E RILASCIARE LA 1.0.0

REQUISITI E ANALISI

DOCUMENTO DI SPECIFICA DEI REQUISITI

Il documento raccoglierà le specifiche sull'intero progetto che mancava di una componente simile, con tutte le funzionalità richieste, implementate e da implementare nel sistema con alcune caratteristiche tecniche.

Contenuti principali del documento:

- Descrizione generale
- Descrizione dell'architettura generale della piattaforma
- □ Specifica di alcune fasi di utilizzo del software
- Descrizione delle specifiche generali
- Dettagli tecnici su alcune parti del sistema
- Impostazione dell'organizzazione di progetto
- Scheduling e analisi COCOMO
- Risk management
- △ Analisi SWOT

Contents

1	Introduzione	3
1.1	Descrizione generale	3
1.2	Obiettivi	3
1.3	Perche lo sviluppo di un nuovo attacker/submitter?	3
2	Specifiche	3
2.1	Composizione del progetto	3
2.1.1	Backend	4
2.1.2	Frontend	4
2.1.3	CLI (xfarm)	4
2.2	Specifiche dei requisiti	5
2.2.1	Setup	5
2.2.2	Dinamicità delle configurazioni	5
2.2.3	Gestione Exploit	5
2.2.4	Esecuzione Attacchi	5
2.2.5	Analisi statistiche e visualizzazione	6
2.2.6	Gestione di attacchi distribuiti	6
2.3	Specifiche Tecniche	6
2.3.1	Databases	6
2.3.2	Gestione funzionalità backend	7
2.3.2.1	API HTTP	7
2.3.2.2	Stats processor	8
2.3.2.3	Submitter Process	8
2.3.3	Funzionalità frontend	8
2.3.4	Strutturazione del client (xfarm)	8
2.3.5	Autobilanciamento del carico sul client per l'esecuzione degli attacchi	9
2.3.6	Gestione e versioning degli exploit	9
2.3.7	Controllo distribuito degli attacchi condivisi	9
3	Project Management	9
3.1	Gestione generale	9
3.2	Kanban (github)	9
3.3	Scheduling	10
3.4	COCOMO Analysis	10
3.4.1	Effort Multipliers	11
3.4.2	Scale Factors	11
3.4.3	Risultato Finale	11
3.5	Risk Management and Analysis	12
3.5.1	Stima dell'Effort e requisiti	12
3.5.2	Rilascio di versioni non totalmente funzionanti	12
3.5.3	Rischio di rendere la piattaforma complessa gestire	12
3.6	Release Management	12
3.7	Modello di Business	12
4	Analisi SWOT	13
4.1	S: Punti di forza	13

ANALISI DEI RISCHI

SOTTOSTIMA

di effort e requisiti

Rischio di aver sottostimato eccessivamente il refactoring e lo sviluppo necessario alle nuove funzionalità: rischio medio. Mitigazione: accelerare lo sviluppo per far emergere il prima possibile eventuali problematiche aggiuntive

COMPLICAZIONE

dell'architettura

Essendoci anche 1 solo sviluppatore e senza avere feedback esterni, c'è la possibilità molto alta che si crei un bias per cui si comincino a sviluppare sistemi difficili da utilizzare.

Mitigazione: chiedere il continuo confronto durante lo sviluppo con i probabili utenti che useranno la piattaforma.

RELEASE

non funzionanti

Attualmente i test sono manuali e finalizzati al verificare il funzionamento di quella componente che stiamo sviluppando. Questo lascia libero spazio a bug un po' più "lateralì" di crearsi a seguito di release. Soluzione: scrivere test automatici da eseguire prima di ogni release nella release build chain.

COCOMO ANALYSIS

Il modello COCOMO utilizzato per la stima è quello del Post-architecture model dato che la fase in sviluppo è una fase di implementazione di feature con un progetto di base già pre-ingegnerizzato.

Size stimata = 2k righe di codice

$$E = A \times \text{Size}^B \times \prod_{i=1}^n EM_i$$

$$B = 0.91 + 0.01 \times \sum \text{Scale Factors}$$

$$\prod_{i=1}^n EM_i = 1.431$$

$$B = 0.91 + 0.01 \times \sum \text{Scale Factors} = 0.99$$

$$E = A \times \text{Size}^B \times \prod_{i=1}^n EM_i = 8.36$$

Moltiplicatori di effort scelti:

Si utilizzerà un subset dei 17 moltiplicatori previsti per COCOMO post-architecture.

- RELY = 1.26 (Very High)
- DATA = 1.00 (Nominal)
- CPLX = 1.17 (High)
- RUSE = 0.95 (Low)
- DOCU = 1.00 (Nominal)
- TIME = 1.29 (Very High)
- PVOL = 1.00 (Nominal)
- PCAP = 0.88 (High)
- TOOL = 0.90 (High)
- SITE = 1.00 (Nominal)
- SCED = 1.00 (Nominal)

Moltiplicatori di scala scelti:

- PREC = 1.24 (Very High)
- FLEX = 1.01 (Very High)
- RESL = 4.24 (Nominal)
- TEAM = 1.0 (No team, not used)
- PMAT = 1.56 (Very High)
- SCED = 1.00 (Nominal)

COCOMO ANALYSIS - DURATION

Utilizzando il risultato precedente usiamo la formula di COCOMO per stimare la durata del progetto:

$$C = 3.67$$

$$D = 0.28 + 0.2 \times (B - 0.91) = 0.296$$

$$T = C \times E^D = 6.8(\text{mesi})$$

Tuttavia assumendo gli obiettivi del progetto, è intuitivo comprendere come la stima eseguita è eccessiva rispetto al reale tempo necessario al raggiungimento dell'obiettivo.

MANAGEMENT

KANBAN & BACKLOG

Il progetto è interamente condiviso e gestito su github, grazie alla funzionalità dei progetti di cui si sfrutta il backlog che permette di gestire le user-stories con tag, ordini di priorità e direttamente associabili ai branch in cui si esegue l'attività di sviluppo, per questo molto integrato con lo sviluppo stesso.

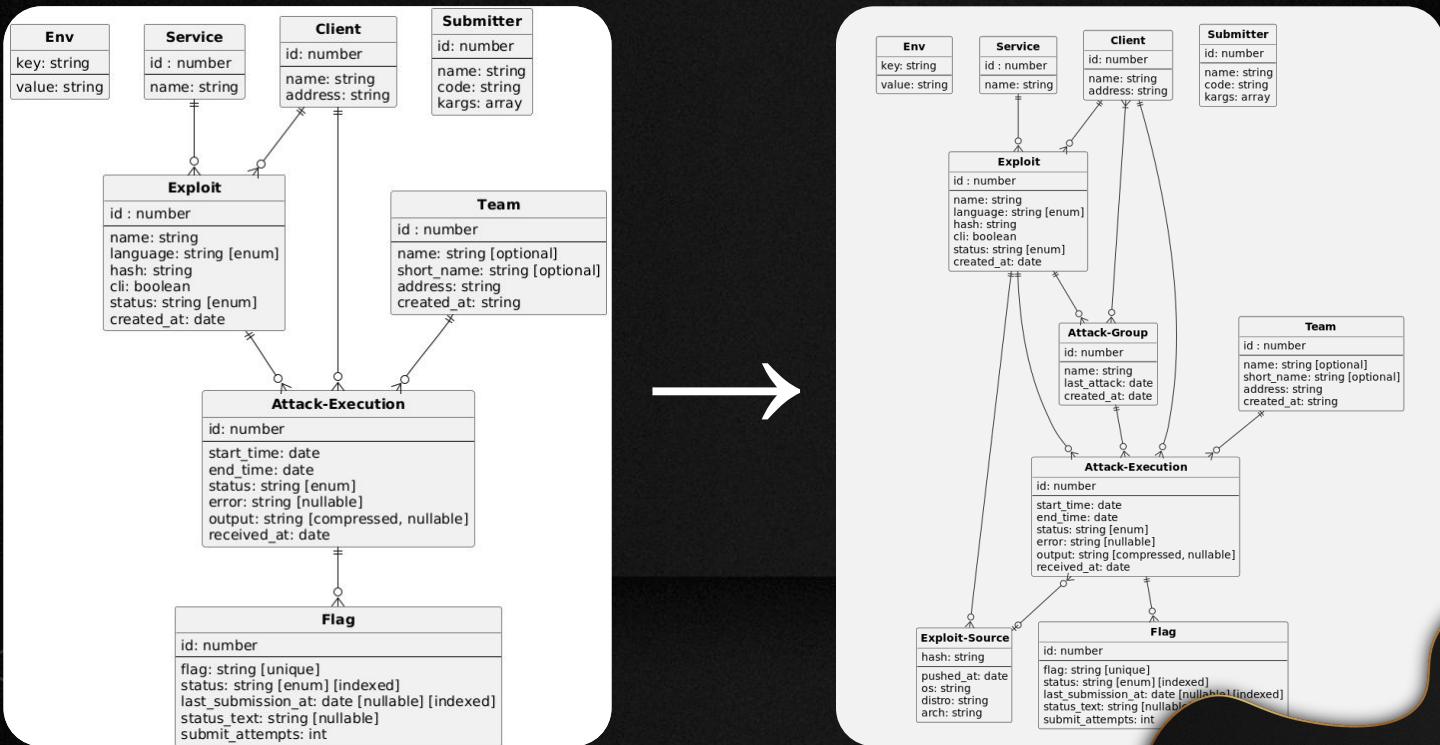
The screenshot shows a digital Kanban board titled "Exploitfarm Shared attacks and Exploit upload". The board is organized into three columns: "Todo", "In Progress", and "Done".

- Todo:** 7 / 100 Estimate: 0
This item hasn't been started
exploitfarm #4
Remote Uploaded exploit start
Exploit Source feature
- In Progress:** 4 / 100 Estimate: 0
This is actively being worked on
exploitfarm #1
Source Upload Feature
Exploit Source feature
- Done:** 1 / 100 Estimate: 0
This has been completed
exploitfarm #2
Download Source feature
Exploit Source feature

Each item card includes a title, priority (e.g., 7/100), estimate (e.g., Estimate: 0), and a list of associated tags (e.g., exploitfarm #4, Remote Uploaded exploit start, Exploit Source, feature). Buttons for "+ Add Item" are visible at the bottom of each column.

RI-PROGETTAZIONE DB

Durante la fase di progettazione delle nuove funzionalità, si è prima fatto brainstorming su idee tecniche di implementazione, e successivamente si è subito provveduto a ristrutturare il database già esistente



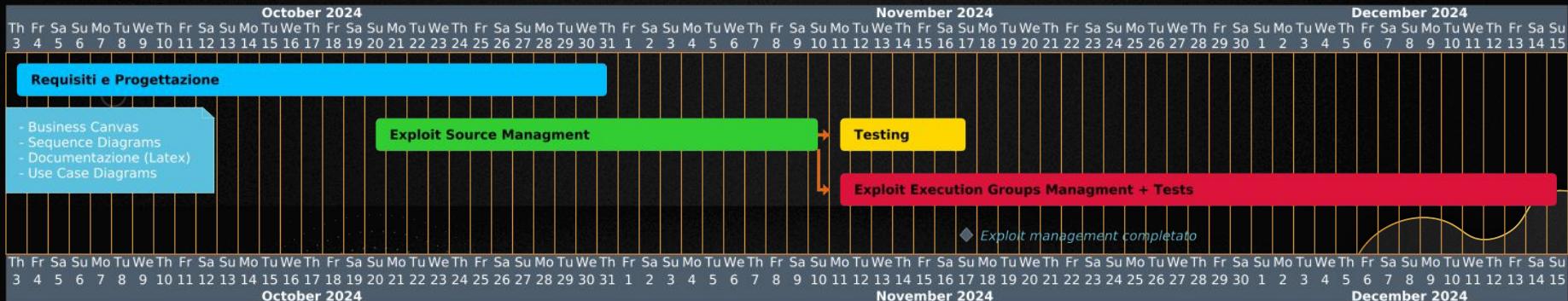
SCHEDULING

Sono definite 3 fasi principali nel progetto:

- Brainstorming, definizione dei requisiti tecnici e documentazione
- 1° Milestone/Sprint: gestione degli exploit
- 2° Milestone/Sprint: gestione dei gruppi di client per gli attacchi condivisi

Le fasi sono definite proprio per la necessità di progettare lo sviluppo stesso dei nuovi requisiti, e dalla mancanza della feature di gestione degli exploit necessaria allo sviluppo della seconda milestone.

- L'inizio del progetto con la prima fase è iniziata il 03-10-2024, la consegna è fissata per il 20-12-2024.





FINE !