

Adam Behan

Pojęcie i prawnokarna ochrona programu komputerowego

1. Wprowadzenie

W poniższym opracowaniu analizowana będzie problematyka wykładni pojęcia programu komputerowego na gruncie Kodeksu karnego¹ oraz wynikające z rozwoju technologii informatycznych problemy interpretacyjne, które z tym zagadnieniem się łączą. Przedstawiony zostanie rys historyczny ewolucji pojęcia, ze wskazaniem na akty prawa międzynarodowego, które stały u podstaw obecnej regulacji i wykładni tego pojęcia na gruncie prawa wspólnotowego. W dalszej części tekstu krytycznej analizie poddane zostaną pojawiające się w literaturze wybrane koncepcje tego, czym jest program komputerowy, w tym rozumienie tego pojęcia na gruncie Kodeksu karnego, a także rozważone zostanie zagadnienie rozumienia stron internetowych jako egzemplifikacji pojęcia programu komputerowego i wynikające z tego konsekwencje.

Pomimo ponad piętnastu lat obowiązywania Kodeksu karnego trudno znaleźć w orzecznictwie czy piśmiennictwie karnistycznym spójną i jednoznaczną definicję pojęcia programu komputerowego, użytego przez ustawodawcę w art. 278 § 2 k.k. Obecnie niemal każde urządzenie elektroniczne posiada bądź wbudowane oprogramowanie sterujące (*firmware*), bądź – tak jak w przypadku choćby smartfonów – zainstalowane systemy operacyjne, których liczba linii kodu źródłowego przekracza milion². Ze względu na

¹ Ustawa z dn. 6 VI 1997 r. – Kodeks karny, Dz.U. 2017, poz. 2204 tekst jedn. ze zm., dalej: „Kodeks karny”, „k.k.”

² Zob. M. Brian, *Google's Andy Rubin: There are over 1 million lines of code in Android*, < <http://thenextweb.com/google/2011/10/19/googles-andy-rubin-there-are-over-1-million-lines-of-code-in-android> >.

wszechogarniającą i wciąż postępującą informatyzację programy komputerowe sterują obecnie każdym, nawet najdrobniejszym aspektem naszego życia: począwszy od programu wbudowanego w sygnalizację świetlną, służącego do sterowania światłami na skrzyżowaniu, przez program sterujący budzikiem w naszym telefonie komórkowym, aż po program odpowiedzialny za przetworzenie i analizę naszego zapytania w przeglądarce internetowej, który walnie przyczynił się do rozwoju firmy wartej obecnie przeszło sto siedemdziesiąt miliardów dolarów³.

O ile bowiem w praktyce nie nastęcza problemów wskazanie oczywistych desygnatów pojęcia programu komputerowego, takich jak np. systemy operacyjne, pakiety biurowe, pakiety do obróbki grafiki czy gry komputerowe⁴, o tyle próżno szukać w orzecznictwie i piśmiennictwie egzemplifikacji konkretnych rozstrzygnięć odnośnie do desygnatów tego pojęcia w przypadku np. fragmentów kodu źródłowego, kompletnych klas, fragmentów skryptów czy zdekompilowanych kodów wynikowych, co obecnie może rodzić ogromne problemy przy wykładni znamion ustawowych. Warto mieć na uwadze, że dzisiejsze serwisy internetowe właściwie nie zawierają już wyłącznie języka HTML, jak miało to miejsce jeszcze dziesięć lat temu. Pisane są w językach skryptowych (np. PHP, Python, Perl, Ruby czy JavaScript), liczą tysiące linii kodu, zawierają niezwykle rozbudowaną architekturę, oferując rozliczne możliwości współpracy z zewnętrznymi aplikacjami (np. CRM, ERP), a często za pośrednictwem bibliotek programistycznych, jak np. jQuery⁵, zawierają różnorakie dodatkowe skrypty, również niejednokrotnie liczące setki linii kodu, będące w pełni funkcjonalnymi programami odpowiedzialnymi za przetwarzanie danych czy zachowań użytkownika, język HTML wykorzystując niemal wyłącznie do prezentacji wyników przetwarzania programów komputerowych napisanych w innych językach.

³ Zob. S. DeCarlo, *The world's 25 most valuable companies: Apple is now on top*, < <http://www.forbes.com/sites/scottdecarlo/2011/08/11/the-worlds-25-most-valuable-companies-apple-is-now-on-top> >.

⁴ Szerzej na temat gier komputerowych jako przedmiotów prawa autorskiego: I. Matusiak, *Gra komputerowa jako przedmiot prawa autorskiego*, Warszawa 2013. Odmienne: J. Lachowski, w: *Kodeks karny. Komentarz*, red. V. Konarska-Wrzosek, Warszawa 2016, komentarz do art. 278 k.k., teza 4.

⁵ < <http://jquery.com> >.

2. Rys historyczny

Pierwszym ustawowym uregulowaniem dotyczącym programów komputerowych w polskim porządku prawnym była regulacja ustawy o prawie autorskim i prawach pokrewnych z dn. 4 lutego 1994 r.⁶ Akt ten zawiera rozdz. VII: „Przepisy szczególne dotyczące programów komputerowych”, jednakże bez definicji programu komputerowego. Warto nadmienić, że jakkolwiek ustawa o prawie autorskim z 1952 r.⁷ nie zawierała w swej treści postanowień odnośnie do programów komputerowych, to doktryna wyraźnie opowiadała się za włączeniem w dyspozycję art. 1 tej ustawy programów komputerowych, co znajdowało potwierdzenie również w orzecznictwie. W wyroku Sądu Apelacyjnego w Gdańsku z dn. 29 stycznia 1993 r. stwierdzono, że:

Oprogramowanie komputerowe może być traktowane jako utwór o charakterze naukowym lub literackim (art. 1 § 1 prawa autorskiego z 1952 r.), jeżeli posiada ono cechę oryginalności twórczej, spełnia przewidziany przez ustawę wymóg odpowiedniego ustalenia (*verba legis*: „ustalony w jakiegokolwiek postaci”) i zawiera elementy indywidualizujące twórcę programu⁸.

W wyroku z 16 września 2002 r. Naczelny Sąd Administracyjny stwierdził, że:

Wykładnia art. 1 § 1 i 2 ustawy z 1952 r. o prawie autorskim, jak i art. 1 ust. 1 i 2 ustawy z 1994 r. o prawie autorskim i prawach pokrewnych prowadzi do wniosku, że brak jest normatywnych przesłanek do uznania, że ustawodawca od dnia wejścia w życie ostatniej z przywołanych ustaw wyłączył należności licencyjne z tytułu korzystania z programów komputerowych spod pojęcia należności z tytułu korzystania z utworów chronionych przez prawo autorskie. Innymi słowy prawa w zakresie oprogramowania stanowią rodzaj własności intelektualnej, które były i są objęte ochroną prawa autorskiego⁹.

Wprawdzie w obecnie obowiązującej ustawie o prawie autorskim i prawach pokrewnych znalazł się rozdział dotyczący programów komputerowych, to nie zawiera on definicji programu komputerowego. Jest to zgodne

⁶ Ustawa z dn. 4 II 1994 r. o prawie autorskim i prawach pokrewnych, Dz.U. 1994, nr 24, poz. 83, dalej: „ustawa o prawie autorskim i prawach pokrewnych”, „pr. aut.”

⁷ Ustawa z dn. 10 VII 1952 r. o prawie autorskim, Dz.U. 1952, nr 34, poz. 234.

⁸ Wyrok SA w Gdańsku z 29 I 1993 r., I AGCr 369/92, OSA 1993, z. 6, poz. 43.

⁹ Wyrok NSA oz. w Katowicach z 16 IX 2002 r., I SA/Ka 1250/01, LEX nr 575152.

z dominującym europejskim trendem¹⁰ nieumieszczenia definicji legalnej pojęcia programu komputerowego w ustawodawstwie, rozpoczętym już w pierwszej propozycji Dyrektywy Wspólnoty Europejskiej dotyczącej prawnoautorskiej ochrony programów komputerowych¹¹.

Stanowisko wyrażone w tejże propozycji poddane zostało szerokim konsultacjom¹², podczas których zgłoszonych zostało wiele uwag odnośnie do charakteru ochrony programów komputerowych i szczegółowych zagadnień ich dotyczących¹³. Nie podnoszono natomiast zastrzeżeń co do rozumienia samego pojęcia programu, które w niemalże niezmienionej formie zostało transponowane do pierwszej Dyrektywy 91/250/EWG w sprawie ochrony prawnej programów komputerowych¹⁴. Zgodnie z jej brzmieniem na potrzeby Dyrektywy pojęcie programu obejmowało:

[...] programy w jakiegokolwiek formie, włącznie z programami zintegrowanymi ze sprzętem komputerowym; pojęcie to obejmuje również przygotowawcze prace projektowe prowadzące do rozwoju programu komputerowego z zastrzeżeniem, że charakter prac przygotowawczych jest taki, że program komputerowy może korzystać z nich na późniejszym etapie.

¹⁰ Odmienne np. rosyjska ustawa o ochronie programów komputerowych i baz danych, która posiada definicję programu komputerowego – szerzej M. Brożyna, *Ochrona programów komputerowych w świetle ustawodawstwa Federacji Rosyjskiej*, „Rzecznik Patentowy” 2001, nr 1, s. 95.

¹¹ Proposal for a Council Directive on the legal protection of computer programs. COM (88) 816 final, 17 March 1989, < [http://aei.pitt.edu/13138/1/COM_\(88\)_816_final.pdf](http://aei.pitt.edu/13138/1/COM_(88)_816_final.pdf) >. W dokumencie już we wstępie zasygnalizowano rosnące znaczenie programów komputerowych, a także podano wyjaśnienie, że na potrzeby dokumentu program rozumiany jest jako zestaw instrukcji, których celem jest wykonanie przez komputer zadanych funkcji („set of instructions the purpose of which is to cause an information processing device, a computer, to perform its functions”), stwierdzając jednocześnie w art. 1 cz. 2, że nie jest właściwym włączać w treść aktu legalną definicję programu z racji ryzyka jej szybkiej dezaktualizacji z powodu szybkiego postępu technicznego („It has been recommended by experts in the field that any definition in a directive of what constitutes a program would of necessity become obsolete as future technology changes the nature of programs as they are known today [...] Given the present state of the art, the word ‘program’ should be taken to encompass the expression in any form, language, notation or code of a set of instructions, the purpose of which is to cause a computer to execute a particular task or function”).

¹² Szczegółowa lista: < <https://www.civil.law.cam.ac.uk/projectseuropean-travaux/copyright-computer-programs> >.

¹³ Między innymi przesądzono wybór ochrony prawnoautorskiej programów komputerowych, rezygnując z modelu ochrony patentowej.

¹⁴ Dyrektywa Rady z dn. 14 V 1991 r. w sprawie ochrony prawnej programów komputerowych, 91/250/EWG, Dz.U. UE L z dn. 17 V 1991 r., dalej: „Dyrektywa 91/250/EWG”.

Co również bardzo istotne, w art. 1 przesądzono, że:

Zgodnie z niniejszą dyrektywą ochronie podlega każda forma wyrażenia programu komputerowego. Koncepcje i zasady, na których opierają się wszystkie elementy programu komputerowego, włącznie z tymi, na których opierają się ich interfejsy, nie podlegają ochronie prawa autorskiego na podstawie niniejszej dyrektywy.

Postanowienia dotyczące rozumienia programu komputerowego po raz kolejny powtórzono w obowiązującej obecnie Dyrektywie 2009/24/WE¹⁵. Na to podobieństwo zwraca uwagę Zbigniew Okoń:

Zbieżność definicji legalnych programu komputerowego nie jest przypadkowa. Począwszy od historycznie najstarszej regulacji amerykańskiej (zob. § 101 Copyright Act w brzmieniu nadanym nowelizacją z 12 grudnia 1980 r., Law No. 96–517), są one bowiem wzorowane na rozwiązaniach zawartych w tzw. przepisach wzorcowych o ochronie programów komputerowych, opracowanych w latach 70. w ramach WIPO¹⁶.

A według WIPO¹⁷ program komputerowy to taki:

[...] zestaw instrukcji, który umieszczony na nośniku, z którego maszyna posiadająca zdolność przetwarzania informacji może pobierać dane, powoduje, że osiąga ona zdolność do wykonywania lub wykonuje określone zadania¹⁸.

Brzmienie art. 74–76 pr. aut. jest w zasadzie tłumaczeniem postanowień Dyrektywy 91/250/EWG. Fakt implementacji postanowień wspólnotowych wynikał z prozaicznej konieczności ratyfikacji przez sejm Traktatu o stosunkach handlowych i gospodarczych między Rzeczpospolitą Polską a Stanami Zjednoczonymi Ameryki¹⁹ i z faktu, że postanowienia Traktatu były tożsame

¹⁵ Dyrektywa Parlamentu Europejskiego i Rady z dn. 23 IV 2009 r. w sprawie ochrony prawnej programów komputerowych, 2009/24/WE, Dz.U. UE L z dn. 5 V 2009 r., dalej: „Dyrektywa 2009/24/WE”.

¹⁶ Z. Okoń, w: *Prawo autorskie i prawa pokrewne. Komentarz*, red. D. Flisak, Warszawa 2015, komentarz do art. 74 pr. aut., teza 9.

¹⁷ Światowa Organizacja Własności Intelektualnej, ang. *World Intellectual Property Organization*.

¹⁸ „[...] a set of instructions capable, when incorporated in a machine-readable medium, of causing a machine having information-processing capabilities to indicate, perform or achieve a particular result” – WIPO Model Provisions on the Protection of Computer Software (1978).

¹⁹ Sporządzony w Waszyngtonie dn. 21 III 1990 r., Dz.U. 1994, nr 97, poz. 467.

co do zakresu z rozwiązaniami przyjętymi w Dyrektywie 91/250/EWG²⁰, a ich przyjęcie czyniło zadość żądaniom strony amerykańskiej.

Zgodnie z art. 74 ust. 1 *in fine* pr. aut.: „programy komputerowe podlegają ochronie jak utwory literackie, o ile przepisy niniejszego rozdziału nie stanowią inaczej”, co sprawia, że w odniesieniu do programów komputerowych zastosowanie mieć będą postanowienia konwencji berneńskiej²¹, ratyfikowanej przez Polskę w 1935 r.²² (do której odwołuje się również w art. 1 Dyrektywa 2009/24/WE oraz odwoływała się Dyrektywa 91/250/EWG). Tworzy to spójny obszar prawny, na gruncie którego należy definiować pojęcie programu komputerowego.

3. Program komputerowy – aspekty techniczne

Konieczne jest kilka słów wyjaśnienia odnośnie do tych zagadnień natury informatycznej, których znajomość jest niezbędna dla zrozumienia poniższych rozważań. Pojęcie programu komputerowego przywodzi na myśl system operacyjny Windows czy Linux, pakiet Office czy Adobe Reader. Istnieją jednak obiekty, które na pierwszy rzut oka programami nie są, a dopiero bliższa analiza informatyczna może dać odpowiedź na pytanie, czy mamy do czynienia z programem komputerowym, czy też nie. Jest to jednakowoż wysoko specjalistyczne pojęcie z zakresu informatyki i dla pełnego zrozumienia niuansów stojących za rozważaniami poczynionymi w dalszej części niniejszego opracowania należy rozpocząć od wyjaśnienia kluczowych pojęć *stricte* technicznych, a także choćby pobieżnie przedstawić proces tworzenia programów komputerowych.

²⁰ W trakcie 69 posiedzenia Sejmu w dn. 26 VII 1991 r., w czasie dyskusji nad projektami uchwały i ustawy o ratyfikacji Traktatu o stosunkach handlowych i gospodarczych między Rzeczpospolitą Polską a Stanami Zjednoczonymi Ameryki, poseł M. Świącicki stwierdził: „ostatecznie Dyrektywa EWG przyjęta 14 maja tego roku dokładnie przyjmuje te same rozwiązania, które są przyjęte w traktacie. Również sprawa wyjątków od tzw. dekompilacji programów komputerowych, co poruszano w naszej uchwale, strona amerykańska dała nam jednoznaczną odpowiedź w trakcie prac podkomisji, że jeśli przyjmujemy te wyjątki w takim zakresie, w jakim przyjęła je EWG, to nie będzie to traktowane jako naruszenie traktatu”. Źródło: < [http://orka2.sejm.gov.pl/StenogramyX.nsf/0/A13C3C4FBCB2CD20C1257D20002CC73C/\\$file/069_000007163.pdf](http://orka2.sejm.gov.pl/StenogramyX.nsf/0/A13C3C4FBCB2CD20C1257D20002CC73C/$file/069_000007163.pdf) >.

²¹ Konwencja berneńska o ochronie dzieł literackich i artystycznych z dn. 9 IX 1886 r., Dz.U. 1935, nr 84, poz. 515.

²² Obecnie Polska związana jest „aktem paryskim”, przyjętym w dn. 24 VII 1971 r., Dz.U. 1990, nr 82, poz. 474.

„Computers are incredibly fast, accurate, and stupid [...]”²³ – nie ma wiele przesady w tym stwierdzeniu. Komputery, a w zasadzie procesory, to urządzenia, które w gruncie rzeczy nie są nadzwyczajnie rozbudowane, jeśli chodzi o liczbę dostępnych operacji, które umieją wykonać. Procesory, które są w naszych laptopach, komórkach czy tabletach, to w największym uproszczeniu połączenie kalkulatora z zegarkiem. Liczba operacji arytmetycznych, które procesor potrafi wykonać po „podaniu” mu danych, też nie jest ogromna – procesor potrafi między innymi: dodać, odjąć, porównać dwie liczby, zmienić znak liczby, mnożyć i dzielić²⁴. Oczywiście, wykorzystując komputer, możemy tworzyć filmy, muzykę czy przepiękne grafiki, ale bynajmniej nie jest tak, że program prześle do procesora instrukcję: „zmień kolory w zdjęciu na czarno-białe”. W największym uproszczeniu program narzędziowy „do zdjęć” w połączeniu z systemem operacyjnym (również będącym programem), wykonując zadaną przez nas funkcję, zmieni ją w zrozumiałe dla procesora instrukcje, rozbijając ją na najdrobniejsze elementy („dodaj komórkę A do komórki B”, „przesuń zawartość pamięci z C do D”, „zmień znak liczby” itd.). Wszystkie operacje dokonywane przez procesor mógłby wykonać każdy z nas, z jedną różnicą – procesor wykonuje takich operacji kilka miliardów na sekundę²⁵.

Co zatem oznacza sformułowanie: „język zrozumiały dla komputera”, występujące we wszystkich definicjach? Wyobraźmy sobie program, który wyświetla nam na ekranie liczby całkowite od 1 do nieskończoności. Weźmy za przykład język Java.

```
public class Count{
    public static void main(String[] args){
        for(long i = 1; ;i++) System.out.println(i);
    }
}
```

²³ Cytat, przypisywany A. Einsteinowi, tak naprawdę pojawił się w „Advances in Instrumentation” 1969, t. 24/4, s. 691. Nie znamy autora tego stwierdzenia, które w całości brzmi: „Computers are incredibly fast, accurate, and stupid. On the other hand, a well-trained operator as compared with a computer is incredibly slow, inaccurate, and brilliant” („Komputery są niezwykle szybkie, dokładne i głupie, podczas gdy dobrze wyszkolony ich operator jest niezwykle wolny, niedokładny i genialny”).

²⁴ Pełny zestaw instrukcji procesorów firmy Intel znaleźć można na stronie: < <https://software.intel.com/en-us/articles/intel-sdm> >.

²⁵ Czym innym jest taktowanie procesora – np. 3,0 GHz, oznaczające liczbę taktów, w których procesor może wykonać pojedyncze rozkazy, np. przenieść pamięć z komórki A do B – a czym innym liczba wykonanych operacji zmiennoprzecinkowych, którą reprezentuje się za pomocą MIPS („millions of instructions per second”).

Czy taki program byłby „rozumiały” dla komputera? I tak, i nie. Programu napisanego w takiej formie procesor nie zrozumie, a podanie do pamięci procesora takiego kodu skutkowałoby błędem. Jakkolwiek gdy do „tłumaczenia” wykorzystamy inny program – zwany kompilatorem²⁶ – zamieni on zrozumiałe dla człowieka polecenia kodu napisanego w języku Java:

```
for(long i = 1; ;i++) System.out.println(i);
(przechodź pętlę [począwszy od 1, dodaj 1] Wyświetl
wynik - wróć na początek pętli)
```

– na polecenia zrozumiałe dla procesora. Skompiluje ten program, zamieniając go na zestaw instrukcji zrozumiałych dla procesora. Następnie (w największym uproszczeniu) procesor ten wykona zadaną operację – zwróci wynik, który zostanie zinterpretowany przez program (system operacyjny), przesłany do pamięci karty graficznej, która przetworzy sygnał i wyśle dane do monitora, który wyświetli nam na ekranie: „1 2 3 4 5 [...]”.

Jedynym „językiem”, który komputer (procesor) jest w stanie zrozumieć, jest bowiem kod maszynowy, który nie jest nawet językiem programowania. Są to instrukcje (polecenia), które producent procesora zaimplementował w czasie jego produkcji. Procesor zna tylko te instrukcje i tylko je rozumie. Zadaniem systemu operacyjnego jest przetłumaczyć procesorowi nasze polecenie, a następnie wynik zwrócony przez procesor przetłumaczyć na zrozumiałą dla nas formę.

Najbliżej języka maszynowego znajduje się język Assembler, będący mnemonicem (instrukcjom procesora odpowiadają zrozumiałe dla człowieka symbole).

```
MOV EAX, [EBX];
```

Co oznacza: „Przenieś 4 bajty pamięci z komórki o adresie EBX do komórki o adresie EAX”. Napisanie jednak czegokolwiek w Assemblerze jest zadaniem katorżniczym – program w nim napisany będzie ekstremalnie szybki, ale każda alokacja pamięci i każda najdrobniejsza czynność musiałaby zostać napisana ręcznie – a poza skrajną nieczytelnością program miałby setki tysięcy linii.

Z tego właśnie powodu zaczęły powstawać kolejne języki programowania²⁷, które dodając instrukcje, stawały się coraz bardziej przyjazne progra-

²⁶ Pierwszy kompilator powstał w 1952 r. jako odpowiedź na pojawienie się pierwszych języków maszynowych.

²⁷ Obecnie na świecie używanych jest ponad 750 języków programowania. Za: < https://en.wikipedia.org/wiki/List_of_programming_languages >.

mistom²⁸. Aby jednak „zaprząć” procesor do pracy, należało ów kod skompilować (za pomocą specjalnego programu – kompilatora – pisanego przez twórcę języka programowania). Innymi słowy, należałoby zamienić polecenia i funkcje oferowane przez dany język programowania na instrukcje zrozumiałe dla procesora²⁹. Oczywiście programista nie ma co do zasady wpływu na to, jak kompilator dojdzie do kodu maszynowego – będzie to efektem działania algorytmu samego kompilatora. Konsekwencją takiego stanu jest to, że dwa różne programy, mające różny – ale np. różniący się niewiele – kod źródłowy, podczas procesu kompilacji doprowadzone mogą zostać do takiego samego kodu maszynowego³⁰. Korzystając też z innej wersji kompilatora³¹, z tego samego kodu źródłowego otrzymamy inny kod wynikowy i dwa programy dla użytkownika wyglądające tak samo, ale np. różniące się wydajnością, zapotrzebowaniem na pamięć czy zużyciem procesora. Nie jest, wbrew twierdzeniom Ł. Zimończyka, tak, że „program zapisany w kodzie źródłowym, jak i kodzie wynikowym ma identyczną treść”³². To oczywiście są te same programy, jakkolwiek proces kompilacji w zależności od kompilatora, wersji systemu, ustawień kompilacji czy języka programowania może dać różne kody wynikowe³³. Sztandarowym przykładem będzie również dodanie jedną linijką biblioteki programistycznej (o tym szerzej poniżej), której kod kompilator skompiluje z naszym programem, dodając do kodu źródłowego liczącego dziesięć linijek kod biblioteki, która liczy linijek np. dziesięć tysięcy.

Aby dokładnie zrozumieć zakres karnoprawnej ochrony programów komputerowych, należałoby głębiej poruszyć także kilka zagadnień dotyczących sposobów tworzenia programów. Pierwszym są biblioteki programistyczne.

²⁸ I tak, dodając dwie wartości, nie trzeba już było np. sięgać do pamięci, sprawdzać komórek, dodawać ich sumy, zapisywać w innej komórce czy czytać – język programowania oferował np. polecenie „sum”, które robiło to samo.

²⁹ Napisane przez programistę „dodaj” – przez kompilator było zamieniane na kod maszynowy, który kolejno realizował sięgnięcie do pamięci, sprawdzenie komórek *etc.*

³⁰ Najbardziej bowiem skomplikowane funkcje matematyczne rozbite zostaną na podstawowe działania: „dodaj”, „odejmij”, „pomnóż”, „zmień znak” itd.

³¹ W związku z tym, że kompilatory są normalnymi programami, ich twórcy oczywiście cały czas je poprawiają, „wypuszczając” kolejne wersje kompilatorów, a same języki programowania mają bardzo wiele różnych kompilatorów, z których może korzystać programista. Dla języka C znajdziemy ponad 60 kompilatorów, różniących się szybkością, jakością, a nawet tym, czy są one darmowe, czy płatne.

³² Ł. Zimończyk, *Ustalenie programu komputerowego*, „Kwartalnik Krajowej Szkoły Sądownictwa i Prokuratury” 2017, z. 1 (5), s. 35.

³³ Kompilator może np. uprościć implementację, usuwając z wejścia nieistotne elementy, np. białe spacje i komentarze, czy dodać tzw. timestampy.

Biblioteka jest to zestaw programów danych lub zmiennych, które są zwyczajowo implementowane w język programowania. Za wzór może nam służyć język programowania Java, w którym chcemy napisać program, który będzie się łączył z bazą danych i pobierał dane w niej zawarte. Napisanie samodzielnie kodu odpowiadającego połączeniu z bazą danych firmy Oracle, obsługa wyjątków i stworzenie warstwy bezpieczeństwa to wiele dziesiątek godzin pracy programisty, którą wykonywać by musiało każdego dnia tysiące programistów, chcąc połączyć swój program z bazą danych. Alternatywnie można w jednej linii kodu implementować (dodać) do naszego programu bibliotekę `java.sql`, która ma już gotowe skompilowane funkcje do połączenia z zadaną bazą, przygotowane przez autora języka bądź społeczność internetową.

Bibliotek, które wykonują rozmaite funkcje, dla najpopularniejszych języków programowania są tysiące, i tak jak sam język – są one bezpłatne (oczywiście znaleźć można również biblioteki płatne – zazwyczaj bardzo specjalistyczne). Wykorzystanie bibliotek może więc dać naszemu programowi ogromne możliwości użytkowe (np. stworzyć rozbudowany kalkulator, który będzie wykonywał skomplikowane obliczenia), nie nadając mu przy tym oryginalności, koniecznej dla przydania mu przymiotu utworu, jeśli nasza rola będzie ograniczać się do wpisania w program pola dla podania liczby i prezentacji wyników, podczas gdy wszystkie operacje matematyczne będzie wykonywał program zawarty w bibliotece, a tylko implementowany w nasz kod. Rozpatrując zatem, czy program jest utworem w rozumieniu prawa autorskiego, a co za tym idzie – czy objęty jest prawnokarną ochroną, pamiętać należy, aby dokonana została analiza pod kątem roli, jaką w programie odgrywają standardowe i ogólnodostępne biblioteki.

Analogiczny przykład stanowi dostępność tysięcy skryptów (programów lub fragmentów programów) udostępnionych przez społeczność internetową, które można wykorzystać we własnym programie, wklejając je w swój kod. Zdecydowana większość z tych skryptów udostępniana jest nieodpłatnie i bez żadnych ograniczeń dla programistów do wykorzystania ich w ich własnych programach.

4. Program komputerowy a utwór

Może się okazać, że jeśli program w całości zbudowany został z fragmentów udostępnionych bezpłatnie kodów czy bibliotek, wkład autora łączącego te fragmenty nie będzie miał charakteru twórczego o indywidualnym

charakterze³⁴. Programy komputerowe – jak wcześniej wskazano, zgodnie z postanowieniami art. 74 pr. aut. – podlegają bowiem ochronie, z zastrzeżeniem postanowień rozdz. VII, jak utwory literackie, zatem kwestią decydującą o przydaniu danemu programowi ochrony prawnoautorskiej powinna być ocena, czy program spełnia przesłanki określone w art. 1 ust. 1 ustawy, czyli odznacza się oryginalnością oraz indywidualnością (stanowiąc „przejaw działalności twórczej o indywidualnym charakterze”). Nie da się jednakże dokonać tej oceny bez uwzględnienia nakazu proeuropejskiej interpretacji polskiego prawa, a co za tym idzie – orzeczeń Trybunału Sprawiedliwości, który przy wielu okazjach wypowiadał się na temat ochrony programów komputerowych³⁵.

Użyta w art. 1 ust. 3 Dyrektywy 91/250/EWG definicja oryginalności – „własna intelektualna twórczość autora” – jest spełniona, jeżeli program został stworzony samodzielnie, nie zaś skopiowany z istniejącego dzieła, i wykazuje przy tym choć minimalny ślad wysiłku umysłowego o charakterze kreatywnym³⁶. Żadnych innych kryteriów nie stosuje się przy dokonywaniu jego kwalifikacji do ochrony. Zgodne jest to również z koncepcją „kleine Münze”, czyli przyznawania ochrony dziełom nawet o niewielkim wkładzie pracy twórczej.

Jednocześnie mieć należy na uwadze fakt, że programy komputerowe w przeważającej liczbie przypadków pisane są³⁷ jako połączenie wielu podprogramów, z których każdy wykonuje określoną przez programistę funkcję³⁸. Każdy z tych programów, z których następnie tworzony jest większy program, przy zachowaniu warunków z art. 1 ust. 1 pr. aut., również jest programem – i w związku z tym przysługuje mu ochrona.

Kilka słów komentarza należy w tym miejscu poświęcić stanowisku Z. Okonia, który stwierdza:

³⁴ Oznaczać to jednocześnie może, że użyte skrypty czy biblioteki mogą mieć przymiot utworu w rozumieniu prawa autorskiego, jakkolwiek ochrona prawnoautorska będzie przysługiwać ich autorom.

³⁵ Liczne przykłady orzecznictwa w: *Prawo...*, red. D. Flisak, komentarz do art. 74 pr. aut.

³⁶ Zob. B. Czarnota, R.J. Hart, *Legal protection of computer programs in Europe – a guide to the EC directive*, Londyn, Dublin, Edynburg, Monachium 1991, s. 44; L. Bently, w: *Concise European copyright law*, red. T. Dreier, P.B. Hugenholtz, Alphen aan den Rijn 2006, s. 217.

³⁷ Zwłaszcza przy – najpopularniejszym obecnie – obiektowym podejściu do programowania. Zob. J. Jabłonowski, J. Sroka, sylabus do przedmiotu „Programowanie obiektowe”, < http://wazniak.mimuw.edu.pl/index.php?title=Programowanie_obiektowe >.

³⁸ Często się zdarza, że nad programem pracuje jednocześnie kilkanaście/kilkadziesiąt osób, z których każda wykonuje jakąś funkcjonalność programu, które następnie łączone są w całość.

Niewątpliwie ochrona programu komputerowego jest wykluczona, jeżeli jest on w całości zdeterminowany przez założoną funkcję, stanowi realizację narzuconych wskazówek, jedyny lub jeden z niewielu możliwych rezultatów pracy należycie wykwalifikowanego programisty, podejmującego się danego zadania. Na indywidualność w rozumieniu art. 1 ust. 1 pr. aut. jest więc miejsce tylko wówczas, gdy programista dysponuje, choćby niewielkim, marginesem swobody twórczej³⁹.

Autor nie ustrzegł się w tym sformułowaniu od wewnętrznej sprzeczności – jeśli istnieją choćby dwa możliwe rezultaty pracy programisty, to już mamy do czynienia właśnie z tym „choćby niewielkim marginesem swobody twórczej”. Zupełnie inną kwestią jest to, czy istnieje tylko jeden sposób rozwiązania danego zadania programistycznego, i pytanie, czy taka możliwość nie jest czysto hipotetyczna. Po podaniu dwóm programistom nawet bardzo szczegółowego opisu, co program ma robić, po wyznaczeniu języka programowania, funkcji, nazw zmiennych – jest zupełnie nieprawdopodobne, że owi programiści napisaliby taki program w identyczny sposób⁴⁰.

Uwzględniając powyższe rozważania, trudno także zgodzić się ze stanowiskiem przedstawionym przez M. Janowskiego⁴¹ dotyczącym nieuznawania za program komputerowy ciągu instrukcji zapisanych na kartce papieru. Sposób utrwalenia utworu na innym nośniku niż komputerowy nie odbiera przymiotu programu, podobnie jak zapisanie utworu muzycznego na kartce z wykorzystaniem pięciolinii nie pozbawia go cech utworu⁴². Odmawianie statusu programu (w znaczeniu informatycznym) w związku z zapisem na np. kartkach papieru stoi też w oczywistej sprzeczności zarówno z postanowieniami art. 74 ust. 2 pr. aut., jak i przywoływanym już art. 1

³⁹ Z. Okoń, w: *Prawo...*, komentarz do art. 74 pr. aut., teza 14.

⁴⁰ Można by wyobrazić sobie postawienie przed dziesięcioma pisarzami zdjęcia ulicy wraz z poleceniem opisania wskazanego domu i oszacowaniem prawdopodobieństwa, że każdy z nich użyje do tego tych samych przymiotników. Tak jak każdy pisarz ma własny styl pisania, tak ma go także każdy programista. Jeśli zadamy mu polecenie napisania programu wyświetlającego na ekranie „Ala ma kota”, większość napisze tak samo za pomocą jednej linijki kodu – jednakże napisanie choćby prostego kalkulatora da zupełnie inne kody źródłowe, mimo że kody wynikowe czy funkcje programu mogą być identyczne, co zostało już wyjaśnione powyżej.

⁴¹ Zob. M. Janowski, *Pojęcie programu komputerowego w obowiązującym Kodeksie karnym (rozważania na tle art. 278 § 2 k.k.)*, „Przegląd Sądowy” 2011, nr 2, s. 82.

⁴² Jednocześnie oczywistym jest, że ów zestaw instrukcji należy w celu wykonania programu wprowadzić do pamięci, a następnie „przełożyć” na język maszynowy rozumiany przez procesor.

ust. 2 Dyrektywy 2009/24/WE, a także orzecznictwem Trybunału Sprawiedliwości⁴³ i sądów krajowych⁴⁴.

Nie można również zgodzić się z definicją programu komputerowego zaproponowaną przez P. Kardasa, według którego program komputerowy to „zakodowany na odpowiednim nośniku informacji zapis składający się na utwór przedstawiający wartość materialną”⁴⁵. Tak nakreślona definicja znacznie przekracza uregulowania Dyrektywy 2009/24/WE, obejmując swym zakresem również np. wiersz zapisany w pliku tekstowym na płycie CD, a jednocześnie wykluczając program zapisany na kartce papieru. Należy też zwrócić uwagę na niefortunne użycie w definicji sformułowania: „zakodowany na odpowiednim nośniku informacji”. Kod programu komputerowego nie jest oczywiście zakodowany, bez względu na to, czy mówimy tu o kodzie źródłowym (rozumiałym dla człowieka), czy o kodzie wynikowym (binarnym), który jest wynikiem kompilacji oraz nie wymaga dla swojego istnienia żadnego „odpowiedniego nośnika”, co zostało wskazane powyżej. Autor trafnie poddaje krytyce stanowisko judykatury, w którym twierdzi się, że popełnienie przestępstwa stypizowanego w art. 287 § 2 k.k. polega na „objęciu przez sprawcę we władanie zapisu magnetycznego, za pomocą którego zakodowany został odpowiedni utwór nazywany programem komputerowym”⁴⁶. Zasadnie uważa, że realizacja znamion „nie musi łączyć się z uzyskaniem przez niego nośnika z zakodowanym programem”. Możliwa jest wszakże sytuacja, gdy sprawca zapamięta kod źródłowy cudzego programu komputerowego, a następnie odtworzy go z pamięci w późniejszym terminie. Zachowanie takie, pozbawione cechy fizycznego kopiowania, a tym bardziej zaboru nośnika, realizować będzie niewątpliwie znamiona określone w art. 278 § 2 k.k.

Nie można zgodzić się też ze stanowiskiem P. Kardasa, gdy pisze:

Wykonanie kopii programu komputerowego może być połączone z przełamaniem specjalistycznych zabezpieczeń, zainstalowanych w nośniku informacji, na

⁴³ Zob. wyrok TS z 22 XII 2010 r. w sprawie C-393/09 *Bezpečnostní softwarová asociace – Svaz softwarové ochrany p. Ministerstvo kultury* (zwłaszcza pkt 29–35), < <http://curia.europa.eu/juris/document/document.jsf?jsessionid=9ea7d0f130d6eef890-afaac943b488bad95bcf7eb8f2.e34KaxiLc3eQc40LaxqMbN4Pax0Te0?text=&docid=83458&pageIndex=0&doclang=pl&mode=lst&dir=&occ=first&part=1&cid=692972> >.

⁴⁴ Zob. wyrok SA w Łodzi z 4 II 2010 r., I ACa 912/09, niepubl.

⁴⁵ M. Dąbrowska-Kardas, P. Kardas, w: *Kodeks karny. Część szczególna. Tom III. Komentarz do art. 278–363*, red. A. Zoll, Warszawa 2016, komentarz do art. 278 k.k., teza 112.

⁴⁶ Wyrok SA w Krakowie z 8 VII 2009 r., II AKa 98/09, „Krakowskie Zeszyty Sądowe” 2009, z. 7–8, poz. 58.

którym zapisany jest program. W takim jednak przypadku sprawca będzie ponosił odpowiedzialność za przestępstwo z art. 279 § 1 (kradzież z włamaniem)⁴⁷.

Sformułowanie to można rozumieć dwojako. Obecnie w zasadzie każde płatne oprogramowanie (nieudostępniane przez twórcę za darmo) jest chronione specjalistycznymi zabezpieczeniami – zwyczajowo za pośrednictwem zwykłego klucza (tzw. *serial key*), którego wpisanie warunkuje dokończenie procesu instalacji oprogramowania. Nie można tracić też z pola widzenia tego, że do realizacji znamion z art. 278 § 2 k.k. nie jest konieczne używanie (zainstalowanie) programu, a wystarczy samo jego uzyskanie (choćby zapisanie go na dysku komputera czy posiadanie płyty z programem). Stanowisko prezentowane przez Autora rozumieć można tak, że w przypadku samego pobrania nielegalnego programu sprawca realizowałby znamiona art. 278 § 2 k.k., natomiast po przełamaniu zabezpieczenia, tj. skorzystaniu np. z cracka⁴⁸ i zainstalowaniu tego nielegalnie uzyskanego wcześniej programu, należałoby chyba przyjąć kwalifikowanie takiego zachowania jako kradzieży z włamaniem, co nie wydaje się trafnym rozwiązaniem.

Stoi to w sprzeczności z wyrażonym przez P. Kardasa stanowiskiem, według którego:

Zachowanie się sprawcy przestępstwa określonego w art. 279 § 1 polega na zaborze cudzej rzeczy ruchomej w celu przywłaszczenia, przy czym zabór musi zostać dokonany z włamaniem. Kradzież z włamaniem różni się od tzw. zwykłej kradzieży sposobem jej popełnienia. Charakteryzuje się ona połączeniem dwóch elementów – włamania oraz zaboru rzeczy⁴⁹.

Trafnie za warunek pociągnięcia sprawcy do odpowiedzialności karnej z tego artykułu Autor uznaje zabór cudzej rzeczy ruchomej. W przypadku „wykonania kopii programu komputerowego” nie dochodzi do zaboru cudzej rzeczy, co wyklucza, zgodnie z poglądem wyrażonym przez Autora, wspomnianą kwalifikację.

Jeśli natomiast rozumieć to sformułowanie w taki sposób, że przełamanie zabezpieczeń miałyby warunkować pobranie programu (włamanie na serwer firmy celem pobrania kopii oprogramowania), to poza przytoczonym już brakiem zaboru rzeczy ruchomej na względzie trzeba mieć następujące fakty. Zachowanie takie, w zakresie uzyskania dostępu do informacji, połączone

⁴⁷ M. Dąbrowska-Kardas, P. Kardas, w: *Kodeks...*, komentarz do art. 278 k.k., teza 119.

⁴⁸ Specjalny program, który omija bądź łamie zabezpieczenia np. gry, umożliwiając instalację nielegalnego oprogramowania.

⁴⁹ M. Dąbrowska-Kardas, P. Kardas, w: *Kodeks...*, komentarz do art. 279 k.k., teza 10.

z przełamaniem lub nawet obejściem zabezpieczeń⁵⁰ realizowałyby znamiona z art. 267 § 1 k.k., stanowiąc w tym względzie *lex specialis* wobec art. 279 k.k. Poprawną kwalifikacją zdaje się w tym przypadku kumulatywny zbieg przepisów pomiędzy art. 267 § 1 oraz art. 278 § 2 k.k.⁵¹, co szerzej zostanie omówione w dalszej części tekstu.

Wydaje się, że jedynym sposobem dokonania kradzieży z włamaniem programu komputerowego jest zabór nośnika, na którym jest on zapisany, po wcześniejszym przełamaniu zabezpieczenia (włamanie do sklepu i kradzież płyt z programami).

Za całkowicie nietrafne należy uznać stanowisko J. Lachowskiego, który twierdzi, że:

[...] program komputerowy to zapis cyfrowy umożliwiający obsługę określonych aplikacji. Programem komputerowym nie jest gra komputerowa, piosenka czy też film. Są to utwory podlegające ochronie na podstawie art. 115 u.p.a.⁵²

Pomijając oczywistą nietrafność sformułowania, że to zapis cyfrowy determinuje istnienie programu komputerowego⁵³, zwrócić uwagę należy na inny problem. O ile w przypadku filmu czy piosenki nie mamy do czynienia z programami, to owo twierdzenie jest całkowicie błędne w przypadku gry komputerowej. To właśnie gry komputerowe są sztandarowymi przykładami programów komputerowych. Gry oczywiście bardzo często zawierają w swej strukturze inne utwory (ścieżka dźwiękowa, grafika itp.) – jakkolwiek nie może ulegać wątpliwości, że są to programy komputerowe⁵⁴. Nie wyklucza to oczywiście dualizmu ochrony prawnej programów komputerowych – tak na gruncie Kodeksu karnego, jak i przepisów karnych ustawy o prawie autorskim i prawach pokrewnych⁵⁵.

Według autora niniejszego opracowania za właściwą należy uznać decyzję o braku legalnej definicji programu komputerowego i mimo upływu niemal trzydziestu lat od *explanatory memorandum* Dyrektywy 91/250/EWG

⁵⁰ Obchodzenie zabezpieczeń przy wykorzystaniu luki oprogramowania jest zdecydowanie powszechniejszym sposobem dokonywania takich naruszeń.

⁵¹ Podobnie A. Wróbel, w: *Kodeks karny. Część szczególna. Komentarz. Tom II*, red. A. Zoll, Warszawa 2013, s. 1509–1510.

⁵² J. Lachowski, w: *Kodeks...*, komentarz do art. 278 k.k., teza 4.

⁵³ Jak wykazywano wcześniej, program może być zapisany na kartce papieru.

⁵⁴ Dość powiedzieć, że większe gry pisze się głównie w C++, C, Assemblerze, C# czy JAVA, a polska gra „Wiedźmin 3” ma ponad milion linii kodu źródłowego.

⁵⁵ Rodzi to jednakowoż liczne problemy dotyczące właściwej kwalifikacji tych zachowań, jednak przekracza to ramy tego opracowania.

powody nieutworzenia definicji legalnej nie uległy zmianie. Zdaje się, że najpełniejsza jest pierwotnie zakreślona „definicja” zaprezentowana na potrzeby Dyrektywy 91/250/EWG, w której za program uznano wyrażony w dowolnej formie, języku, notacji albo kodzie zestaw instrukcji, których celem jest wykonanie przez komputer zadanych funkcji lub zadań. Tylko tak szeroko zakrojona definicja, przy uwzględnieniu różnorodnych ograniczeń poruszonych powyżej, pozwala objąć swym zakresem różnorodny zakres oprogramowania komputerowego, jakkolwiek w mojej ocenie ma ona wyłącznie charakter negatywny – tzn. pozwala stwierdzić, co programem nie jest. Spełnienie warunku powyższej definicji nie zawsze będzie oznaczało, że mamy do czynienia z programem komputerowym⁵⁶.

5. Uzyskanie programu komputerowego na gruncie Kodeksu karnego

Przyjmuje się, że:

[...] „uzyskanie” jako znamię przestępstwa nie musi się łączyć z pozbawieniem prawnego użytkownika władztwa nad programem komputerowym. Jego istota polega natomiast na nielegalnym zwielokrotnieniu programu komputerowego, jako że „uzyskanie” cudzego programu nie jest możliwe bez jego skopiowania. Chodzi przy tym o zwielokrotnienie polegające zarówno na sporządzaniu trwałych fizycznych kopii programu (np. na dyskietkach), jak i [na] skopiowaniu plików programu w pamięci komputera⁵⁷.

Interesujący problem powstał w związku z rozwojem usług internetowych, ściślej – chmury obliczeniowej⁵⁸ (*cloud service*). Rozpoczęty kilka lat temu i coraz dynamiczniej rozwijający się rynek usług świadczonych w chmurze zmienia nieco postrzeganie znamienia „uzyskania cudzego programu komputerowego”. Coraz popularniejsze stają się usługi świadczone w modelu SaaS – *Software as a Service*⁵⁹ (oprogramowanie jako usługa).

⁵⁶ Strona napisana wyłącznie w języku HTML spełnia taką definicję, a programem nie jest.

⁵⁷ A. Adamski, *Przestępstwa komputerowe w nowym kodeksie karnym*, w: *Nowa kodyfikacja karna. Kodeks karny. Krótkie komentarze*, Warszawa 1998 (z. 17), s. 115.

⁵⁸ Szerzej na temat chmury komputerowej: < <https://azure.microsoft.com/pl-pl/overview/what-is-cloud-computing> >.

⁵⁹ Szerzej na temat SaaS: M. Cusumano, *Cloud computing and SaaS as new computing platforms*, „Communications of the ACM” 2010, t. 53, nr 4, s. 27–29, < http://ebusiness.mit.edu/research/papers/2010.01_Cusumano_Technology%20Strategy%20and%20Management_272.pdf >.

Zapewne większość z czytelników korzystała z Gmaila, Dysku Google, Kalendarza Google, Dokumentów Google czy z Office 365. To nic innego jak programy komputerowe udostępniane przez Internet, do których dostęp uzyskujemy za pomocą przeglądarki internetowej. Taki model dystrybucji oprogramowania powoli zaczyna przejmować domenę zarezerwowaną dla lokalnych instalacji⁶⁰. Sztandarowym przykładem może być np. program LEX. Jeszcze kilka lat temu jego obsługa wiązała się z koniecznością instalacji co miesiąc (lub co tydzień – w zależności od wersji) otrzymywanych pocztą płyt DVD z aktualizacją. W kancelariach i urzędach, w których było wielu użytkowników, proces taki zajmował nawet kilka godzin i angażował wiele osób. Przeciwstawmy temu modelowi produkt tej samej firmy, mający tę samą funkcjonalność, tę samą treść, dostępny z każdego komputera na świecie przez przeglądarkę internetową, który zawsze jest aktualny, zawsze działa szybko – i nie wymaga miejsca na dysku⁶¹.

Czym od zainstalowanego lokalnie na komputerze programu Word różni się Office 365 i Word dostępny przez przeglądarkę? Spośród tysięcy oferowanych przez ten program funkcjonalności różnicuje go w zasadzie wyłącznie możliwość łatwej pracy wspólnej i wygodniejsze przechowywanie plików w chmurze, a co za tym idzie – możliwość rozpoczęcia pracy na komputerze w biurze i jej kontynuacji w domu na innym komputerze⁶². A więc z punktu widzenia użytkownika – w zasadzie niczym istotnym. Czy różnią się w znaczeniu informatycznym? Oczywiście tak, jakkolwiek nie są to różnice, które w mojej ocenie winny rzutować na ich odmienne postrzeganie na gruncie art. 278 § 2 k.k. Różnica sprowadza się bowiem do miejsca, w którym przetwarzane są dane programu. W przypadku instalacji

⁶⁰ Nie zagłębiając się nadto w techniczne aspekty, warto tylko zwrócić uwagę na zalety takiego modelu dystrybucji oprogramowania. Dla końcowego użytkownika oznacza oszczędności związane z brakiem konieczności utrzymywania infrastruktury w firmie (serwera, na którym będzie instalowany dany program, z którego będą korzystał pracownicy), brak konieczności regularnej aktualizacji oprogramowania, bo przez przeglądarkę dostępna jest zawsze najnowsza wersja (co paradoksalnie jest korzystne również dla producenta oprogramowania, który ma kontrolę nad korzystaniem ze starszych wersji, mających np. luki w zabezpieczeniach), a przede wszystkim występuje całkowita kompatybilność między systemami (nie ma konieczności tworzenia kilku wersji dla różnych systemów operacyjnych – program uruchomiony w przeglądarce będzie wszędzie wyglądać tak samo).

⁶¹ Pełna instalacja LEX Omega to ponad 35 GB, co przy coraz popularniejszych dyskach SSD, które standardowo montowane w laptopach mają 128 lub 256 GB, jest znaczną wartością.

⁶² Funkcjonalność tę oczywiście można uzyskać w tradycyjnym Wordzie, zapisując plik w lokalizacji sieciowej.

na komputerze użytkownika zwyczajowo włączenie programu oznacza (w największym uproszeniu) odwołanie systemu do miejsca na dysku, gdzie znajdują się pliki programu, i rozpoczęcie ich przetwarzania – załadowanie do pamięci RAM, wyświetlenie okna programu i oczekiwanie na interakcję ze strony użytkownika.

Weźmy za przykład program LEX. Po kliknięciu np. w zakładkę „Kodeksy” następuje zapytanie do odpowiedniej części bazy danych⁶³, obsługa przez program otrzymanych danych i przekazanie ich karcie graficznej celem wyświetlenia na ekranie. Zapytania obsługiwane są przez procesor i pamięć RAM, którą użytkownik ma zamontowaną w komputerze. W przypadku modelu SaaS i np. programu LEX dostępnego przez przeglądarkę zapytania są za jej pośrednictwem kierowane do systemu operacyjnego zainstalowanego na komputerze zdalnym, który obsługuje udostępniany program, a obsługiwane są przez procesor i pamięć RAM zlokalizowane na serwerze dostawcy aplikacji. Następnie przetworzone wyniki przesyłane są do przeglądarki użytkownika, celem ich wyświetlenia oraz w oczekiwaniu na jego interakcję.

Czy zatem fakt obsługi programu na komputerze zdalnym, a nie lokalnym, winien determinować realizację znamion przestępstwa opisanego w omawianym przepisie? Nietrudno wyobrazić sobie sytuację, gdy sprawca wynajmuje serwer wirtualny, na którym instaluje wspomniany już program LEX. Łącząc się zdalnie z komputerem za pomocą przeglądarki lub innego programu⁶⁴, korzysta z tak zainstalowanego nielegalnego oprogramowania. Zdaje się nie budzić wątpliwości, że w tak nakreślonym stanie faktycznym oczywistym byłoby pociągnięcie go do odpowiedzialności karnej za przestępstwo z art. 278 § 2 k.k. Irracjonalnie byłoby jednocześnie sądzić, że inna osoba, która nie dokonała tej instalacji, a korzysta z tego programu zdalnie za przyzwoleniem sprawcy, odpowiedzialności by nie ponosiła. Nie można bowiem tracić z pola widzenia dobra, które stoi za tym przepisem. Chroni ono bowiem prawa majątkowe dysponenta majątkowych praw do tego programu i zapobiega nielegalnemu obrotowi jego pirackich wersji, sankcjonując jednocześnie bezprawne uzyskanie programu.

Trzeba mieć na uwadze, że przez ostatnie dwadzieścia lat zmienił się gruntownie sposób dystrybucji nielegalnego oprogramowania. W czasie uchwalania obecnie obowiązujących przepisów podstawowym sposobem jego dystrybucji były giełdy i sprzedawane na nich pirackie płyty CD/DVD.

⁶³ Oczywiście zainstalowanej lokalnie wraz z plikami programu.

⁶⁴ Choćby zwykłego TeamViewera.

Dziś dostępność szerokopasmowego Internetu zmieniła sposób rozpowszechniania nielegalnego oprogramowania i przeniosła ten proceder do sieci. Rozpatrując zatem znamiona tego artykułu, nie można wyklądać ich w zupełnym oderwaniu od dzisiejszych realiów i postępu technologicznego.

Powyższe stanowisko może początkowo rodzić wątpliwości, sprowadzające się do stwierdzenia, że przecież w takim przypadku korzystamy ze strony internetowej, a nie z programu komputerowego. W tym stwierdzeniu kryje się jednak błąd, polegający na rozłącznym traktowaniu pojęć „program komputerowy” i „strona internetowa”, podczas gdy większość obecnie funkcjonujących serwisów internetowych jest w istocie programami komputerowymi – w całości bądź w dużej części.

Kilkanaście lat temu większość stron internetowych była „statyczna”, tzn. prezentowana zawartość, przygotowywana przez programistę, funkcjonowała niejako jak zadrukowana kartka i była prezentowana odwiedzającemu za pomocą języka HTML⁶⁵. Oczywiście strona taka nie była programem komputerowym⁶⁶. Dynamiczny rozwój Internetu i olbrzymi wzrost jego popularności sprawił, że zaczęły powstawać nowe narzędzia programistyczne, które znacząco rozszerzyły możliwości informacji prezentowanych na stronie, wychodząc poza tekst i grafikę. Pojawiły się języki programowania⁶⁷ i napisane w nich systemy zarządzania treścią, które umożliwiły rozwój dynamicznych stron internetowych⁶⁸. W roku 2005 pojawił się AJAX⁶⁹, dokonując rewolucji na rynku usług sieciowych. Nagle możliwym stało się zaoferowanie użytkownikom aplikacji dostępnej za pośrednictwem strony

⁶⁵ „HyperText Markup Language” – nie jest to język programowania w ścisłym tego słowa znaczeniu. Służy do informowania przeglądarki, w jaki sposób ma wyświetlać tekst na stronie. Nawet jego najnowsza wersja HTML5 – mimo zdecydowanie większych możliwości – nie jest uznawana za język programowania. (Aplikacje programowane są tak naprawdę w języku JavaScript, a HTML5 służy jedynie do ich wyświetlania).

⁶⁶ Podobnie uznał OLG we Frankfurcie w wyroku z 22 III 2015 r., 11 U 64/04, JurPC Web-Dok. 92/2005, ust. 1–51, za: Z. Okoń, w: *Prawo...*, komentarz do art. 74 pr. aut., teza 10.

⁶⁷ Python, Pearl, Ruby, JavaScript – to tylko niektóre z nich.

⁶⁸ W odróżnieniu od stron statycznych – strony dynamiczne generowane są na bieżąco przez serwer na podstawie zmiennych i parametrów przekazanych poprzez przeglądarkę internetową.

⁶⁹ „Asynchronous JavaScript and XML”, asynchroniczny JavaScript i XML – technika tworzenia aplikacji internetowych, w której interakcja użytkownika z serwerem odbywa się bez przeładowywania strony, w sposób asynchroniczny, co umożliwia dynamiczną interakcję z użytkownikiem (wcześniej akcja użytkownika oznaczała konieczność przeładowania całej strony). Zob. J.J. Garrett, *Ajax: a new approach to web applications*, < <http://adaptivepath.org/ideas/ajax-new-approach-web-applications> >.

internetowej⁷⁰. Powstała pierwsza większa aplikacja internetowa wykorzystująca zalety tej technologii – Gmail. Kolejno pojawiało się coraz więcej serwisów, które wykorzystuje niemal każdy z nas na co dzień, zbudowanych w oparciu o AJAX: Google Maps, Google Drive czy YouTube (to tylko niektóre przykłady). Symultanicznie rozwijały się inne technologie, jak Microsoft Silverlight, Java ze swoimi apletami, JavaFX czy spinający wszystko JavaScript. Wszystkie je łączy jedno – wykorzystane zostały do tego, żeby zaofiarować użytkownikom komputerów oprogramowanie dostarczane za pośrednictwem przeglądarki internetowej.

Należy w tym miejscu powtórzyć argumenty podniesione powyżej. Czym różni się program pocztowy zainstalowany na komputerze od pełnego różnorodności funkcjonalności Gmaila? Odpowiedź brzmi: wyłącznie procesorem, który obsługuje nasze zapytania. A gdyby Gmaila można było pobrać na dysk i zainstalować – czy ktokolwiek powiedziałby wówczas: „to nie jest program komputerowy”? Z pewnością odpowiedź jest negatywna. Z czego zatem miałyby wynikać odebranie tego przymiotu i ochrony, jaką daje Kodeks karny, programowi, który funkcjonuje i komunikuje się z użytkownikiem wyłącznie za pośrednictwem strony internetowej i przez nią jest udostępniany?⁷¹

Nie jest oczywiście tak, że dzisiaj każda strona internetowa jest programem komputerowym⁷². Statyczna strona, na której znajduje się sam tekst, rzecz jasna programem nie jest. Co jednak jeśli na tej stronie za pomocą skryptu zamieścimy kalkulator napisany w języku JavaScript? Strona nie stanie się nagle programem, ale skrypt (który jest programem) nie straci tego przymiotu przez zamieszczenie go na stronie. Jak bowiem wspomniano wcześniej, bardzo często na stronach wykorzystuje się skrypty napisane np. w JavaScript, realizujące jakąś funkcjonalność, które są w istocie samodzielnymi programami, częstokroć połączonymi z zewnętrzną bazą danych. Czy są to skrypty autorskie, czy udostępniane darmowo przez twórców w Internecie – nie ma to wpływu na fakt, że są to programy, które współdziałając ze sobą, wzbogacają funkcje strony. Jednocześnie w bardziej rozbudowanych serwisach internetowych na stronie zdarzają się dziesiątki skryptów, z których każdy samodzielnie może być oceniany jako program

⁷⁰ Gwoli ścisłości dodać należy, że taka możliwość istniała również wcześniej, jednakże w zdecydowanie bardziej skomplikowanej i mniej popularnej formie.

⁷¹ Jakkolwiek jest dostępna wtyczka do przeglądarki Chrome, która umożliwia korzystanie z Gmaila w trybie offline.

⁷² Co nie wyklucza oczywiście jej ochrony na gruncie innych przepisów prawa autorskiego jako utworu.

komputerowy. Jeśli popatrzymy na największe serwisy, to liczba linii kodu iść może w miliony, czyniąc je jednymi z najbardziej rozbudowanych programów komputerowych⁷³.

Analizując przedstawiony przykład korzystania z nielegalnie zainstalowanego na serwerze oprogramowania, nie można nie odnieść się do treści art. 267 § 2 k.k., który stanowi: „Tej samej karze [dwóch lat pozbawienia wolności – przyp. A.B.] podlega, kto bez uprawnienia uzyskuje dostęp do całości lub części systemu informatycznego”. Oczywiście jest, że serwer komputerowy jest systemem informatycznym, a zatem uzyskanie dostępu do niego bez uprawnienia realizować będzie znamiona tego typu. Poprzez uzyskanie dostępu do części systemu⁷⁴ rozumieć trzeba, oczywiście, uzyskanie dostępu nie tylko do systemu operacyjnego, ale do dowolnej jego części, co oczywiście oznacza również nieuprawniony dostęp do innego oprogramowania na nim zainstalowanego. Uzyskanie zatem nieuprawnionego dostępu do wspomnianego wcześniej oprogramowania (LEX Online, Office 365) – co nie musi wiązać się z przełamaniem żadnych zabezpieczeń – realizować będzie znamiona opisane w tym paragrafie, przy jednoczesnej realizacji znamion art. 278 § 2 k.k. Kumulatywna kwalifikacja prawna takiego zachowania winna obejmować obydwa przepisy, tj. art. 278 § 2 k.k. w zw. z art. 267 § 2 k.k., co uzasadnione jest różnymi dobrami prawnymi przezeń chronionymi: autorskie prawa majątkowe w przypadku art. 278 § 2 k.k. oraz bezpieczeństwo systemów informatycznych w przypadku art. 267 § 2 k.k.

⁷³ Według szacunków Windows 7 to ok. 40 mln linii kodu, a Facebook – ok. 60 mln (z czego ok. 10 mln to *front end*, czyli warstwa prezentowana na stronie). Zob. < <http://www.informationisbeautiful.net/visualizations/million-lines-of-code> >.

⁷⁴ Zgodnie z brzmieniem art. 1a Konwencji Rady Europy o cyberprzestępczości (sporządzonej w Budapeszcie dn. 23 XI 2001 r., Dz.U. 2015, poz. 728) system informatyczny oznacza: „każde urządzenie lub grupę wzajemnie połączonych lub związanych ze sobą urządzeń, z których jedno lub więcej, zgodnie z programem, wykonuje automatyczne przetwarzanie danych”; zgodnie z art. 2 Dyrektywy Parlamentu Europejskiego i Rady 2013/40/UE z dn. 12 VIII 2013 r. dotyczącej ataków na systemy informatyczne i zastępującej decyzję ramową Rady 2005/222/WsiSW (Dz.U. UE L z dn. 14 VIII 2013 r.), system informatyczny oznacza: „urządzenie lub grupę wzajemnie połączonych lub powiązanych ze sobą urządzeń, z których jedno lub więcej, zgodnie z programem, dokonuje automatycznego przetwarzania danych komputerowych, jak również danych komputerowych przechowywanych, przetwarzanych, odzyskanych lub przekazanych przez to urządzenie lub tę grupę urządzeń w celach ich eksploatacji, użycia, ochrony lub utrzymania”; zgodnie z art. 7 pkt 2a ustawy z dn. 29 VIII 1997 r. o ochronie danych osobowych (Dz.U. 2016, poz. 992 tekst jedn. ze zm.) przez system informatyczny rozumie się: „zespół współpracujących ze sobą urządzeń, programów, procedur przetwarzania informacji i narzędzi programowych zastosowanych w celu przetwarzania danych”.

Nie wyklucza to oczywiście sytuacji, w której sprawca, przełamując albo omijając elektroniczne, magnetyczne, informatyczne lub inne szczególne jej zabezpieczenie, uzyskuje dostęp do programu. Takie zachowanie, co zostało poruszone powyżej, należałoby kumulatywnie kwalifikować z art. 267 § 1 w zw. z art. 278 § 2 k.k.⁷⁵

6. Podsumowanie

Prawnokarna ochrona programów komputerowych, ze względu na konieczność rozważania przy ich ocenie szczegółowych aspektów technicznych warunkujących przyznanie i zakres ich ochrony, rodzić może wiele problemów praktycznych, których w mniej oczywistych przypadkach, bez pomocy biegłego, nie sposób rozwiązać. Artykuł miał za zadanie – poza zwróceniem uwagi na kilka zagadnień istotnych z punktu widzenia wykładni przepisów ustawy – na przybliżenie czytelnikowi przedmiotu ochrony, jakim jest program komputerowy.

W rezultacie przeprowadzonych rozważań można postawić następujące tezy:

1. Jedynym językiem, który „rozumie” komputer (procesor), jest kod maszynowy.

2. Kompilacja różnych kodów źródłowych programu może dać takie same kody wynikowe, a kompilacja takich samych kodów źródłowych może dać inne kody wynikowe.

3. Rozbudowany graficznie program (w znaczeniu informatycznym), wykonujący niezliczone funkcje i operacje, może nie być utworem w rozumieniu prawa autorskiego.

4. Program komputerowy objęty ochroną może składać się z wielu podprogramów.

5. Napisanie skomplikowanego programu przez dwóch programistów w identyczny sposób, nawet jeśli posiadaliby taki sam, bardzo szczegółowy opis zadania przed nimi postawionego, jest zupełnie nieprawdopodobne.

6. Program komputerowy może być utrwalony na dowolnym nośniku, włącznie z pamięcią ludzką.

7. Wydaje się, że jedynym sposobem dokonania kradzieży z włamaniem programu komputerowego jest zabór nośnika z tymże, po wcześniej-

⁷⁵ Podobnie P. Kozłowska-Kalisz, w: *Kodeks karny. Komentarz*, red. M. Mozgawa, Warszawa 2015, komentarz do art. 267 k.k., teza 16.

szym przełamaniu zabezpieczenia (włamanie do sklepu i kradzież płyt z programami).

8. Za właściwą należy uznać decyzję o braku legalnej definicji programu komputerowego. Mimo upływu niemal trzydziestu lat od *explanatory memorandum* Dyrektywy 91/250/EWG powody niestwarzania definicji legalnej nie uległy zmianie.

9. Strona internetowa może być programem komputerowym.

10. Na stronie internetowej może znajdować się wiele programów.

11. Uzyskanie programu komputerowego w rozumieniu art. 267 § 2 k.k. może być zrealizowane również poprzez uzyskanie dostępu do serwera, na którym zainstalowany jest nielegalnie uzyskany program, bądź poprzez uzyskanie dostępu do niego za pośrednictwem przeglądarki internetowej.

12. Uzyskanie nieuprawnionego dostępu do oprogramowania udostępnionego w chmurze (LEX Online, Office 365) – co nie musi wiązać się z przełamaniem żadnych zabezpieczeń – realizować będzie znamiona opisane w art. 267 § 2 k.k., przy jednoczesnej realizacji znamion art. 278 § 2 k.k. Kumulatywna kwalifikacja prawna takiego zachowania winna obejmować obydwie przepisy, tj. art. 278 § 2 k.k. w zw. z art. 267 § 2 k.k.

Definition and protection of a computer program in criminal law

Summary

This article deals with the problem of interpretation of the concept of computer program in criminal law, and interpretation problems arising from the development of information technology. These issues are addressed by using dogmatic and comparative methods. In this context article presents the history of the concept of computer program in legal acts, including international and European ones. In the following part a critical analysis of the definition of computer program in the Polish Criminal Code is presented, as well as author's conclusions regarding the legal qualification of cybercrimes. The conclusions of the article cover both the informatic issues important for the interpretation and application of criminal law, and the legal implications of this position. One of the main conclusions of the article is that there is no justification for implementation of a legal definition of computer program into the Criminal Code, regardless of the problems with its application associated with an improper understanding of computer program by lawyers.

Key words: computer program, cloud computing, programming language, copyright law, programing libraries, website

Adam Behan – Uniwersytet Jagielloński, Katedra Prawa Karnego (doktorant)