#### 2121Funkcja skrótu SHA3 1

#### Łamanie skrótu MD5 metodą słownikową

#### Odszyfrowanie pliku szyfrem AES

#### Kod uwierzytelniający HMAC

#### Algorytm PBKDF2 z niestandardową liczbą iteracji

#### Identyfikacja algorytmu szyfrującego

#### Szyfrowany plik Zip

**Generowanie kluczy RSA**

<https://www.youtube.com/watch?v=18q1WOcyzWM>

#### Generowanie kluczy na krzywych eliptycznych

#### <https://www.youtube.com/watch?v=JpFwTpIubBo>

#### Szyfrowanie kluczem publicznym

#### <https://youtu.be/bR74jBpT2G0>

#### Certyfikat na podstawie zapytania

#### <https://youtu.be/CCimhx4Qe8o>

#### Generowanie klucza PGP

#### <https://youtu.be/77PLPCd250g>

#### Szyfrowanie za pomocą klucza PGP

#### <https://youtu.be/7pyJb6hy1eo>

#### Odszyfrowanie za pomocą klucza PGP

#### <https://youtu.be/0oxzn0b75KY>

#### Automatyczne wylogowywanie

#### <https://youtu.be/7gF8aXn0N5M>

#### ssh -l alice

#### ssh -l alice [host\_ip] "cat secret.txt"

#### Dostęp do plików należących do użytkownika root

#### <https://youtu.be/1DISmbatA2A>

#### cd archive

#### ./file

#### ../root/secret.txt

#### Podwyższenie uprawnień użytkownika

#### <https://youtu.be/PKY6ULiuXug>

#### find / -perm /6000 2>/dev/null

#### php7.4 /root/secret.txt

#### Inna powłoka niż Bash

#### <https://youtu.be/9vUhS8QU1hI>

#### cat /etc/passwd

#### cd /usr/bin/

#### cat bob

#### cd ~

#### ls -la (-> bob\_key.pem)

#### ssh -l bob localhost -i bob\_key.pem ( na zmniejszonym oknie terminala by zatrzymać działanie programu ‘more’)

#### klawisz ‘v’

#### : e secret.txt

#### Usunięty plik

#### <https://youtu.be/n6bWUYs8qMw>

#### ps -x ( lub ps -aux)

#### szukamy w /proc danego PIDa i lecimy do /fd

#### Analiza logów z konkretnego zakresu czasowego

#### <https://youtu.be/cz1N_5UT9PI>

#### cd /var/log

#### cat audit.log | grep -e "08:0[0-2]:[0-5][0-9]"

#### Analiza logów konkretnej usługi

#### <https://youtu.be/uHVRAqXXxAE>

#### cd /var/log

#### cat audit.log | grep -e "vsftpd\|ftp\|xinetd"

#### Analiza udanych prób logowania się na SSH

#### <https://youtu.be/EiLWkwlr-WI>

#### cd /var/log

#### cat audit.log | grep -e "sshd.\*invalid"

#### Analiza wielu nieudanych prób logowania się na SSH

#### <https://youtu.be/Wo9RXZpXjdQ>

#### cd /var/log

#### cat audit.log | grep -e “user\_”

#### Klasyczna podatność SQL Injection

#### Odczyt schematu bazy danych

#### Podatność SQL Injection typu Blind

#### Podatność SQL Injection typu Time Based

#### Podgląd nagłówków odpowiedzi

#### <https://youtu.be/-6K7SqADd2s>

#### Manipulowanie parametrami HTTP

#### <https://youtu.be/0VoR903FNqk>

#### ?id=6

#### Automatyczny skan aplikacji

#### <https://youtu.be/R5cjc2nEgU4>

#### odpalamy OWASP -> Automatyczne skanowanie -> wpsujemy url, zaznaczamy Sidera ->

#### wyszukujemy url ze sćieżką /robots.txt i sprawdzamy response i odczytujemy kod.

#### Modyfikacja żądań HTTP

#### <https://youtu.be/0vP1NliWQhE>

#### firefoxem (opera nie działała) wysyłamy zmieniony nagłówek przez inspectora i szukamy flagi z odpowiedzią

#### Powtarzanie żądania HTTP

#### <https://youtu.be/S_gVNlVh1QM>

#### (chyba z firefoxa wystarczy od razu jeden request zrobić) a jak nie to odpalić burpa -> załączyć proxy w firefoxie -> wysłać żadanie -> dodać w burpie przechwycone rządanie do repeatera i tam jeszcze raz wysłać żądanie

#### Wyszukiwanie właściwej wartości parametru metodą siłową

#### <https://youtu.be/yW_ZCp0OpIE>

#### odpalamy OWASP -> manual explorer -> wklejamy url, odhaczamy ‘Enbable HUD’ i odpalamy -> szukamy żądania z najmniejszym rozmiarem response body (1615b) i tam powinna być odpowiedź z secret