SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Evidenčné číslo: FEI-5382-5982

MODERNÉ TECHNIKY NA UKRÝVANIE ?INNOSTI MALVÉRU

BAKALÁRSKA PRÁCA

2019 **Lukáš Gnip**

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Evidenčné číslo: FEI-5382-5982

MODERNÉ TECHNIKY NA UKRÝVANIE ?INNOSTI MALVÉRU

BAKALÁRSKA PRÁCA

Študijný program:	Aplikovaná informatika
Číslo študijného odboru:	2511
Názov študijného odboru:	9.2.9 Aplikovaná informatika
Školiace pracovisko:	Ústav informatiky a matematiky
Vedúci záverečnej práce:	Mgr. Ing. Matú? Jókay, PhD.

Bratislava 2019 **Lukáš Gnip**

Slovenská technická univerzita v Bratislave
Ústav informatiky a matematiky

Fakulta elektrotechniky a informatiky
Akademický rok: 2012/2013
Evidenčné číslo: FEI-5382-5982

STU FEI

ZADANIE BAKALÁRSKEJ PRÁCE

Študent: **Michal Ližičiar**
ID študenta: 5982
Študijný program: Aplikovaná informatika
Študijný odbor: 9.2.9 aplikovaná informatika
Vedúci práce: Ing. Matúš Jókay, PhD.

Názov práce: **Anonymizácia internetového prístupu**

Špecifikácia zadania:

Cieľom práce je vytvoriť zásuvný modul pre internetový prehliadač, ktorý bude schopný buď náhodne alebo selektívne meniť informácie používané na identifikáciu používateľa pri jeho prístupe na cieľový server.

Úlohy:
1. Analyzujte dostupnosť a funkčnosť podobných modulov.
2. Analyzujte informácie používané na identifikáciu používateľa pri prístupe na stránku.
3. Navrhnite, implementujte a otestujte anonymizačný modul pre zvolený internetový prehliadač.

Zoznam odbornej literatúry:

1. YARDLEY, G. Better Privacy. [online]. 2012. URL: http://nc.ddns.us/BetterPrivacy/BetterPrivacy.htm.
2. ECKERSLEY, P. A Primer on Information Theory and Privacy. [online]. 2010. URL: https://www.eff.org/deeplinks/2010/01/primer-information-theory-and-privacy.

Riešenie zadania práce od: 24. 09. 2012
Dátum odovzdania práce: 24. 05. 2013

L.S.

Michal Ližičiar
študent

prof. RNDr. Otokar Grošek, PhD.
vedúci pracoviska

prof. RNDr. Gabriel Juhás, PhD.
garant študijného programu

SÚHRN

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Študijný program:	Aplikovaná informatika
Autor:	Lukáš Gnip
Bakalárska práca:	Moderné techniky na ukrývanie ?innosti malvéru
Vedúci záverečnej práce:	Mgr. Ing. Matú? Jókay, PhD.
Miesto a rok predloženia práce:	Bratislava 2019

Práca sa zaoberá vytvorením zásuvného modulu pre internetový prehliadač, ktorý modifikuje informácie používané na identifikáciu používateľa pri prístupe na server. V prvej časti práce sa nachádza prehľad metód, ktoré zvyšujú anonymitu pri prehliadaní webových stránok. Práca tiež obsahuje zoznam dnes najpoužívanejších rozšírení, ktorých úlohou je zmena niektorých identifikačných prvkov prehliadača alebo anonymizácia pomocou špeciálnych techník. V ďalšej časti sa nachádza prehľad charakteristík prehliadača. Kombináciou týchto charakteristík sa dá s vysokou mierou úspešnosti identifikovať používateľ, ktorý danú stránku navštívil. Posledná časť práce obsahuje návrh, implementáciu a testovanie rozšírenia vytvoreného pre internetový prehliadač Mozilla Firefox. Popisuje zdrojový kód rozšírenia, súvislosť medzi charakteristikami prehliadača, zistené obmedzenia a postup riešenia. Výsledné rozšírenie zvyšuje anonymitu používateľa modifikáciou niektorých charakteristických prvkov prehliadača alebo blokovaním odosielania prvkov, ktoré nie je možné v rámci rozšírenia zmeniť. Na rozdiel od dnes najpoužívanejších modulov dokáže rozšírenie okrem modifikácie HTTP hlavičky, meniť aj charakteristiky zisťované pomocou JavaScript príkazov.

Kľúčové slová: anonymizácia, identifikácia pou?ívate?a, zásuvný modul, Mozilla Firefox, internet

ABSTRACT

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
FACULTY OF ELECTRICAL ENGINEERING AND INFORMATION TECHNOLOGY

Study Programme:	Applied Informatics
Author:	Lukáš Gnip
Bachelor Thesis:	Anonymization of internet access
Supervisor:	Mgr. Ing. Matú? Jókay, PhD.
Place and year of submission:	Bratislava 2019

The bachelor thesis is about creating of a plugin for web browser, that modifies information used to identification of user during accessing a server. There is an overview of methods that increase anonymity during browsing websites, in the first part. The thesis also contains a list of the most used extensions nowadays, that function is a change of some identification components of browser or special ways of anonymization. In the next part of the thesis is an overview of the characteristics of web browser. By combination of these characteristics we can with high level of success identify a user, who have visited the web site. The last part of thesis contains project, implementation and testing of extension created for the web browser Mozilla Firefox. There is also description of source code of extension, the link between the characteristics of web browser, detected limitations and way how to solve them. The resulting extension increases anonymity of user by modification of some characteristic components of web browser or by blocking sending components, that can not be in extension changed. In comparison with most used modules nowadays, this module can modify HTTP headers including characteristics detected by JavaScript commands.

Keywords: anonymization, identification of user, plugin, Mozilla Firefox, internet

Obsah

Zoznam obrázkov a tabuliek

Zoznam skratiek a značiek

WWW - World Wide Web

Zoznam algoritmov

Úvod

Pri každom využívaní internetového prehliadača zanechávame v celosvetovej sieti internet stopy, ktoré o nás dokážu veľa vecí prezradiť. Zoznam navštívených stránok prezrádza informácie o našich záľubách, záujmoch, ale v istých súvislostiach dokáže prezradiť aj naše zamestnanie alebo školu, na ktorej študujeme. Reklamné spoločnosti napríklad na základe týchto údajov dokážu cielene zamerať reklamy, ktoré sa nám pri surfovaní zobrazujú a tým zvyšujú svoje zisky.

Existuje niekoľko metód, pomocou ktorých sa dá aspoň...

1 Malvér a spôsoby ukrytia malvéru

1.1 Malvér

Je to softvér, ktorého cieľom je poškodť, zablokovať, zmocniť sa alebo odcudziť dôležité informácie uložené v počítači. Cieľom malvéru je získať informácie pre útočníka a následné zneužitie informácií na rôzne druhy nelegálnej činnosti za účelom danej obete uškodiť alebo sa na nej finančne obohatiť. Malvér kedysi zah??al rôzne počítačové vírusy, backdoory, spyware, ramsomware a rôzne iné softvéry. V súčasnosti už toto delenie nie je aktuálne, pretože súčasné malvéry sú už kombináciou týchto vírusov a využívajú rôzne časti jednotlivých vírusov. V nasledujúcej časti kapitoly sa podrobnejšie budem zaoberať rôznymi spôsobmi ukrytia malvéru, ktoré môžu byť v súčasnosti využívané na ukrytie malvéru.

1.2 DLL Injection / Reflective DLL Injection

DLL Injection je technika používaná na ukrytie funkcie na spustenie malvéru vo vnútri iného programu. Najžastejšie sa na to využíva alokovaná pamäť vyhradená pre beh infikovaného programu, kde sa funkcia na spustenie malvéru ukryje a pre antivírusové programy je ťažko detekovateľná. Po nakopírovaní DLL funkcie do alokovanej pamäte iného programu môže byť následne vyvolané spustenie infikovaného programu a spolu s ním aj spustenie malvéru. Hlavným cieľom DLL Injection je škodlivý kód ukryť do oficiálneho alebo overeného programu, kde je malvér ukrytý pred antivírovým softvérom, odkiaľ môže byť následne spustený.

1.3 Process hollowing

Proces Hollowing využíva rovnaké alebo podobné princípy ukrývania škodlivého softvéru ako DLL Injection. Cieľom je schovať škodlivý kód do existujúce programu z ktorého sa po spustení, spustí aj volanie škodlivého malvéru. Oproti DLL Injection ide ale ukrytie malvérového programu do iného programu. Malvér si podľa potreby alokuje virtuálnu pamäť v pamäti iného programu. Po spustení infikovaného programu malvér pozastaví thread v ktorom program beží, následne zmení obsah legitímneho súboru zmapovaním pamäte cieľového procesu. Po zmapovaní uvoľní všetku pamäť programu a alokuje pamäť pre malvér a zapíše každú z častí malvéru do cieľovej pamäte programu. Malvér volá SetThreadContext, aby ukazoval vstupný bod na novú časť kódu, ktorú napísal. Na konci malvér obnoví pozastavený thread volaním ResumeThread, aby sa proces dostal z pozastaveného stavu a nasledovným spustením programu umožní získavanie údajov.

1.4 Thread Execution Hijacking

Táto technika ukrytia malvéru spočíva v napojení sa na už existujúci thread vyvolaný iným programom. Po získaní prístupu do threadu malvér uvedie thread do pozastaveného režimu aby vykonal vloženie volania škodlivého malvéru do threadu iného procesu. Malvér si alokuje virtuálnu pamäť v programe a následne do tejto pamäte vloží škodlivý shell kód, ktorý obsahuje cestu k volaniu DLL so škodlivým malvérom. Po vykonaní týchto úkonov, spustí pozastavený thread programu. Nevýhodou takéhoto spustenia pozastaveného programu je, že môže spôsobiť zlyhanie systému v rámci systémového volania. Preto aby sa tomu predišlo modernejší malvér proces ukončí a vyvolá ho znovu s už modifikovanou zmenou na infikovanie.

1.5 Portable Executable Injection

Výhodou tohto spôsobu ukrytia malvéru je využitie nakopírovania malvéru do už existujúceho bežiaceho procesu pomocou shell skriptu, ktorý vyvolá spustenie čkodlivého malvéru. Namiesto toho aby proces prepisoval cesty volaním DLL, táto technika zapisuje obsah malvéru priamo pomocou WriteProcessMemory. Počas doby zapisovania aby malvér nebol ľahko odhalený využíva vnorené cykly, ktoré spomaľujú systém buď diagnostikou alebo volaním zbytočných funkcií a snaží sa zahltiť systém a neumožniť skorú diagnostiku malvéru. Keď malvér preformuluje všetky potrebné adresy, všetko, o musí urobiť, je odovzdať jeho počiatočnú adresu CreateRemoteThread a nechať spustiť malvér.

1.6 Hook injection

Hook injection je technika používaná na zachytávanie volania funkcií. Malvér môže využívať hook injection funkcie na načítanie škodlivého súboru v DLL pri spustení udalosti v konkrétnom vlákne. Malvér na to využíva volanie funkcie SetWindowsHookEx, ktorý obsahuje štyri parametre. Prvý je tip udalosti na, ktorý sa spustí škodlivý malvér napríklad na stlačenie klávesnice alebo tlačidla na myši. Druhý je ukazovateľ na funkciu, ktorá sa ma po stlačení daného tlačidla vykonať. Tretí je modul obsahujúci danú funkciu. Preto je pred volaním funkcie SetWindowsHookEx vidieť volania do LoadLibrary. Posledný parameter je vlákno ktoré má vykonať procedúru a hook injection. Pokiaľ je posledný parameter prázdny alebo nastavený na nulu tak danú procedúru vykonajú všetky bežiace vlákna. Malvér sa ale zameriava len na jedno vlákno, kvôli zníženiu detekcii svojej práce.

1.7 APC Injection

Škodlivý softvér môže využívať výhody asynchrónnych volaní procedúr (APC), aby prinútil ďalšie vlákno spustiť svoj vlastný kód jeho pripojením do fronty cieľového vlákna. Každé vlákno má rad asynchrónnych volaní procedúr, ktoré čakajú na vykonanie, keď cieľové vlákno vstupuje do zmeniteľného stavu. Vlákno vstúpi do výstra?ného stavu, ak volá funkcie SleepEx, SignalObjectAndWait, MsgWaitForMultipleObjectsEx, WaitForMultipleObjectsEx alebo WaitForSingleObjectEx. Malvér zvyčajne hľadá akékoľvek vlákno, ktoré je v zmeniteľnom stave, a potom zavolá OpenThread a QueueUserAPC, aby zaradil APC do vlákna. Čo umožňuje jeho následným spustením infikovať zariadenie malvérom.

1.8 Extra windows memory injection

Tento spôsob schovávania softvéru sa spolieha na rozširovanie pamäte aplikačných okien v systéme. Pri otváraní nového okna aplikácie, softvér špecifikuje ďalšie bajty pamäte, ktoré rozšíria veľkosť alokovanej pamäte pre spustenú aplikáciu. Tento proces sa nazýva extra windows memory(EWM). V tejto časti ale nevzniká dostatok miesta na uloženie údajov. Aby sa toto obmedzenie obišlo, škodlivý softvér zapíše kód do zdieľanej sekcie aplikácie a do EWM vloží ukazovateľ na danú časť. Do tejto rozšírenej časti zieľanej pamäte ďalej softvér zapíše ukazovateľ funkcie do pamäte, ktorá obsahuje shell skript na spustenie malvéru. Malvér následne nastaví v EWM fukciu SetWindowLong na zmenu hodnôt na zadanom ofsete. Čím malvér môže jednoducho zmeniť posun ukazovateľa funckie pamäti aplikácie a nasmerovať ho do EWM, na kód so škodlivým shell skriptom.

2 Aktuálne vyu?ívané malvéry

Táto kapitola obsahuje opis jednotlivých malvérov používaných v roku 2019, ktoré boli detekované spoločnosťami ako Avast a McAfee. Tieto malvéry sú najčastejšie využívané v oblasti Európy. Kapitola obsahuje bližší opis malvérov, ich využitie, použité spôsoby úrokov a ukrytie malvéru v systéme.

2.1 Sodinokibi

Tento malvér bol detekovaný v období okolo apríla 2019. Patrí do rodiny ransomvéru, ktorých cieľom je šifrovať informácie v zariadení a následne za dešifrovanie pýta nemalý obnos peňazí. Názov bol objavený v hash kóde, ktorý obsahoval názov "Sodinokibi.exe". vírus sa šíri sám zmeužívaním zraniteľnosti na serveroch Oracle WebLogic. Softvér je navrhnutý tak, aby rýchlo vykonával šifrovanie definovaných súborov v konfigurácii ransomvéru. Prvou akciou škodlivého softvéru je získať vťetky funkcie potrebné pri behu programu a vytvoriť dynamický IAT, ktorý sa pokúsi zahltiť volanie systému Windows statickou analýzou. Po zahltení systému dôjde k spusteniu malvéru. Technika využívaná na ukrytie malvéru je Portable Executable Injection ktorú volá po spomalení RunPE na jej spustenie z pamäte. táto technika ukrytia malvéru je opísaná v predošlej kapitole. Analýza spoločnosti McAfee ukazuje podobnosť s iným starším malvérom GandCrab.(pridať odkaz na analýzu respektíve literatúru)

2.2 Emotet

Emotet je malvér, ktorý sa primárne šíri pomocou rôznych spam emailov. Na infikovanie zariadenia používa rôzne skripty, makrá v dokumetoch alebo linky. Emotet stavia na infikovaní pomocou sociálneho inžinierstva. Prezentuje sa ako hodnoverný zástupca napr. banky, Rôznych internetových obchodov, atď. . Emotet malvér sa prvýkrát objavil v roku 2014 kedy vyuťíval na infikovanie rôzne JavaScript súbory. V roku 2019 sa tento vírus objavil znova tentokrát v pokročilejšej verzii. V novej verzii je Emotet polymorfným malvérom čo mu umožňuje vyhnúť sa klasickej detekcii. Emotet používa modulárne knižnice Dynamic Link (DLL) na nepretržitý vývoj a aktualizáciu svojich schopností. Emotet môže navyše generovať falošné indikátory, ak je spustený vo virtuálnom prostredí čo zhoršuje jeho detekciu systéme.

2.3 ZeuS

Prvýkrát odhalený v roku 2007 sa Zeus Trojan, ktorý sa často nazýva Zbot, stal jedným z najúspešnejších kúskov botnetového softvéru na svete, postihol milióny počítačov

a vytvoril množstvo podobných kusov škodlivého softvéru vytvoreného z jeho kódu. Po jeho minimalizovaní sa znovu objavil v pozmenenej podobe so zameraním na odchytávanie bankových operácii (Odchytávanie prihlasovacích údajov do internet bankingu). Dosahuje to prostredníctvom monitorovania webových stránok a zaznamenávania klávesov, keď malvér zistí, že sa používateľ nachádza na bankovej webovej stránke, začne zaznamenávať stlačenia klávesov použité na prihlásenie. To znamená, že trójsky kôň dokáže obísť zabezpečenie na týchto webových stránkach. Infekcia prebieha pomocou spamov. Keď užívateľ klikne na odkaz v správe alebo stiahne obsah súboru. spolu s ním stiahne a spustí aj makro. Ktoré po nainštalovaní umožňuje sledovanie zariadenia.

2.4 Dridex

Dridex je známy trójsky kôň, ktorý sa špecializuje na krádež kreditných údajov online bankovníctva. Tento typ škodlivého softvérusa objavil v roku 2014 a stále napreduje a vyvíja. Nový variant Dridex je schoplný vyhnúť sa detekcii tradičnými antivírusovými produktami. Dridex tiež zvýšil svoju infraštruktúru knižníc. Využíva knižnice ako DLL Dridex čo sú 64-bitové knižnice DLL - s pridruženými hashmi SHA256 - ktoré používajú názvy súborov načítané legitímnymi spustiteľnými súbormi systému Windows. Názvy súborov a hash sa však obnovujú a menia zakaždým, keď sa obeť prihlási do infikovaného hostiteľa Windows. Tento malvér je v súčastnosti schopný detekovať približne 25 až 30 percent aktuálnych antivírusových softvérov.

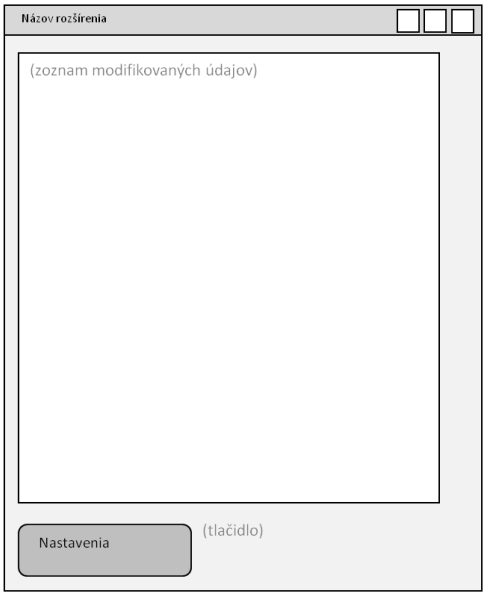
2.5 Mirai

Mirai je samo sa množiaci vírus botnetov. Zdrojový kód pre Mirai bol autorom verejne sprístupnený po úspešnom a dobre propagovanom útoku na webovú stránku Krebbs. Kód botnetu Mirai infikuje zle chránené internetové zariadenia pomocou telnetu (sieťový komunikačný protokol založený na TCP)na nájdenie tých, ktoré stále používajú svoje predvolené uživateľské meno a heslo. Účinnosť systému Mirai je spôsobená jeho schopnosťou infikovať desiatky tisíc týchto nezabezpečených zariadení a koordinovať ich tak, aby začali útok DDoS proti vybranej obeti.

Mirai má dve hlavné zložky, samotný vírus a veliteľské a kontrolné stredisko (CnC). Vírus obsahuje útočné algoritmy, Mirai má desať algoritmov, ktoré môže spustiť, a proces skenovania, ktorý aktívne hľadá iné zariadenia, aby ohrozil ich fungovanie. CnC je samostatný obraz, ktorý ovláda kompromitované zariadenia (BOT) a posiela im pokyny na spustenie jedného z útokov proti jednej alebo viacerým obetiam. Proces skenera prebieha nepretržite na každom infikovanom PC pomocou protokolu telnet (na porte TCP

23 alebo 2323)

CnC predstavuje jednoduché rozhranie príkazového riadku, ktoré umožňuje útočníkovi určiť útočný algoritmus, IP adresu obete a trvanie útoku. CnC tiež čaká na to, aby jej existujúce BOTy vrátili novoobjavené adresy zariadení a poverenia, ktoré používa na skopírovanie vírusového kódu a následne na vytváranie nových BOTov. Útokové algoritmy sú konfigurovateľné z CnC, ale v predvolenom nastavení má Mirai tendenciu náhodne rozdeľovať rôzne polia (čísla portov, poradové čísla, identifikátory atď.) V útočných paketoch, takže sa menia s každým odoslaným paketom.



Obrázok 1: Predpokladaný vzh?ad roz?írenia.

Dôle?itou po?iadavkou kladenou na roz?írenie bolo príjemné pou?ívate?ské rozhranie.[**?**] Z tohto dôvodu malo roz?írenie obsahova? zoznam modifikovaných vlastností a tla?idlo pre prístup k nastaveniam roz?írenia v jednoduchej a praktickej forme. Predpokladaný vzh?ad je zobrazený na obrázku ?. 1.

Algoritmus 1 Uká?ka algoritmu

```
1  /* Hello World program */
2
3  #include<stdio.h>
4
5  struct cpu_info {
6      long unsigned utime, ntime, stime, itime;
7      long unsigned iowtime, irqtime, sirqtime;
8  };
9
10 main()
11 {
12     printf("Hello World");
13 }
```

Záver

Cieľom práce bola analýza anonymizačných modulov, identifikačných prvkov prehliadača a vytvorenie anonymizačného modulu pre internetový prehliadač.

Analýzou najpoužívanejších modulov a vlastností prehliadača, ktoré slúžia na identifikáciu používateľa, sme zistili aktuálny stav a funkcionalitu rozšírení, ktorými je možné anonymizovať prístup na internet. Väčšina týchto rozšírení modifikuje len časť vlastností prehliadača, ktoré sú odosielané na server, alebo úplne blokuje ich odosielanie. Nami vytvorené rozšírenie dokáže modifikovať väčšinu identifikačných prvkov rozšírenia, pričom dodržiava súvislosti medzi vlastnosťami (používateľský agent odosielaný v hlavičke dopytu je totožný s používateľským agentom zisťovaním pomocou JavaScript príkazu, súvislosť medzi šírkou a dĺžkou rozšírenia obrazovky). Dokáže blokovať údaje, ktoré sú posielané v otvorenej podobe na server a obsahujú informácie o identifikačných údajoch prehliadača, ktoré sa nedajú na úrovni rozšírení modifikovať.

Testovanie rozšírenia nám overilo funkčnosť a správnosť implementácie. Rozšírenie dokáže buď vždy, alebo v časových intervaloch modifikovať väčšinu charakteristických prvkov prehliadača odsielaných na server, a tým zvyšuje anonymitu používateľa.

Prílohy

A ?truktúra elektronického nosi?a

\
\Bakalarska_praca.pdf
\FEIk_Identuty.xpi
\FEIkIdentity
\FEIkIdentity\chrome.manifest
\FEIkIdentity\install.rdf
\FEIkIdemtity\content
\FEIkIdemtity\content \function.js
\FEIkIdemtity\content \options.xul
\FEIkIdemtity\content \overlay.xul
\FEIkIdemtity\content \window.js
\FEIkIdemtity\content \window.xul
\FEIkIdemtity\defaults
\FEIkIdemtity\defaults\preferences
\FEIkIdemtity\defaults\preferences \prefs.js
\FEIkIdemtity\locale
\FEIkIdemtity\locale \sk-SK
\FEIkIdemtity\locale \sk-SK\options.dtd
\FEIkIdemtity\locale \sk-SK\window.dtd
\FEIkIdemtity\skin

B Algoritmus

Algoritmus B.1 Ukážka algoritmu

```
1  /* Hello World program */
2
3  #include<stdio.h>
4
5  struct cpu_info {
6      long unsigned utime, ntime, stime, itime;
7      long unsigned iowtime, irqtime, sirqtime;
8  };
9
10 main()
11 {
12     printf("Hello World");
13 }
```