

KURS
PENTESTERA

```
$ cat temat_zajęć.txt
```

Wprowadzenie do kursu i branży cyberbezpieczeństwa

04.11.2026

Kacper Powolny

$ cat disclaimer.txt

DISCLAIMER

Na dzisiejszych zajęciach omówione zostaną podstawowe pojęcia z zakresu cyberbezpieczeństwa, w tym techniki i etapy ataków na systemy informatyczne. Wiedza ta służy zrozumieniu, jak działają atakujący, aby skutecznie chronić systemy.
Wykorzystywanie jej wobec systemów, których nie jesteś właścicielem i na które nie masz pisemnej zgody, jest nielegalne. KN White Hats nie popiera i nie zachęca do łamania prawa. Prezentacja służy wyłącznie celom edukacyjnym.

AGENDA

```
$ cat agenda.txt
```

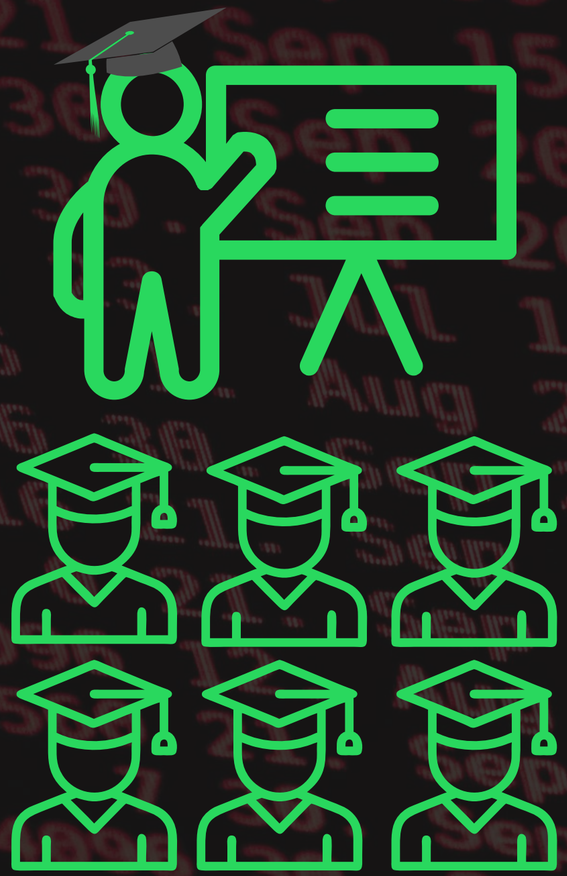
- Idea kursu
- Program kursu
- Kurs vs studia
- Fundamenty cyberbezpieczeństwa
- Czym jest pentest?
- Branża i ścieżki kariery
- Pierwsze kroki

```
$ cat rozdzial_1.txt
```

Idea

KURS
PENTESTERA

KURS PENTESTERA 3.0



KURS 3.0

20 x Lecture

Od studentów dla studentów

20 zajęć, 2 semestry
od zera do pentestera

KURS PENTESTERA

DLA KOGO JEST TEN KURS?



```
$ cat czy_to_dla_mnie.txt
```

- Nie musisz studiować cyberbezpieczeństwa → inne kierunki mile widziane
- Nie musisz nic umieć → zaczynamy od podstaw
- Musisz chcieć → reszty cię nauczymy

KURS PENTESTERA

PO CO TEN KURS?




```
$ cat 1_powod.txt
```

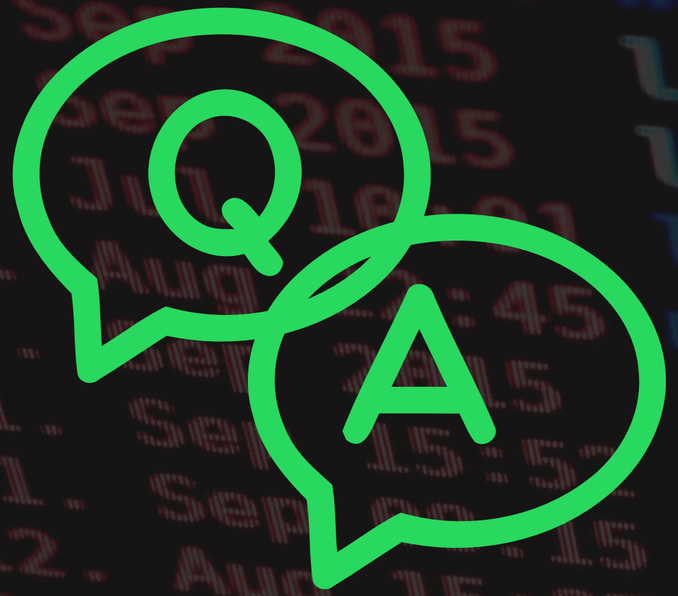
Sprawdzić, czy cyberbezpieczeństwo i IT jest dla ciebie

```
$ cat 2_powod.txt
```

Nauczyć się czegoś więcej niż na studiach i zrozumieć na czym polega praca testera bezpieczeństwa

KURS
PENTESTERA

FILOZOFIA KURSU




```
$ cat 1_filozofia.txt
```

Nie ma głupich pytań ani wstydu z niewiedzy

```
$ cat 2_filozofia.txt
```

Nie umiesz? Naucz się.
Umiesz? Przekaż dalej.



KURS
PENTESTERA

FORMAT ZAJĘĆ



```
$ cat wykład.txt
teoria + przykłady
```

```
$ cat hands-on.txt
robisz zadanie razem
z prowadzącym, na
żywo lub po prostu
oglądasz
```

```
$ cat quiz.txt
5 pytań na koniec,
punkty dodatkowe na
egzamin końcowy
```

OBECNOŚĆ

```
$ cat jest_przypal_bo_jest_lista_wpiszesz_mnie_pls.txt
```

Kurs Pentestera składa się z 20 zajęć, z których każde kończy się krótkim quizem (formularz z kodem QR i hasłem dnia). Obecność nie jest wymagana, aby przystąpić do egzaminu, ale ułatwia jego zaliczenie.

Za zdobycie co najmniej 2 z 5 punktów w quizie uczestnik otrzymuje 0,1 pkt dodatkowego. Przy komplecie zajęć daje to maksymalnie 2 pkt, które wliczają się do progu zaliczenia.

Uwaga: w quizie podajesz imię i nazwisko oraz, jeśli studiujesz na PWr, numer albumu. Na tej podstawie przypisujemy punkty za obecność.

EGZAMIN KOŃCOWY

```
$ cat egzamin_smuteczek.txt
```

Egzamin końcowy to 10 zadań w stylu CTF na platformie White Hats Gym, opartych wyłącznie na materiale z zajęć: 9 zadań o poziomie od easy do easy/medium oraz 1 zadanie hard. Na rozwiązanie zadań przeznaczone są 3 dni, a na napisanie writeupów 1 dzień.

Za każde zadanie można zdobyć 1 pkt: 0,5 pkt za flagę, 0,2 pkt za przedstawienie toku rozumowania, 0,2 pkt za rekomendację remediacji i 0,1 pkt za spełnienie wymagań formalnych writeupu. Zadanie bez poprawnego writeupu otrzymuje 0 pkt.

Próg zaliczenia wynosi 5 pkt z 12 możliwych (10 za egzamin + 2 za obecność).

CERTYFIKAT

```
$ cat certyfikat.txt
```

Każdy uczestnik, który zdobędzie co najmniej 5 pkt, otrzymuje papierowy certyfikat ukończenia Kursu Pentestera, podpisany odręcznie przez Koordynatora Kursu Pentestera oraz Prezesa KN White Hats.

Pracujemy nad tym, aby certyfikat można było również dodać do profilu na LinkedIn.



KONTAKT I INFORMACJE

GRUPA NA FACEBOOKU

STRONA KURSU

SERWER NA DISCORDZIE

SZYBKA ANKIETA

```
$ cat pytanie_1.txt
Kto z was korzystał kiedyś z Linuxa?
```



```
$ cat pytanie_2.txt
Kto wie już, czym chciałby się zajmować w
cyberbezpieczeństwie?
```

```
$ cat pytanie_3.txt
Kto rozwiązywał kiedyś CTF albo maszynę na Hack The Box
lub TryHackMe?
```

```
$ cat rozdzial_2.txt
```

Program kursu

KURS
PENTESTERA

SEMESTR ZIMOWY

```
$ cat plan_kursu.txt
```

1.	Wprowadzenie do kursu i branży	04.11
2.	Systemy operacyjne, wirtualizacja i git	18.11
3.	Podstawy sieci komputerowych	25.11
4.	OSINT i Social Engineering	02.12
5.	Recon aktywny i enumeracja usług	09.12
6.	Skanery podatności i analiza CVE	16.12
7.	Podstawy kryptografii	13.01
8.	Łamanie haseł	20.01
9.	Eskalacja uprawnień	27.01
10.	Praktyczny wstęp do testów penetracyjnych	03.02

KURS
PENTESTERA

SEMESTR LETNI

```
$ cat plan_kursu.txt
```

1.	Web pentest cz. 1	10.03
2.	Web pentest cz. 2	17.03
3.	Web pentest cz. 3	24.03
4.	Wstęp do Active Directory	31.03
5.	Bezpieczeństwo środowisk chmurowych	07.04
6.	Malware, payloady i oszukiwanie AV	21.04
7.	Bezpieczeństwo aplikacji mobilnych	28.04
8.	LLM w testach penetracyjnych	05.05
9.	Forensics, RE, Binary Exploitation	12.05
10.	Aspekty prawne, normy i raportowanie	26.05
11.	Gość specjalny	02.06
12.	Egzamin	09.06

```
$ cat rozdzial_3.txt
```

Kurs vs studia

KURS PENTESTERA

STUDIA W LICZBACH

```
$ cat inżynierka.txt
5335 godzin CNPS
```

```
$ cat magisterka.txt
2285 godzin CNPS
```

```
$ cat gpt_dodaj_mi_pls.txt
7620 godzin CNPS
```

```
$ cat definicja.txt

CNPS (całkowity nakład pracy studenta) - liczba godzin, które według programu studiów trzeba poświęcić na zajęcia i pracę własną, aby uzyskać tytuł inżyniera lub magistra inżyniera.
```

KURS
PENTESTERA

KURS W LICZBACH

```
$ cat semestr_zimowy.txt
20 godzin z nami
```

```
$ cat semestr_letni.txt
24 godziny z nami
```





```
$ cat gpt_dodaj_mi_pls.txt
44 godziny + egzamin
```

PUENTA

```
$ cat porownanie.txt
```

W 44 godziny pokazujemy przekrój tematów ofensywnych, które na studiach są rozłożone na kilka lat. Dzięki temu szybko zobaczysz, co cię najbardziej interesuje, i łatwiej wybierzesz swoją ścieżkę.

$ cat rozdzial_4.txt

Fundamenty bezpieczeństwa

PO CO TO PENTESTEROWI?

```
$ cat dlaczego.txt
```

- Bez podstaw nie zrozumiesz, co i dlaczego atakujesz
- Na tych pojęciach opierają się metodyki testów i raportów
- Wspólny język z zarządem, osobami nietechnicznymi i blue teamem



CZYM JEST CYBERBEZPIECZEŃSTWO?

```
$ cat definicja.txt
```

Cyberbezpieczeństwo to odporność systemów informacyjnych na działania naruszające poufność, integralność, dostępność i autentyczność przetwarzanych danych lub związanych z nimi usług oferowanych przez te systemy.



DLACZEGO TO WAŻNE?

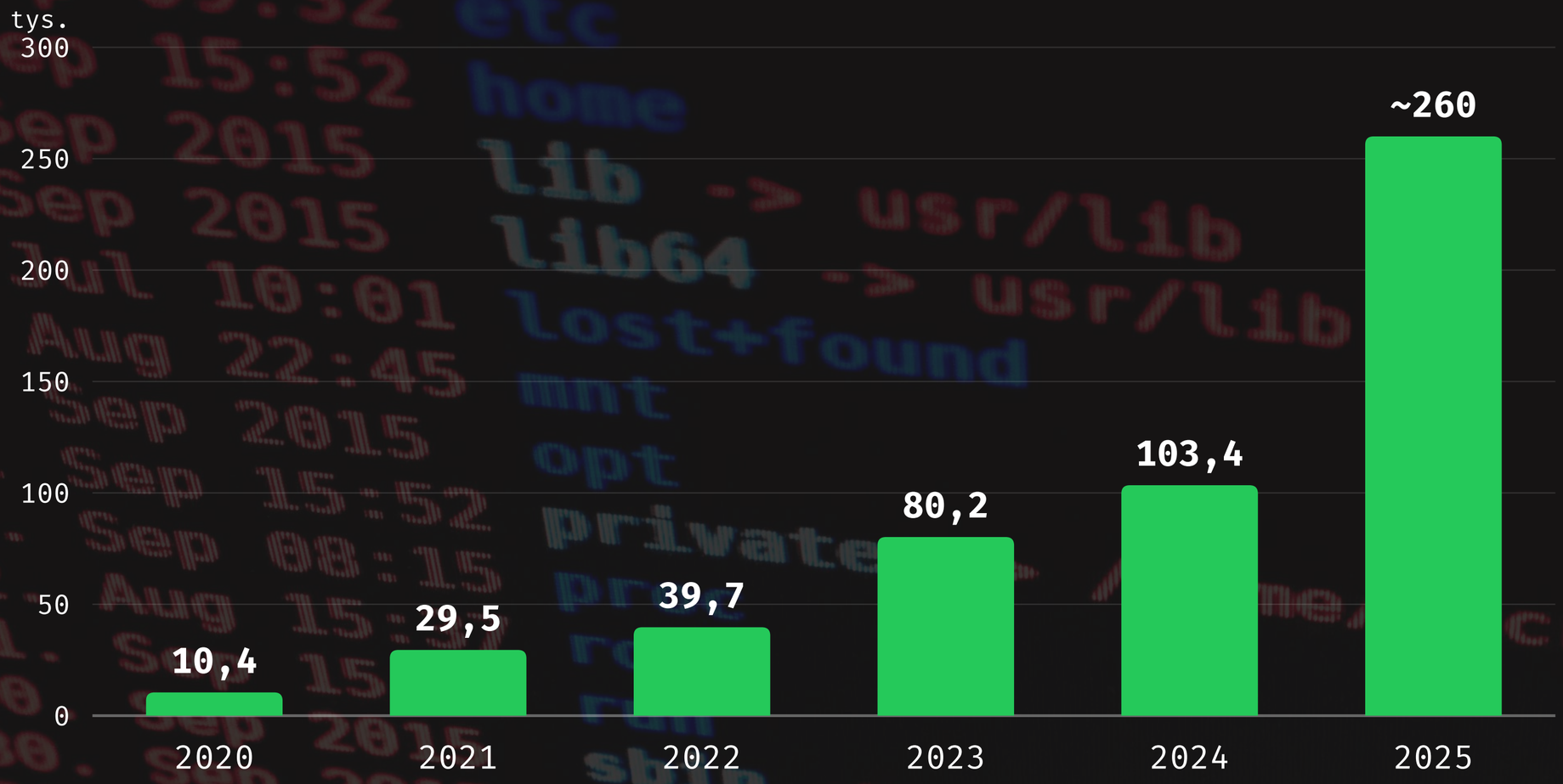
```
$ cat wszystko_jest_cyfrowe.txt
```
- banki
- szpitale
- poczta
- elektrownie
- wodociągi
- urzędy
- firmy
- uczelnie

```
$ cat skala.txt
```
CERT Polska w 2025 roku obsługiwał średnio około 30 incydentów na godzinę

```
$ cat konsekwencje.txt
```
- utrata pieniędzy
- kradzież danych
- kradzież tożsamości
- szpiegowanie
- cyberwojna
- zagrożenie zdrowia i życia
- sabotaż infrastruktury krytycznej



WZROST LICZBY ATAKÓW



Incydenty obsłużone przez CERT Polska (CSIRT NASK), w tysiącach

Źródło: raporty roczne CERT Polska 2020-2025

CO CHRONIMY?

```
$ cat uslugi.txt
```
- bankowość
- opieka zdrowotna
- energia
- transport

```
$ cat ludzie.txt
```
- majątek
- prywatność
- godność
- wolność
- zdrowie i życie



```
$ cat dane.txt
```
- dane osobowe
- hasła
- tajemnice firm
- wyniki badań

```
$ cat systemy.txt
```
- serwery
- sieci
- chmura
- urządzenia końcowe

KTO PILNUJE CYBERBEZPIECZEŃSTWA?

```
$ cat polska.txt
```

- CSIRT NASK (CERT Polska)
- CSIRT GOV (ABW)
- CSIRT MON (Wojska Obrony Cyberprzestrzeni)
- CBZC (policja)
- Ministerstwo Cyfryzacji



```
$ cat swiat.txt
```

- CISA (USA)
- NCSC (Wielka Brytania)
- FIRST (opiekun CVSS i EPSS)
- MITRE (program CVE)

```
$ cat unia.txt
```

- ENISA
- CERT-EU

NORMY, STANDARDY, PRZEPISY

```
$ cat normy.txt
```

Normy (dobrowolne):

- ISO/IEC 27001
- ISO/IEC 27002

```
$ cat przepisy.txt
```

Przepisy (obowiązkowe):

- dyrektywa NIS2 (UE)
- ustawa o KSC
- RODO (rozporządzenie UE)
- DORA (rozporządzenie UE, sektor finansowy)

```
$ cat standardy.txt
```

Standardy i frameworki (dobre praktyki):

- NIST CSF
- OWASP
- MITRE ATT&CK



NIESZCZĘSNE KAPELUSZE :)

```
$ cat white_hat.txt
White hat - działa legalnie, za zgodą właściciela systemu → (pentester, bug bounty hunter)
```



NIESZCZĘSNE KAPELUSZE :)

```
$ cat grey_hat.txt
Grey hat - włamuje się bez zgody, ale
bez złych zamiarów, często potem
zgłasza lukę → (nadal może łamać
prawo)
```



NIESZCZĘSNE KAPELUSZE :)

```
$ cat black_hat.txt
Black hat - włamuje się bez zgody dla
zysku lub szkody → (cyberprzestępca)
```



KIM SĄ ATAKUJĄCY (1/5)?

```
$ cat script_kiddies.txt
```

Script kiddies - używają gotowych narzędzi bez zrozumienia, jak działają → (teoretycznie najmniej groźni, ale najliczniejsi i najbardziej nieobliczalni)



KIM SĄ ATAKUJĄCY (2/5)?

```
$ cat haktywisci.txt
```

Haktywiści - atakują z pobudek politycznych lub ideologicznych → (DDoS, podmiana stron, wycieki)



KIM SĄ ATAKUJĄCY (3/5)?

```
$ cat cyberprzestepcy.txt
```

Cyberprzestępcy - działają dla pieniędzy, często jak zorganizowane firmy → (grupy ransomware, phishing)



KIM SĄ ATAKUJĄCY (4/5)?

```
$ cat insiderzy.txt
```

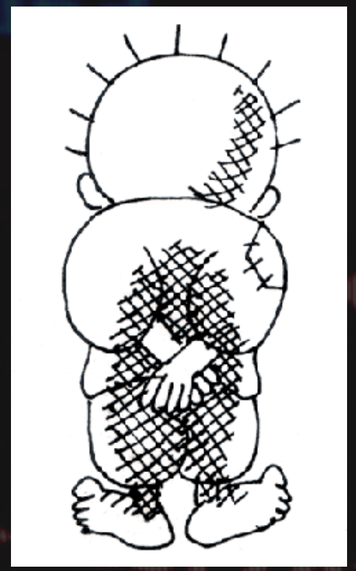
Insiderzy - pracownicy lub osoby z dostępem od środka → (zemsta, przekupstwo albo zwykła nieuwaga)



KIM SĄ ATAKUJĄCY (5/5)?

```
$ cat apt.txt
```

APT (Advanced Persistent Threat) - grupy wspierane przez państwa, cierpliwe i dobrze finansowane → (szpiegostwo, sabotaż)

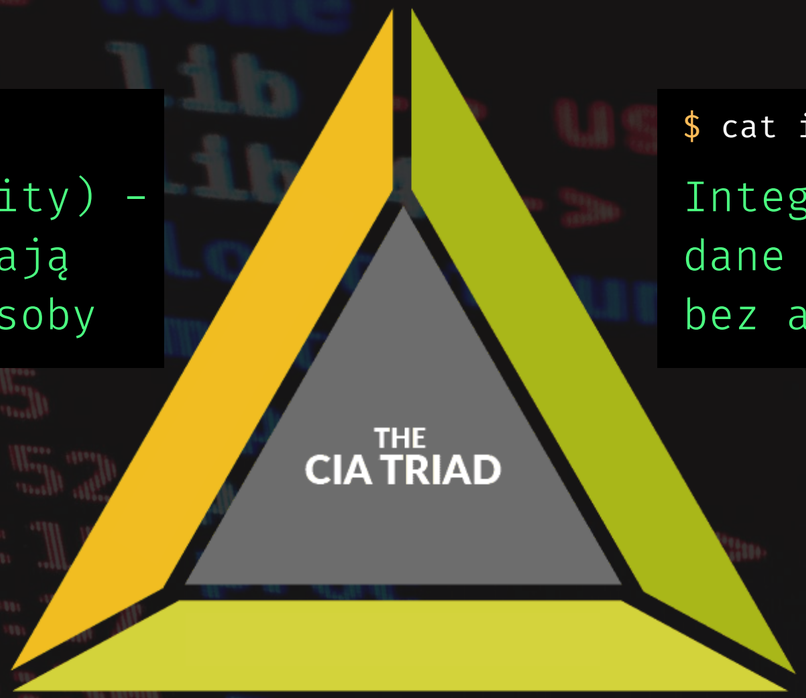


LAZARUS

TRIADA CIA

```
$ cat confidentiality.txt
```

Poufność (Confidentiality) - dostęp do informacji mają wyłącznie uprawnione osoby




```
$ cat integrity.txt
```

Integralność (Integrity) - dane nie mogą zostać zmienione bez autoryzacji

```
$ cat availability.txt
```

Dostępność (Availability) - system i dane są dostępne wtedy, gdy są potrzebne

PARKERIAN HEXAD I?

```
$ cat possession.txt
```

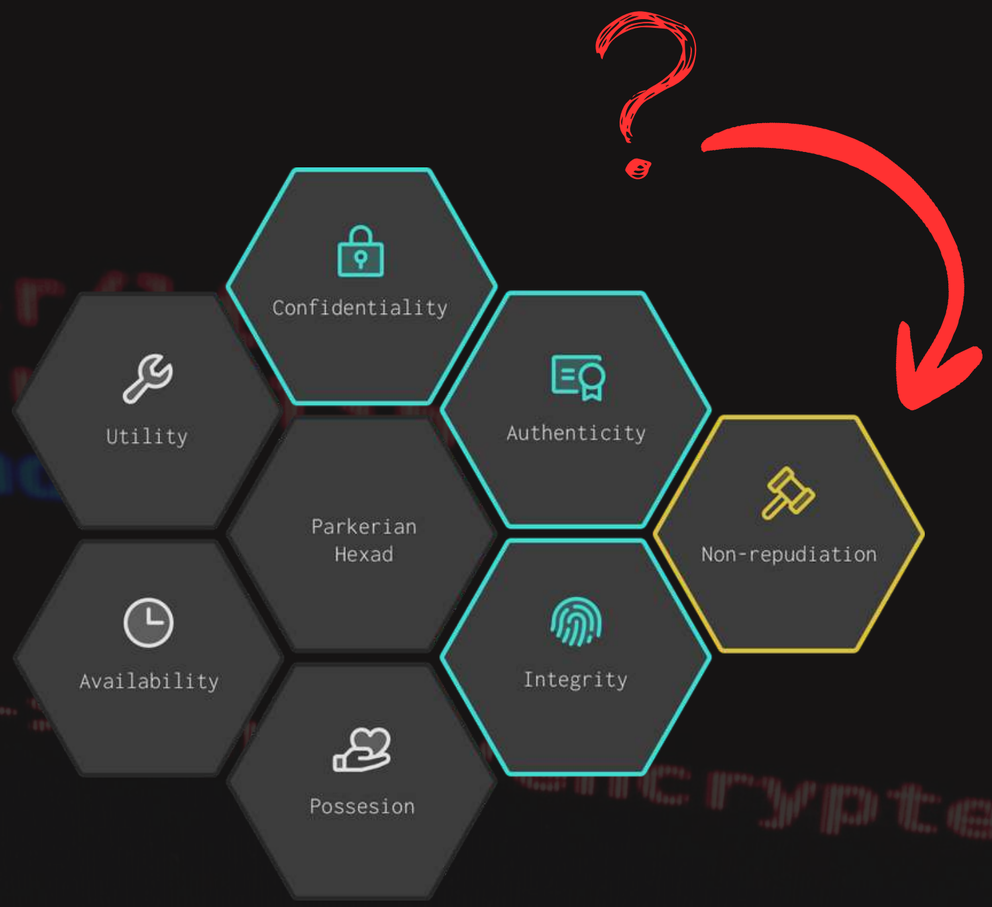
Kontrola (Possession) - właściciel ma władzę nad nośnikiem lub kopią danych

```
$ cat authenticity.txt
```

Autentyczność (Authenticity) - informacja pochodzi od tego, od kogo deklaruje

```
$ cat utility.txt
```

Użyteczność (Utility) - dane da się faktycznie wykorzystać




```
$ cat parker.txt
```

Donn Parker (1998) uznał, że triada CIA nie opisuje wszystkich sytuacji, i dodał do niej trzy własności.

NIEZAPRZECZALNOŚĆ




```
$ cat non_repudiation.txt
```

Niezaprzeczalność (Non-repudiation) - autor działania nie może się go wyprzeć, bo istnieje niepodważalny dowód, że to on je wykonał

KURS
PENTESTERA

MODELOWANIE ZAGROŻEŃ

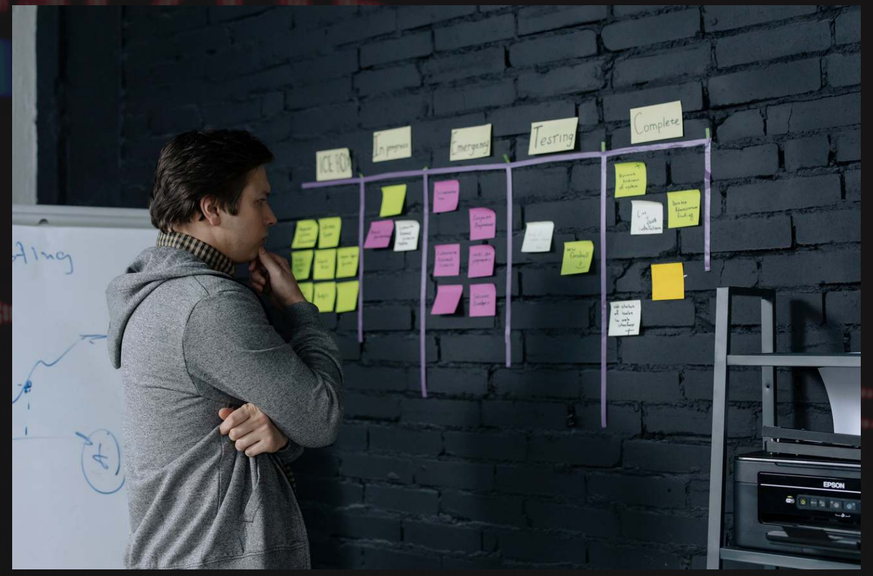
```
$ cat definicja.txt
```

Modelowanie zagrożeń (Threat Modeling) - systematyczne szukanie tego, co może pójść źle, zanim zrobi to atakujący - dosłownie staramy się wyprzedzić atakującego.

```
$ cat shostack.txt
```

Cztery pytania Adama Shostacka:

1. Nad czym pracujemy?
2. Co może pójść źle?
3. Co z tym zrobimy?
4. Czy dobrze nam poszło?



STRIDE (1/2)

```
$ cat stride.txt
```

Model Microsoftu z 1999 r. Każda litera to jedno zagrożenie i własność, którą ono narusza.

```
$ cat spoofing.txt
```

Spoofing - podszywanie się pod kogoś → (narusza uwierzytelnianie)

```
$ cat tampering.txt
```

Tampering - nieuprawniona zmiana danych → (narusza integralność)

```
$ cat repudiation.txt
```

Repudiation - wypieranie się wykonanych działań → (narusza niezaprzeczalność)



STRIDE (2/2)

```
$ cat information_disclosure.txt
```

Information disclosure - wyciek informacji → (narusza poufność)

```
$ cat denial_of_service.txt
```

Denial of Service - odmowa usługi → (narusza dostępność)

```
$ cat elevation_of_privilege.txt
```

Elevation of privilege - zdobycie wyższych uprawnień → (narusza autoryzację)



DREAD

```
$ cat dread.txt

Jest to model oceny ryzyka zagrożenia.
Każde kryterium oceniamy w skali od 0 do 10, a wynik to średnia:
Damage - jak duże szkody?
Reproducibility - jak łatwo powtórzyć atak?
Exploitability - jak łatwo go przeprowadzić?
Affected users - ilu użytkowników dotyczy?
Discoverability - jak łatwo znaleźć podatność?
```



```
$ cat dlaczego_nie.txt

Każda dokonana ocena jest subiektywna, bo dwie osoby dadzą
praktycznie zawsze różne wyniki.

Sam Microsoft przestał go stosować, a w praktyce zastąpił go
CVSS.
```



PASTA

```
$ cat pasta.txt
```

PASTA - **P**rocess for **A**ttack **S**imulation and **T**hreat **A**nalysis, składa się z:

1. Zdefiniowanie celów
2. Zdefiniowanie zakresu technicznego (scope)
3. Dekompozycja aplikacji i jej analiza
4. Analiza zagrożeń
5. Analiza podatności i słabości
6. Modelowanie ataków i ich symulacja
7. Analiza ryzyka i zarządzanie nim



STRIDE VS DREAD VS PASTA

```
$ cat stride.txt
STRIDE - co może pójść źle? → (klasyfikacja)
```

```
$ cat dread.txt
DREAD - jak bardzo może być źle? → (ocena)
```

```
$ cat pasta.txt
PASTA - co to oznacza dla biznesu i $$$? → (proces)
```



VULNERABILITY, EXPLOIT, PAYLOAD

```
$ cat vulnerability.txt
```

Vulnerability (podatność) - słabość w systemie → (wada zamka)

```
$ cat exploit.txt
```

Exploit - sposób wykorzystania podatności → (wytrych)

```
$ cat payload.txt
```

Payload - kod wykonywany po udanym ataku → (to, co robisz po wejściu do środka)



```
$ cat przyklad.txt
```

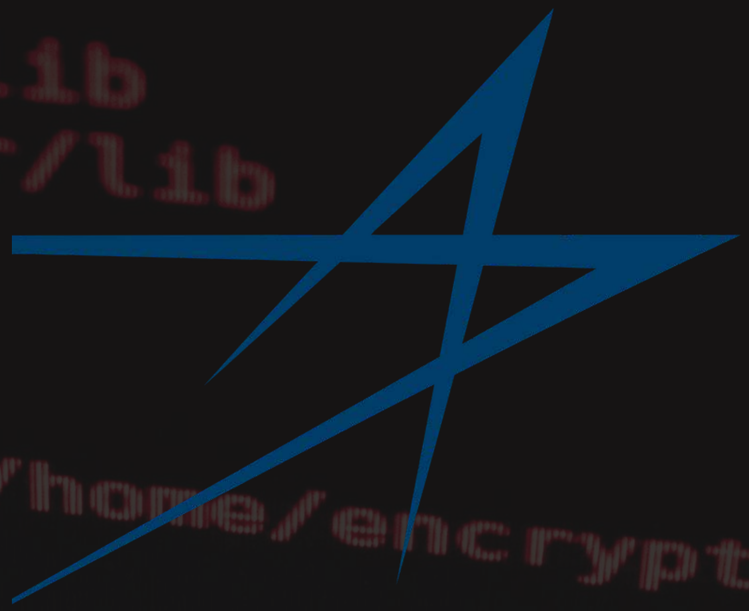
Podatność w SMBv1 (CVE-2017-0144) → exploit EternalBlue → payload: ransomware WannaCry (2017)

CYBER KILL CHAIN



```
$ cat kill_chain.txt
```

- Model Lockheed Martin z 2011 r., zaczerpnięty z wojska
- Atak to łańcuch 7 etapów
- Przerwij jedno ogniwo, a zatrzymasz atak



CYBER KILL CHAIN: PRZYGOTOWANIE

```
$ cat reconnaissance.txt
```

1. Rekonesans - zbieranie informacji na temat celu → (LinkedIn, adresy mailowe pracowników)

```
$ cat weaponization.txt
```

2. Uzbrojenie - przygotowanie narzędzia ataku → (dokument ze złośliwym payloadem)

```
$ cat delivery.txt
```

3. Dostarczenie - przekazanie go ofierze → (e-mail “Faktura_10_2026.docm”)



CYBER KILL CHAIN: ATAK

```
$ cat exploitation.txt
```

4. Eksploitacja - uruchomienie złośliwego kodu → (użytkownik aktywuje payload)

```
$ cat installation.txt
```

5. Instalacja - zapewnienie sobie stałego dostępu → (backdoor)



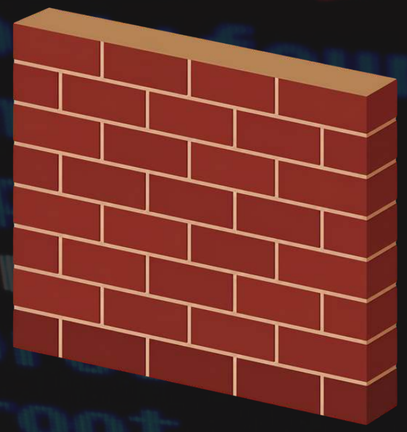
```
$ cat command_and_control.txt
```

6. Command & Control - zdalne sterowanie przejętym systemem → (połączenie z serwerem atakującego)

CYBER KILL CHAIN: CEL I OGRANICZENIA

```
$ cat actions_on_objectives.txt
```

7. Działania na celu - to, po co atakujący przyszedł → (kradzież danych, szyfrowanie danych)



```
$ cat ograniczenia.txt
```

Model skupia się bardziej na malware i atakach z zewnątrz.
Dość słabo opisuje insiderów i ruch wewnątrz sieci, m.in. dlatego powstało MITRE ATT&CK.

IOC, IOA, TTP

```
$ cat ioc.txt
```

IOC (Indicator of Compromise) - ślad po ataku, np. hash pliku, adres IP, domena → (co się stało)

```
$ cat ioa.txt
```

IOA (Indicator of Attack) - podejrzane zachowanie w trakcie ataku, np. dokument Worda uruchamia CMD → (co się dzieje)

```
$ cat ttp.txt
```

TTP (Tactics, Techniques, Procedures) - opis sposobu działania przeciwnika: cel, technika i konkretne wykonanie

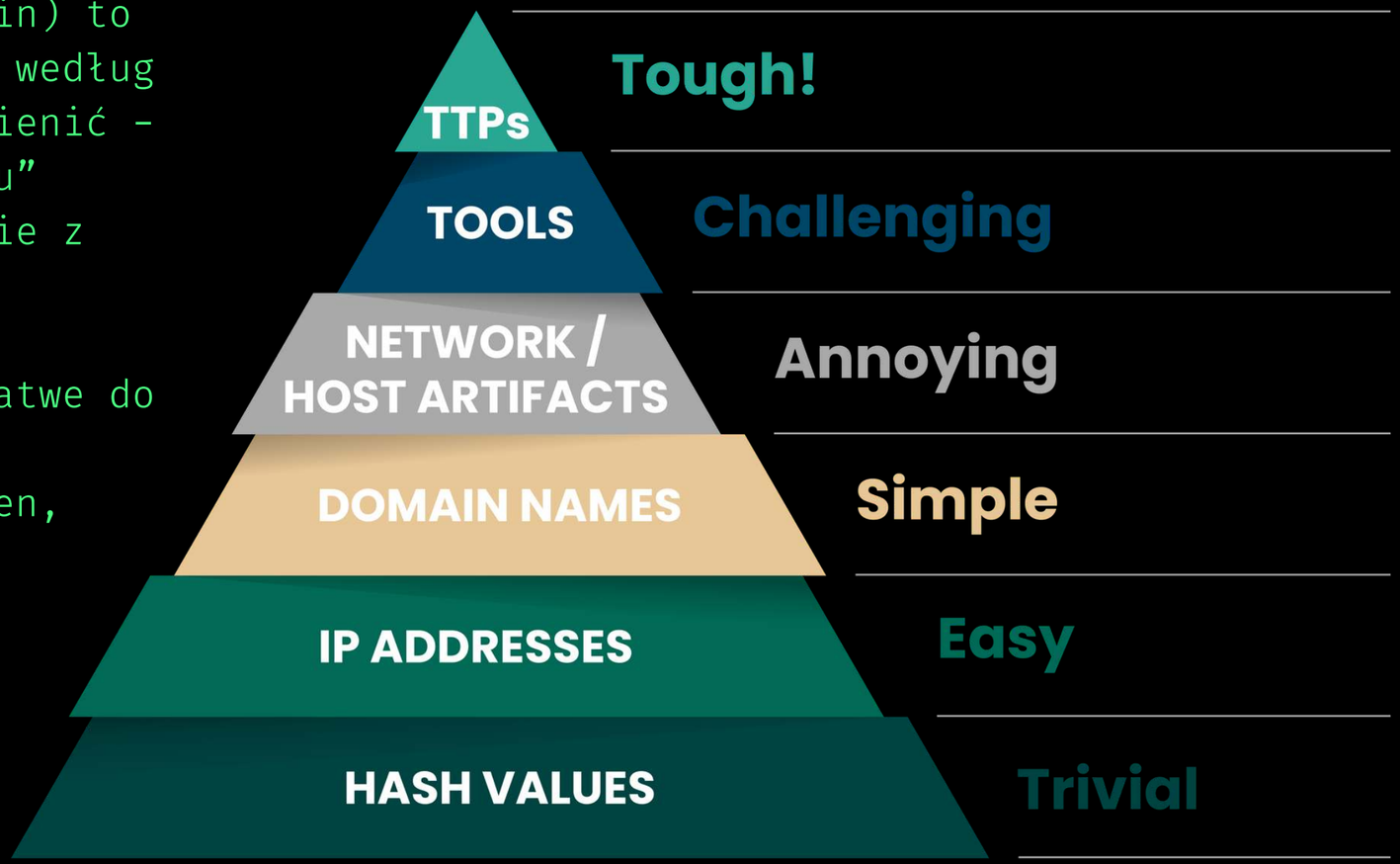


PIRAMIDA BÓLU

```
$ cat czuje_tylko_bool_rozumiecie_hehe_bo_nie_bol_tylko_bool_pls_niech_ktos_sie_zasmieje_:((((.txt
```

Piramida bólu (ang. Pyramid of Pain) to model, który porządkuje wskaźniki według tego, jak trudno atakującym je zmienić - im wyższy poziom, tym więcej “bólu” sprawia atakującym poradzenie sobie z nimi.

Model obejmuje zarówno elementy łatwe do zmiany, takie jak wyniki funkcji haszującej, adresy IP i nazwy domen, jak i najtrudniejszy, najbardziej dotkliwy poziom: TTP.



CVE, CVSS, EPSS

```
$ cat cve.txt

CVE - unikalny identyfikator podatności, prowadzony przez MITRE →
(np. CVE-2021-44228, czyli Log4Shell)
```

```
$ cat cvss.txt

CVSS - jak groźna jest podatność, w skali od 0 do 10 →
(np. wspomniany wyżej Log4Shell: 10.0)
```

```
$ cat epss.txt

EPSS - prawdopodobieństwo wykorzystania podatności w ciągu 30 dni, w
skali od 0 do 1 → (CVSS i EPSS rozwija FIRST)
```



SKALA CVSS

Severity Rating	CVSS 3.1 Score	Description
CRITICAL	9.0 - 10	Exploitation of the vulnerability allows an attacker administrative-level access to systems and/or high-level data that would catastrophically impact the organization. Vulnerabilities marked CRITICAL require immediate attention and must be fixed without delay, especially if they occur in a production environment.
HIGH	7.0 – 8.9	Exploitation of the vulnerability makes it possible to access high-value data. However, there are certain pre-requisites that need to be met for the attack to be successful. These vulnerabilities should be reviewed and remedied wherever possible.
MEDIUM	4.0 – 6.9	Exploitation of the vulnerability might depend on external factors or other conditions that are difficult to achieve, like requiring user privileges for a successful exploitation. These are moderate security issues that require some effort to successfully impact the environment.
LOW	0.1 – 3.9	Vulnerabilities in the low range typically have very little impact on an organization's business. Exploitation of such vulnerabilities usually requires local or physical system access and depends on conditions that are very difficult to achieve practically.
INFORMA-TIONAL	0.0	These vulnerabilities represent significantly less risk and are informational in nature. These items can be remediated to increase security.

DLACZEGO NIE WYSTARCZY SAMO CVSS?

```
$ cat priorytety.txt
```

1. Tylko niewielka część opublikowanych CVE jest kiedykolwiek wykorzystywana w praktyce
2. CVSS mówi, jak źle byłoby, gdyby ktoś wykorzystał podatność, a EPSS, czy ktoś to zrobi
3. W praktyce priorytety ustala się na podstawie CVSS, EPSS i katalogu CISA KEV (lista podatności wykorzystywanych w realnych atakach)



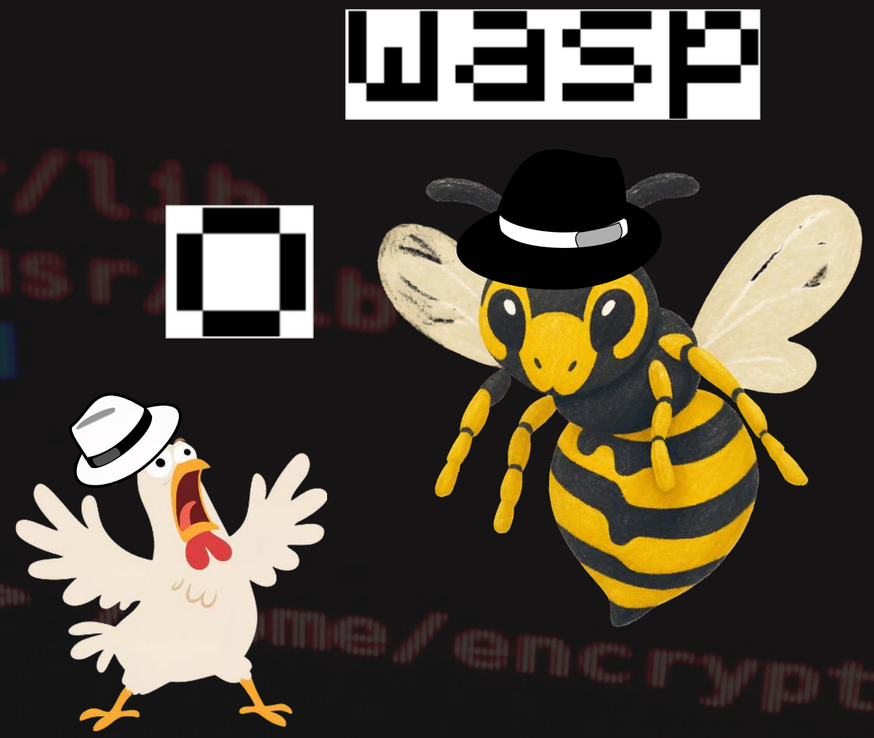
OWASP

```
$ cat owasp.txt
```

OWASP (Open Worldwide Application Security Project) - fundacja non-profit, która tworzy darmowe standardy, narzędzia i materiały o bezpieczeństwie aplikacji

```
$ cat projekty.txt
```

- Top 10 - najczęstsze ryzyka aplikacji webowych
- API Security Top 10, Mobile Top 10, Top 10 for LLM - to samo dla API, aplikacji mobilnych i wielkich modeli językowych
- ASVS - lista wymagań bezpieczeństwa aplikacji
- WSTG - przewodnik, jak testować aplikacje webowe
- ZAP - darmowy skaner aplikacji webowych



OWASP TOP 10:2025

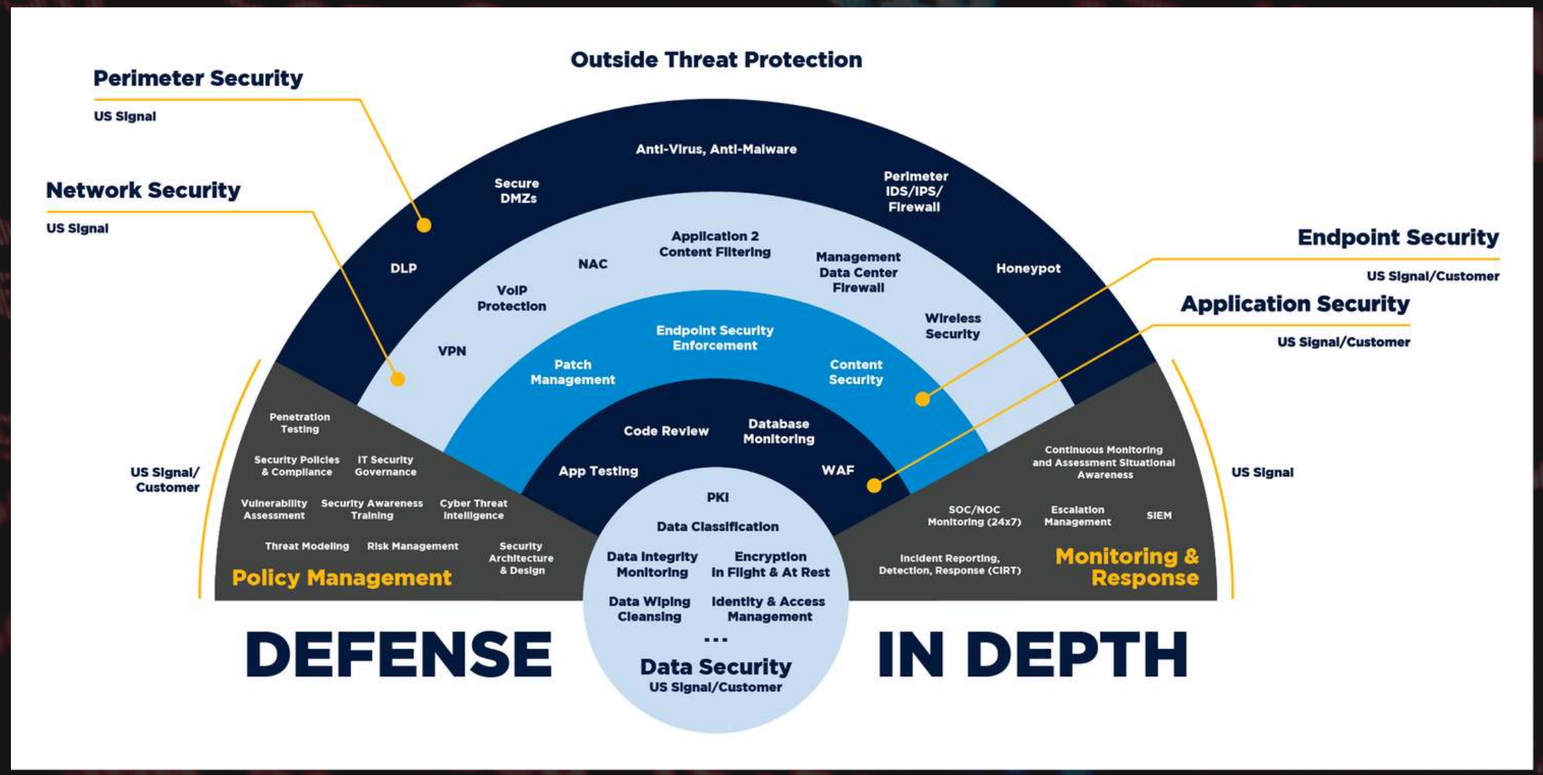
```
$ cat top10.txt
```

A01 - Broken Access Control - dostajesz się do czegoś, do czego nie masz uprawnień → (zmieniasz id w linku i widzisz cudze konto)
A02 - Security Misconfiguration - coś jest źle skonfigurowane → (domyślne hasło admin/admin, włączony tryb debug)
A03 - Software Supply Chain Failures - dziurawa lub zainfekowana biblioteka, z której korzysta aplikacja → (nowe, podszywające się biblioteki)
A04 - Cryptographic Failures - dane niezaszyfrowane albo słabo zaszyfrowane → (hasła w MD5, HTTP zamiast HTTPS)
A05 - Injection - aplikacja traktuje dane od użytkownika jak kod → (SQL injection, XSS)
A06 - Insecure Design - błąd jest już w samym projekcie → (reset hasła przez pytanie “imię twojego psa”)
A07 - Authentication Failures - słabe logowanie i sesje → (można zgadywać hasła bez końca)
A08 - Software or Data Integrity Failures - aplikacja ufa danym, nie sprawdzając, czy ktoś ich nie zmienił → (niepodpisane aktualizacje)
A09 - Security Logging & Alerting Failures - atak trwa, a nikt tego nie widzi → (brak logów i alertów)
A10 - Mishandling of Exceptional Conditions - aplikacja źle reaguje na błędy → (pokazuje szczegóły błędu albo przy awarii wpuszcza wszystkich)

DEFENSE IN DEPTH

```
$ cat defense_in_depth.txt
```

Obrona z użyciem warstw: fizyczna → sieć → host → aplikacja → dane → ludzie i procedury.
Model szwajcarskiego sera: każda warstwa ma dziury, ale rzadko układają się w jednej linii.





ZERO TRUST

```
$ cat zamek_i_fosa.txt
```

Model klasyczny: zamek i fosa → (kto jest w środku sieci, temu ufamy)



```
$ cat zero_trust.txt
```

Zero trust: nigdy nie ufaj, zawsze weryfikuj → (każde żądanie sprawdzane osobno)



```
$ cat zasady.txt
```

Zasady:

- weryfikujemy każde żądanie
- przydzielamy najmniejsze potrzebne uprawnienia
- zakładamy, że do włamania już doszło, i ograniczamy jego skutki

CO TO OZNACZA DLA PENTESTERA?

```
$ cat did.txt

Defense in depth → (jedna podatność rzadko
wystarcza, liczą się łańcuchy podatności →
więc ich szukamy)
```

```
$ cat zt.txt

Zero trust → (testy “assumed breach”:
zakładamy, że atakujący już jest w środku,
i sprawdzamy, jak daleko zajdzie)
```



```
$ cat rozdzial_5.txt
```

Czym jest pentest?

CZYM JEST PENTEST?

```
$ cat definicja.txt

Test penetracyjny (pentest) - autoryzowana symulacja
ataku na system, prowadzona w ustalonym zakresie (scope)
i czasie, której celem jest znalezienie i potwierdzenie
jak największej liczby podatności, zakończona raportem z
rekomendacjami
```



ZGODA, ZAKRES, RAPORT

```
$ cat zgoda.txt
```

Zgoda → (bez niej to przestępstwo, a nie pentest)

```
$ cat zakres.txt
```

Zakres → (testujemy tylko to, na co się umówiliśmy)

```
$ cat raport.txt
```

Raport → (to, za co płaci klient)



PENTEST VS INNE RODZAJE TESTÓW

```
$ cat vulnerability_assessment.txt
```

Vulnerability assessment - szeroko: znajdź i wypisz podatności, często automatycznie, bez ich wykorzystywania → (lista)

```
$ cat pentest.txt
```

Pentest - głęboko w zakresie: potwierdzenie podatności przez ich wykorzystanie, zespół klienta wie o teście → (raport z dowodami)

```
$ cat red_team.txt
```

Red team - konkretny cel, działanie po cichu, sprawdza też, czy obrońcy wykryją atak i jak zareagują → (zrealizowany scenariusz)



BUG BOUNTY

```
$ cat bug_bounty.txt
```

Bug bounty to program, w którym organizacja (firma, instytucja) nagradza osoby, które znajdą i zgłoszą podatności w jej systemach, zgodnie z zasadami programu.

Daje to legalną drogę do szukania podatności, zanim znajdą je przestępcy.

Nagrody mogą mieć formę finansową, rzeczową lub publicznego uznania, zależnie od wagi błędu.

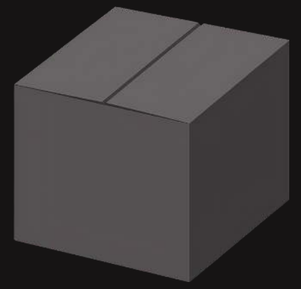
Programy prowadzone są m.in. na HackerOne, Bugcrowd i Intigriti.



BLACK, GREY, WHITE BOX

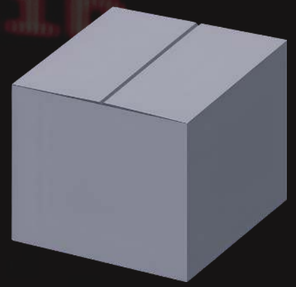
```
$ cat black_box.txt
```

Black box - tester nie wie nic o systemie → (perspektywa atakującego z zewnątrz)



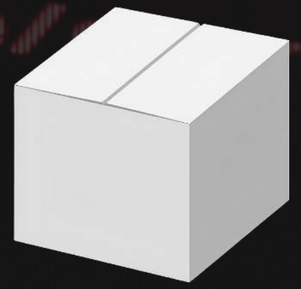
```
$ cat grey_box.txt
```

Grey box - tester ma częściową wiedzę, np. konto użytkownika → (najczęstszy wariant)



```
$ cat white_box.txt
```

White box - tester ma pełny dostęp, łącznie z kodem i dokumentacją → (najdokładniejszy)



RODZAJE PENTESTÓW

```
$ cat rodzaje.txt
```

- Socjotechnika → (semestr zimowy 4)
- Infrastruktura sieciowa: zewnętrzna i wewnętrzna → (semestr zimowy 3, 5, 6)
- Aplikacje webowe i API → (semestr letni 1-3)
- Active Directory → (semestr letni 4)
- Chmura i kontenery → (semestr letni 5)
- Aplikacje mobilne → (semestr letni 7)
- Aplikacje z LLM → (semestr letni 8)



PISEMNA ZGODA I ZAKRES

```
$ cat rules_of_engagement.txt
```

- zakres: adresy IP, domeny, aplikacje
- co jest wyłączone z testów
- okno czasowe testów
- dozwolone techniki, np. czy wolno robić DoS albo socjotechnikę
- czy pracujemy na klonie środowiska, czy na produkcji
- kontakt awaryjny, gdyby coś przestało działać
- osoba kontaktowa po stronie klienta i sposób zmiany zakresu w trakcie testu

```
$ cat prawo.txt
```

Działanie poza zakresem to nieuprawniony dostęp, czyli przestępstwo (art. 267 kk), nawet w trakcie zleconego pentestu.



PRZEBIEG PENTESTU

```
$ cat fazy.txt
```

1. Ustalanie zakresu i zasad
2. Podpisanie umowy
3. Rekonesans
4. Enumeracja usług
5. Identyfikacja podatności
6. Eksploitacja
7. Eskalacja uprawnień i dalszy ruch w sieci
8. Raport
9. Konsultacje (opcjonalne)
10. Otrzymanie wynagrodzenia

*. Potencjalne znalezienie “stałego klienta”, jeżeli dobrze się spisaliśmy



PRZEBIEG PENTESTU = MAPA KURSU



```
$ cat mapa_kursu.txt
```

- Ustalenie zakresu → (semestr zimowy 1, semestr letni 10)
- Rekonesans → (semestr zimowy 4)
- Enumeracja → (semestr zimowy 5)
- Identyfikacja podatności → (semestr zimowy 6, semestr letni 1-3)
- Eksploitacja → (semestr zimowy 10, semestr letni 1-3)
- Eskalacja i ruch w sieci → (semestr zimowy 9, semestr letni 4)
- Raport → (semestr letni 10)

MINDSET PENTESTERA

```
$ cat ciekawosc.txt
```

Ciekawość → (“a co się stanie, jeśli…?”)

```
$ cat upor.txt
```

Upór → (możesz niczego nie znaleźć przez dwa dni)

```
$ cat etyka.txt
```

Etyka → (bez zgody to nie pentest, tylko przestępstwo)



```
$ cat dokumentacja.txt
```

Dokumentacja → (czego nie zapisałeś, tego nie było; raport to twój produkt)

$ cat rozdzial_6.txt

Branża i ścieżki kariery

MAPA RÓL W CYBERBEZPIECZEŃSTWIE



```
$ cat ofensywa.txt
```

Ofensywa → (pentester, red teamer, bug bounty hunter, security researcher)



```
$ cat defensywa.txt
```

Defensywa → (analityk SOC, IR, threat hunter, specjalista forensics, inżynier bezpieczeństwa)



```
$ cat zarzadzanie.txt
```

Zarządzanie i budowanie → (GRC, audytor, AppSec / DevSecOps, cloud security)

RED, BLUE, PURPLE TEAM



```
$ cat red_team.txt
```

Red team - wciela się w atakujących i sprawdza, czy da się osiągnąć cel → (pentesty, red teaming)



```
$ cat blue_team.txt
```

Blue team - broni: monitoruje, wykrywa i reaguje → (SOC, IR, threat hunting)



```
$ cat purple_team.txt
```

Purple team - red i blue pracują razem: jedni atakują, drudzy na bieżąco uczą się to wykrywać → (najszybsza poprawa obrony)

KURS PENTESTERA

BLUE TEAM OD ŚRODKA



```
$ cat soc.txt
```

SOC (Security Operations Center) - monitoruje alerty 24/7, analitycy L1-L3 → (częsta pierwsza praca w branży)



```
$ cat ir.txt
```

IR (Incident Response) - gdy dojdzie do incydentu: ogranicza go, usuwa atakującego i przywraca działanie systemów → (bardzo stresująca praca pod presją czasu i pełna dostępność 24/7)



```
$ cat threat_hunting.txt
```

Threat hunting - aktywnie szuka atakujących, których nie wykryły zabezpieczenia → (polowanie na TTP z piramidy bólu)

POZOSTAŁE ŚCIEŻKI



```
$ cat forensics.txt
```

Forensics (informatyka śledcza) - zabezpiecza i analizuje dowody cyfrowe: dyski, pamięć, logi → (firmy, organy ścigania)



```
$ cat grc.txt
```

GRC (Governance, Risk, Compliance) - polityki bezpieczeństwa, analiza ryzyka, zgodność z ISO 27001, NIS2, RODO → (najmniej techniczna, duży popyt)

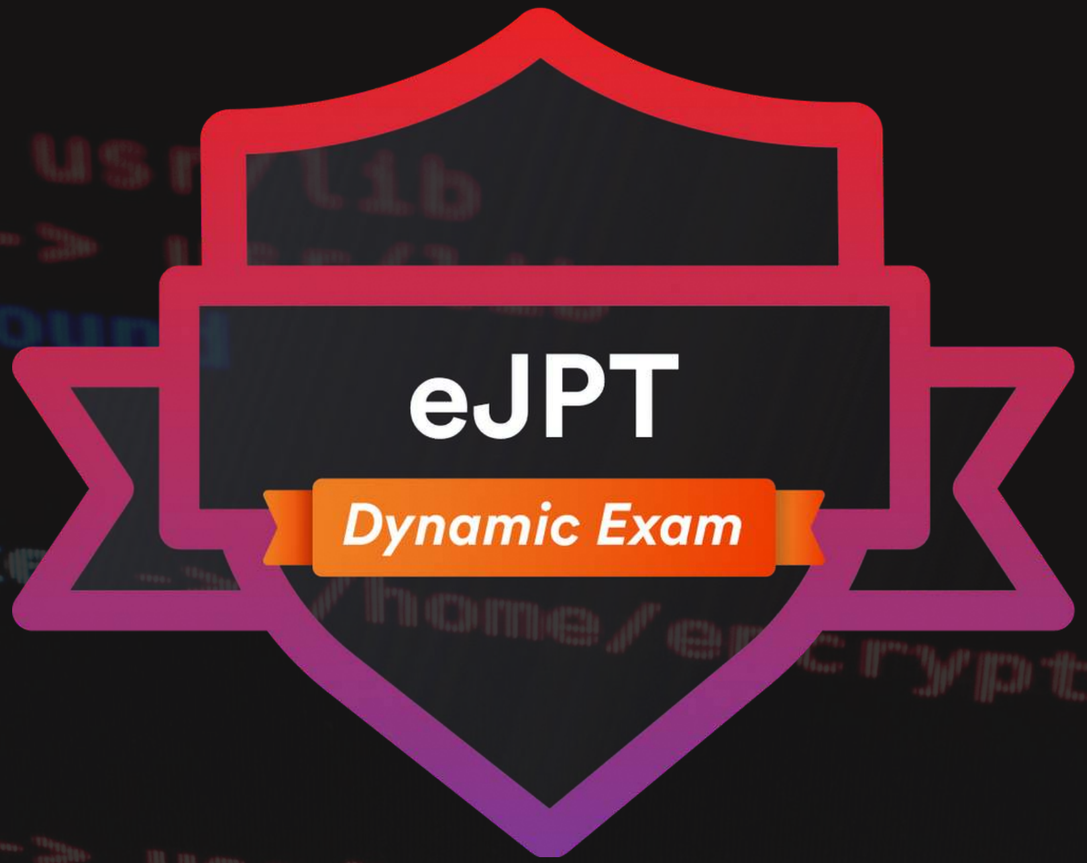


```
$ cat research.txt
```

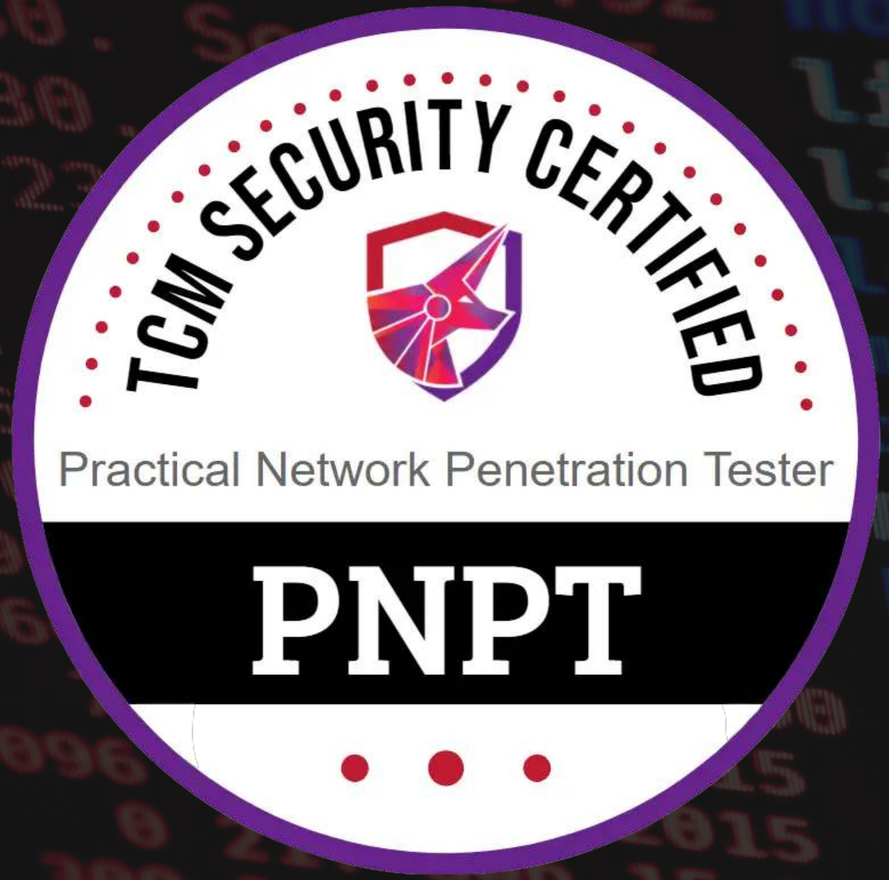
Research - szuka nowych podatności (zero day), analizuje malware, robi reverse engineering → (teoretycznie najwyższy próg wejścia)

CERTYFIKATY: NA START





CERTYFIKATY: PIERWSZA PRACA PENTESTERA





CERTYFIKAT: KRÓL JEST JEDEN



POLSKI RYNEK PRACY

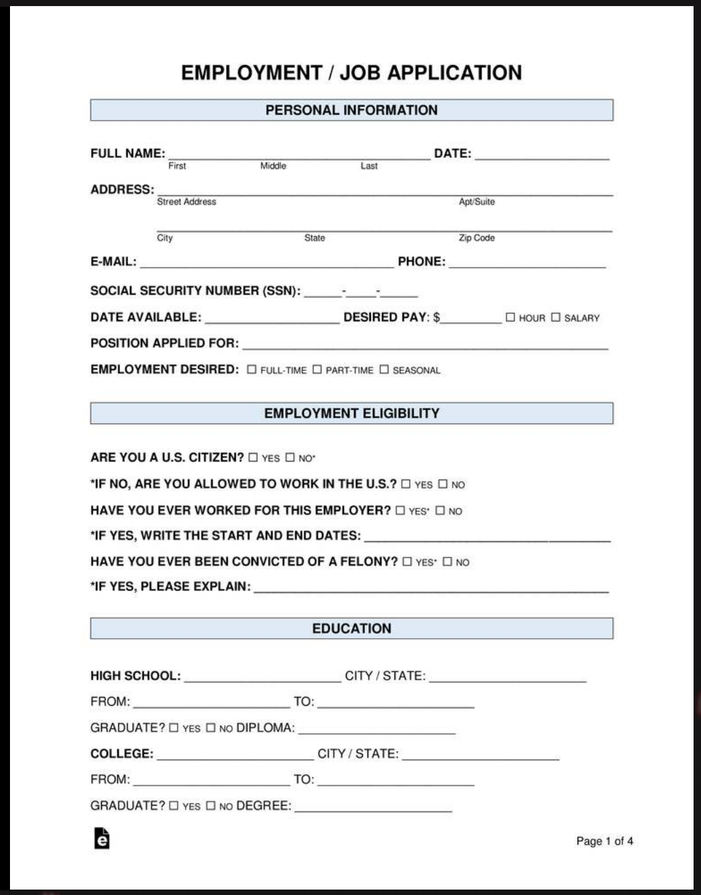
```
$ cat ai_pls_nie_zabierz_mi_pracy.txt
```

Sektor prywatny:

- Firmy pentesterskie i konsultingowe
- Dostawcy usług SOC i MSSP
- Banki i fintechy
- Firmy technologiczne i software house’y
- Startupy

Sektor publiczny:

- CERT Polska
- Wojska Obrony Cyberprzestrzeni
- ABW (Agencja Bezpieczeństwa Wewnętrznego)
- CBZC (Centralne Biuro Zwalczania Cyberprzestępczości)
- AW (Agencja Wywiadu)
- MSWiA (Ministerstwo Spraw Wewnętrznych i Administracji)
- Ministerstwo Cyfryzacji, KNF (Komisja Nadzoru Finansowego) i inne urzędy
- Spółki infrastruktury krytycznej: energetyka, transport, wodociągi



EMPLOYMENT / JOB APPLICATION

PERSONAL INFORMATION

FULL NAME: _____ DATE: _____
First Middle Last

ADDRESS: _____
Street Address Apt/Suite

_____
City State Zip Code

E-MAIL: _____ PHONE: _____

SOCIAL SECURITY NUMBER (SSN): ___-___-____

DATE AVAILABLE: _____ DESIRED PAY: $_____ ☐ HOUR ☐ SALARY

POSITION APPLIED FOR: _____

EMPLOYMENT DESIRED: ☐ FULL-TIME ☐ PART-TIME ☐ SEASONAL

EMPLOYMENT ELIGIBILITY

ARE YOU A U.S. CITIZEN? ☐ YES ☐ NO*

*IF NO, ARE YOU ALLOWED TO WORK IN THE U.S.? ☐ YES ☐ NO

HAVE YOU EVER WORKED FOR THIS EMPLOYER? ☐ YES* ☐ NO

*IF YES, WRITE THE START AND END DATES: _____

HAVE YOU EVER BEEN CONVICTED OF A FELONY? ☐ YES* ☐ NO

*IF YES, PLEASE EXPLAIN: _____

EDUCATION

HIGH SCHOOL: _____ CITY / STATE: _____

FROM: _____ TO: _____

GRADUATE? ☐ YES ☐ NO DIPLOMA: _____

COLLEGE: _____ CITY / STATE: _____

FROM: _____ TO: _____

GRADUATE? ☐ YES ☐ NO DEGREE: _____

Page 1 of 4

JAK WEJŚĆ DO BRANŻY?

```
$ cat ktos_pamieta_jeszcze_poradniki_na_bandicam_z_notatnikiem.txt
```

1. Praktyka → (Hack The Box, TryHackMe, CTF-y, White Hats Gym ;3)
2. Portfolio → (writeupy, projekty na GitHubie, blog, własna strona internetowa)
3. Certyfikaty → (CompTIA Security+, Offensive Security Certified Professional)
4. Ludzie → (koła naukowe, studia, konferencje: OMH, CONFidence)
5. Pierwsza praca → (staże, SOC L1, administracja, helpdesk, testowanie oprogramowania)

GDZIE USTAWIA CIĘ TEN KURS?

```
$ cat poradniki.txt
```

Semestr zimowy → (fundamenty: systemy, sieci, rekonesans, pierwsze ataki; dobra baza pod eJPT)

Semestr letni → (web, AD, chmura, raportowanie; solidny start do przygotowań do PNPT, CPTS czy OSCP)

Egzamin kursu → (zadania praktyczne + writeupy, czyli podobny format co certyfikaty branżowe)

MYŚL KOŃCOWA

```
$ cat mysl_koncowa.txt
Kurs Pentestera nie zrobi z
ciebie seniora, ale pokaże
ci czy w ogóle chcesz
siedzieć w IT lub
cyberbezpieczeństwie, a
jeżeli tak, to którą
ścieżką chcesz iść <3...
```



$ cat rozdzial_7.txt

Pierwsze kroki

KONTO NA WHITE HATS GYM

```
$ cat gym.txt
```

1. Wejdź na gym.wh.edu.pl (kod QR)
2. Załóż konto (najlepiej adres e-mail studencki)
3. **Jeśli studiujesz na PWr, załóż konto na studencki e-mail**

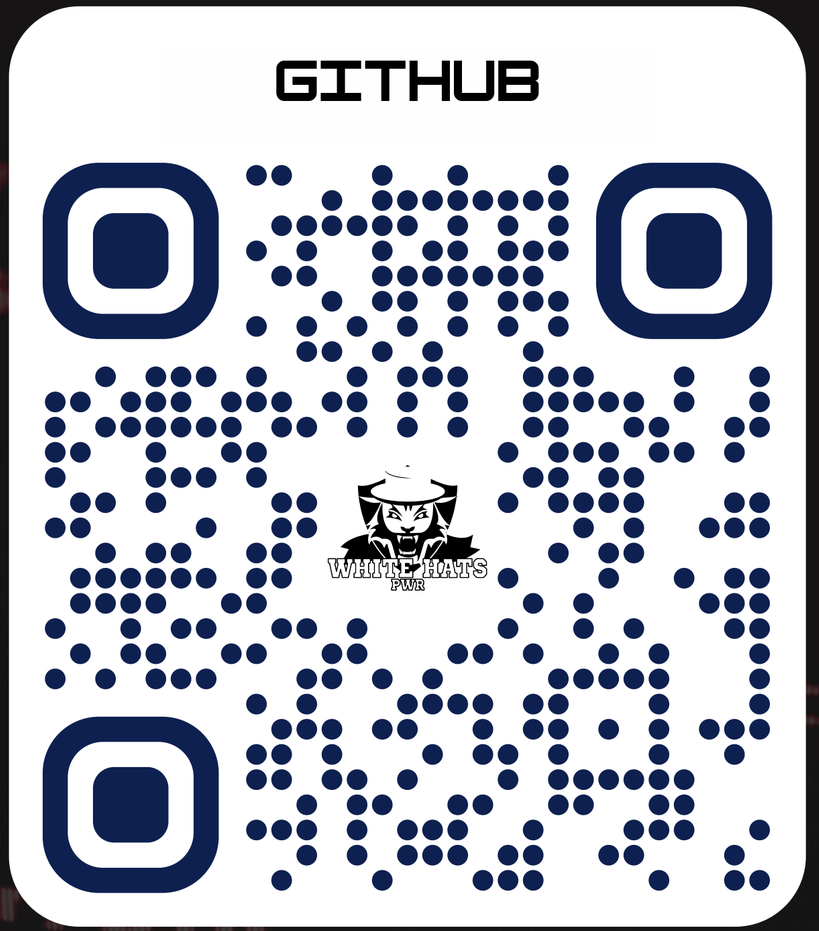
WHITE HATS GYM

KONTO NA GITHUBIE

```
$ cat github.txt
```

1. Załóż konto na github.com
2. Aktywuj status studenta

GITHUB



OD NASTĘPNYCH ZAJĘĆ (1/4)

```
$ cat laptop.txt
```

Od zajęć 2 (18.11) przynieście własny laptop, pracujcie w parze z osobą, która go ma lub po prostu oglądajcie co robimy my.



OD NASTĘPNYCH ZAJĘĆ (2/4)

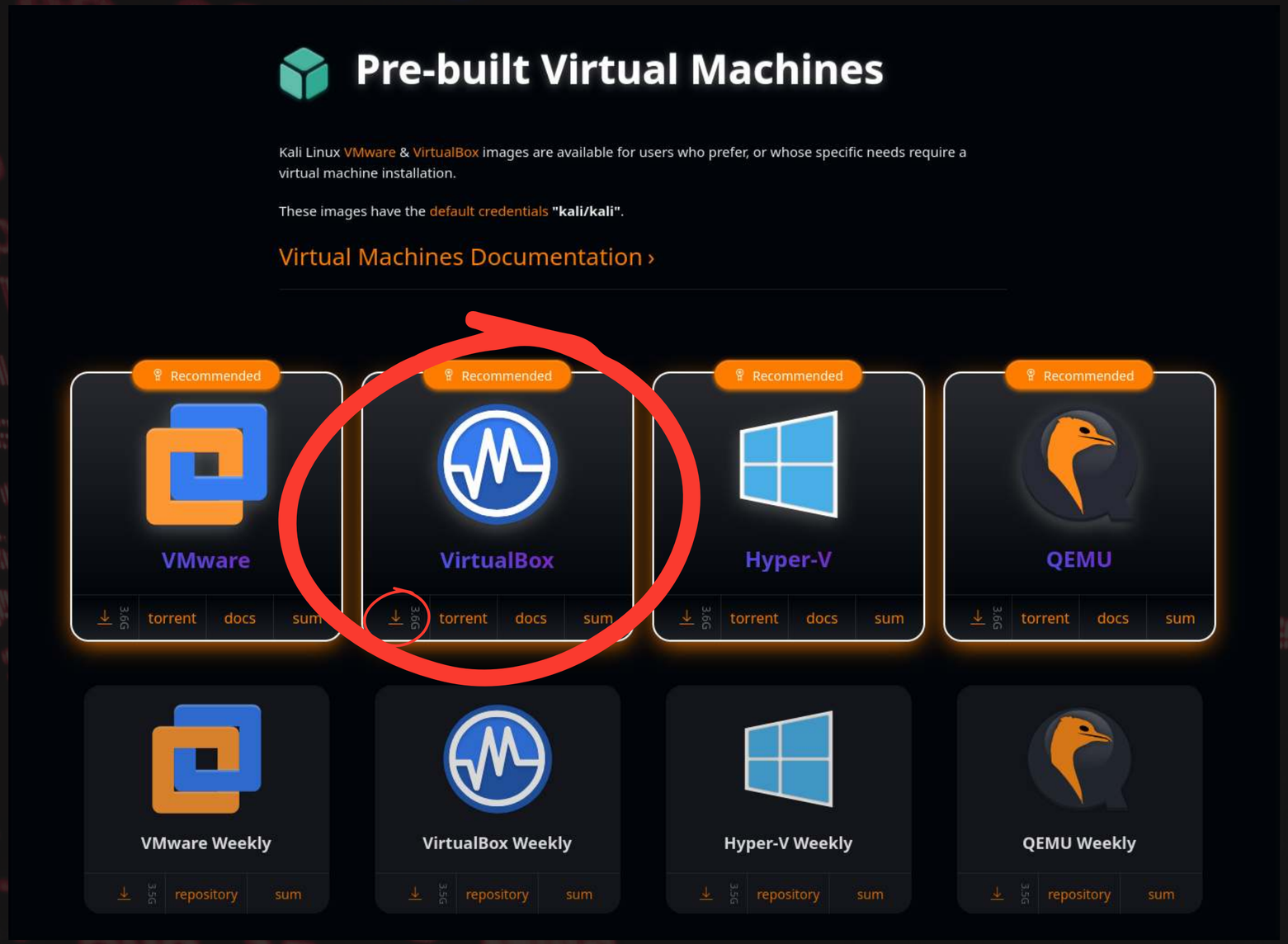
```
$ cat kali.txt
```

Przed zajęciami 2 pobierz obraz Kali Linux i zainstaluj Oracle VirtualBox.
Nie zostawiaj tego na zajęcia, bo już od 10 osób pobierających kilka GB naraz będzie problem.

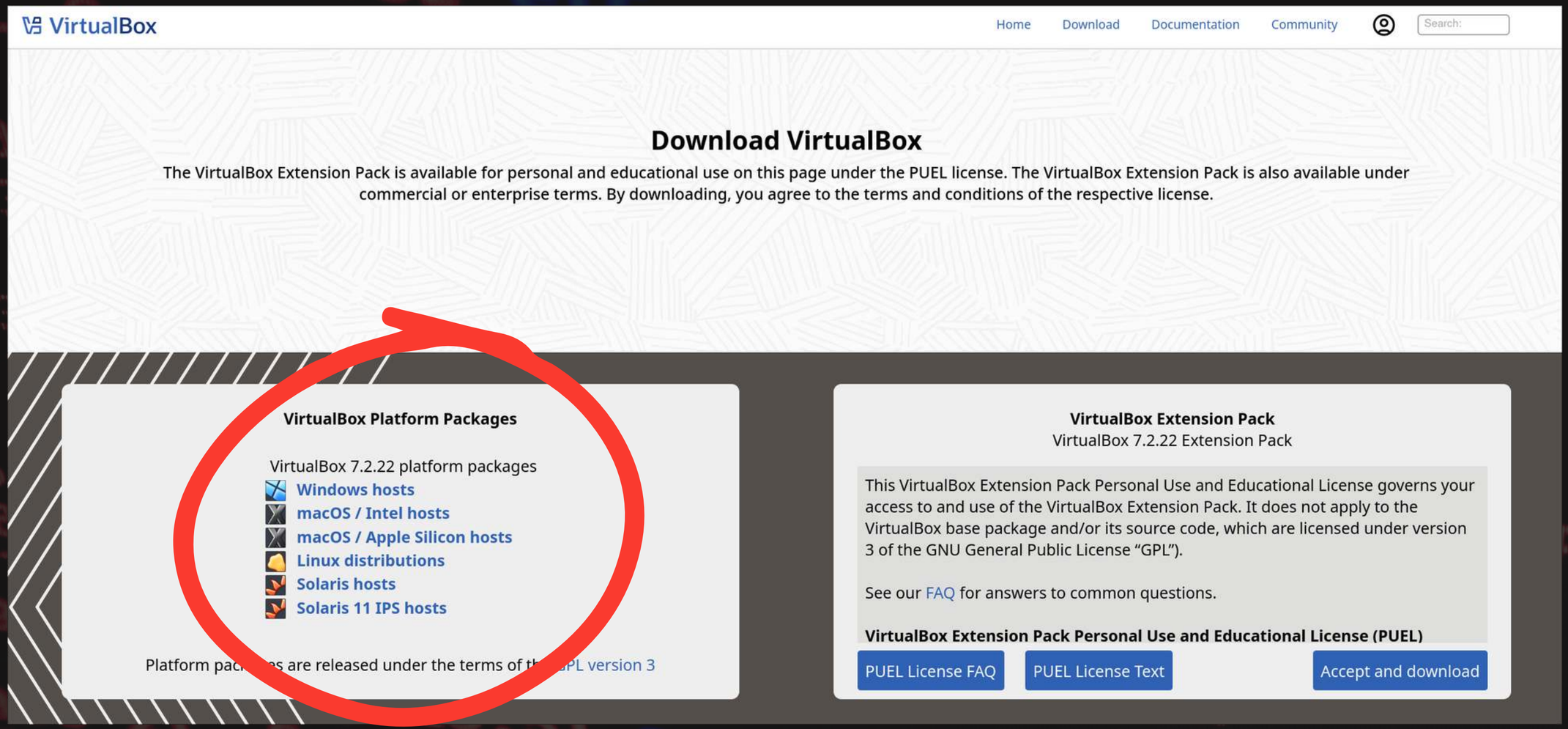
KALI LINUX

VIRTUALBOX

OD NASTĘPNYCH ZAJĘĆ (3/4)



OD NASTĘPNYCH ZAJĘĆ (4/4)





KURS
PENTESTERA

```
$ cat pytania.txt
```

Pytania?



KURS PENTESTERA

```
$ cat podziękowanie.txt
```

Dziękujemy za uwagę