

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Tvorba škodlivých dokumentov

Patrik Velčický

(BIT semestrálne zadanie)

Cvičenie: Pondelok 10:00

Cvičiaci: Ing. Ján Skalný

Rok: 2022

Obsah

Zoznam obrázkov	3
Úvod	4
Šírenie škodlivých dokumentov	4
Sociálne inžinierstvo	5
Phishing	5
Obfuskácia	5
Najčastejšie druhy škodlivých dokumentov	5
HTML - HyperText Markup Language	5
RTF - Rich Text Format	5
DOC – Microsoft Word	6
PDF – Portable Document Format	6
Programy na vytváranie škodlivých dokumentov	6
Lucky Strike	6
Office-DDE-Payloads	6
wePWNise	6
MacroShop	7
Macro_pack	7
Worse-PDF	7
VBA makrá	7
Vytváranie škodlivého dokumentu	8
Vytvorenie dokumentu typu .doc	8
Zapnutie Developer tabu	9
Vytvorenie VBA makra	11
Sociálne inžinierstvo	16
Obfuskácia VBA makra	19
Sken škodlivého dokumentu pomocou VirusTotal	19
Bibliografia	21

Zoznam obrázkov

Obrázok 1 Útok pomocou škodlivého dokumentu	4
Obrázok 2 Exploit vs Macro	8
Obrázok 3 Uloženie dokumentu ako .doc.....	8
Obrázok 4 Obsah škodlivého dokumentu.....	9
Obrázok 5 Zapnutie Developer tab Krok-1	10
Obrázok 6 Zapnutie Developer tab Krok-2	10
Obrázok 7 Zapnutie Developer tab Krok-3	10
Obrázok 8 Vloženie InkPicture Krok-1	11
Obrázok 9 Vloženie InkPicture Krok-2	11
Obrázok 10 Vloženie InkPicture Krok-3	12
Obrázok 11 Vzhľad dokumentu po vložení InkPicture.....	13
Obrázok 12 Vzhľad dokumentu po vložení InkPicture Show/Hide.....	14
Obrázok 13 Vloženie VBA makra Krok-1	15
Obrázok 14 Vloženie VBA makra Krok-2	15
Obrázok 15 Vloženie VBA makra Krok-3	15
Obrázok 16 Vložené VBA makro	16
Obrázok 17 Vloženie TextBox	16
Obrázok 18 TextBox tab.....	17
Obrázok 19 Vzhľad dokumentu po vložení TextBox	17
Obrázok 20 Premenovanie TextBox Krok-1	18
Obrázok 21 Premenovanie TextBox Krok-2	18
Obrázok 22 Premenovanie TextBox Krok-3	18
Obrázok 23 Sken dokumentu bez obfuskácie.....	19
Obrázok 24 Sken dokumentu s obfuskáciou.....	20

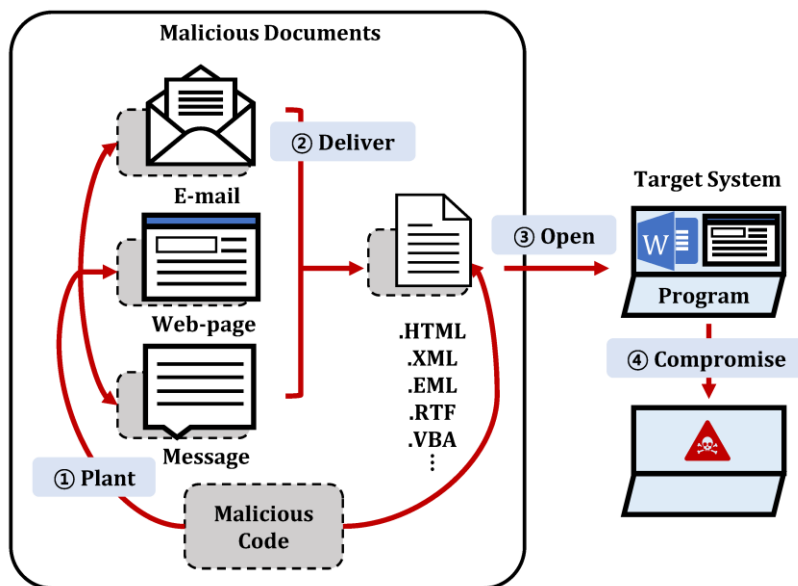
Úvod

Škodlivé dokumenty sú dokumenty, ktoré obsahujú samo-spustiteľný kód alebo kód, ktorý vyžaduje, aby používateľ pred spustením udelil povolenie, poprípade so škodlivým dokumentom nejakým spôsobom interagoval. Tieto dokumenty môžu byť napríklad PDF so zabudovaným škodlivým java skriptom alebo dokument Microsoft Office s vloženými VBA makrami. Škodlivé dokumenty sa najčastejšie doručujú používateľom prostredníctvom phishingových e-mailov, môžu sa však doručovať aj prostredníctvom fyzických USB kľúčov alebo iných typov útokov sociálneho inžinierstva. V mnohých prípadoch, napríklad pri dokumentoch Microsoft Office, sa od používateľa vyžaduje interakcia s dokumentom pred úspešným spustením akéhokoľvek VBA makra. Po otvorení škodlivého dokumentu a vykonaní akejkoľvek požadovanej interakcie medzi používateľom a škodlivým dokumentom, sa spustí škodlivý kód, ako napríklad Powershell, cmd alebo podobný skriptovací kód na nadviazanie komunikácie s útočníkom, stiahnutie dát alebo vykonanie rôznych akcií so zámerom poškodenia obete (1).

Šírenie škodlivých dokumentov

Útočník vloží škodlivý kód do elektronického dokumentu, tento škodlivý dokument doručí obeti prostredníctvom webovej stránky, e-mailu alebo správy maskovanej ako neškodná a presvedčí obeť, aby si stiahla a otvorila ním zaslaný škodlivý dokument. Keď obeť škodlivý dokument otvorí pomocou legitímneho, ale zraniteľného programu, škodlivý kód skrytý v dokumente sa aktivuje (2).

Tento proces je opísaný aj pomocou Obrázok 1 Útok pomocou škodlivého dokumentu.



Obrázok 1 Útok pomocou škodlivého dokumentu¹

¹ https://www.mdpi.com/applsci/applsci-12-04088/article_deploy/html/images/applsci-12-04088-g001.png

Sociálne inžinierstvo

V kontexte informačnej bezpečnosti je sociálne inžinierstvo psychologická manipulácia ľudí, aby vykonali akcie alebo prezradili dôverné informácie. Jedná sa o trik, ktorý vyvoláva pocit dôvery za účelom zhromažďovania informácií, podvodu alebo prístupu k systému, ktorý sa líši od tradičného podvodu v tom, že je často jedným z mnohých krokov v zložitejšej schéme podvodov (3).

Phishing

Phishing je druh sociálneho inžinierstva a neustála hrozba. K phishingu často dochádza pri e-mailovom spoofingu alebo okamžitých správach. Phishingové e-maily obsahujú nereálne požiadavky alebo hrozby. Používajú niekoľko fráz alebo slov ako „vyžaduje sa okamžitá akcia“, „vyhrali ste telefón“, „vyhrali ste v lotérii“, „kliknite na odkaz na pôžičku“, čo používateľovi poukazuje na naliehavosť situácie, aby čo najskôr vykonal akciu, ktorú od neho daný e-mail alebo správa žiada. Niektoré z hlavných ukazovateľov phishingového e-mailu sú prudko stúpajúci počet hypertextových odkazov a počet obrázkov, ktoré slúžia ako hypertextové odkazy (4).

Obfuskácia

Pri vývoji softvéru je obfuskácia akt vytvárania zdrojového alebo strojového kódu, ktorý je pre ľudí alebo počítače ťažko pochopiteľný. Programátori môžu úmyselne obfuskovať kód, aby zakryli jeho účel (zabezpečenie prostredníctvom nejasností) alebo jeho logiku, alebo implicitné hodnoty, ktoré sú v ňom vložené, predovšetkým s cieľom zabrániť neoprávnenej manipulácii alebo reverznému inžinierstvu. Obfuskácia sa dá urobiť ručne alebo pomocou automatizovaného nástroja, ktorý je v priemysle preferovanou technikou (5).

Najčastejšie druhy škodlivých dokumentov

HTML - HyperText Markup Language

Dynamický HTML dáva útočníkom silnú techniku na kompromitovanie počítačových systémov. Škodlivý dynamický kód HTML je zvyčajne vložený do bežnej webovej stránky. Táto škodlivá webová stránka svoju obeť infikuje, keď ju používateľ prehliada. Okrem toho sa takéto dynamický HTML kód môže ľahko maskovať obfuskovaním alebo transformáciou, čo ešte viac sťažuje jeho detekciu. Antivírusové softvéry bežne používajú prístupy založené na podpisoch, ktoré nemusia byť schopné efektívne identifikovať maskované škodlivé kódy HTML (6).

RTF - Rich Text Format

Súbory RTF patria medzi najpopulárnejšie formáty súborov používané pri phishingových útokoch. Rich Text Format vyvinula spoločnosť Microsoft v rokoch 1987 až 2008 a naďalej ho podporujú operačné systémy Windows, Mac a Linux. RTF bol vytvorený s cieľom umožniť multiplatformové výmeny dokumentov. Tento formát súboru je už roky obľúbeným cieľom výskumníkov v oblasti zraniteľnosti a vývojárov exploitov, pretože môže hostiť rôzne typy objektov. Typy objektov zahŕňajú: anotácie, fonty, obrázky, OLE a SWF. To umožňuje doručovať exploity z rôznych typov objektov (7).

DOC – Microsoft Word

Škodlivý dokument Microsoft Word má dlhú históriu v oblasti kybernetickej bezpečnosti a rýchlo sa rozrástá s obrovským výskytom útokov. Z dôvodu maskovania obfuskovaním a zložitosti nie sú bežné metódy detekcie ideálne a špecifické metódy detekcie sú tiež obmedzené. Binárny formát dokumentu Word podporuje úlohy spracovania textu. Tieto úlohy zahŕňajú vytváranie a manipuláciu s textom, obrázkami, tabuľkami a správou vlastných XML schém, ktoré sú spojené s obsahom dokumentu. Existujú tri dôvody, prečo sa dokumenty Word stali tak populárne pri útokoch. Po prvé, škodlivý kód vložený do dokumentov je široko používaný. Po druhé, Microsoft Word má veľa používateľov a zložitú štruktúru. Po tretie, je ľahké vložiť objekt do dokumentu bez toho, aby bol objavený. Pri útoku hackeri vždy používajú techniky sociálneho inžinierstva alebo malvér, prípadne kombináciu oboch, takže Microsoft Word je ideálnym nosičom (8).

PDF – Portable Document Format

Vďaka svojej všestrannej funkčnosti a veľkému rozšíreniu sa dokumenty PDF stali populárnou cestou na zneužívanie používateľov, od rozsiahlych phishingových útokov až po cieľené útoky. Mnohé nedávne štúdie ukázali, že škodlivé dokumenty PDF sa často používajú pri phishingových útokoch spojených so sociálnym inžinierstvom na vysokej úrovni, ktoré páchajú skupiny vysoko sofistikovaných a vytrvalých útočníkov, ktorých cieľom je špionáž. PDF dokumenty sa stávajú jedným z najpopulárnejších formátov súborov využívaných pre cieľené útoky (9).

Programy na vytváranie škodlivých dokumentov

Lucky Strike

Jedná sa o generátor škodlivých dokumentov typu .xls alebo .doc v prostredí PowerShell. Všetky payloady, ktoré vytvoríte sa ukladajú do databázy, aby ste ich mohli ľahko vyhľadať a vložiť do nového alebo existujúceho dokumentu. Nástroj Lucky Strike je určený na automatizáciu vytvárania škodlivých payloadov. Tento Powershell skript je riadený pomocou menu. K dispozícii sú tiež zabudované techniky na obchádzanie antivírusových programov (10).

Office-DDE-Payloads

Jedná sa o kolekciu skriptov a šablón na generovanie dokumentov balíka Office s DDE technikou, ktorá spúšťa príkazy bez použitia makier. Po inštalácii ho môžeme spustiť pomocou pythonu ddeexcel.py pre Excel alebo python ddeword.py pre Word. DDE je skratka pre Dynamic Data Exchange. DDE posiela správy medzi aplikáciami, ktoré zdieľajú údaje, a zdieľanú pamäť na výmenu údajov medzi aplikáciami, takže používame jednu aplikáciu na spustenie ďalšej (10).

wePWNise

Tento nástroj generuje VBA kód nezávislý od architektúry, ktorý sa používa v dokumentoch alebo šablónach balíka Microsoft Office. Taktiež automatizuje obchádzanie kontroly aplikácií a softvér na zmiernenie zneužitia. Nástroj bol navrhnutý s ohľadom na automatizáciu a integráciu (10).

MacroShop

Ide o zbierku skriptov určených na pomoc pri doručovaní payloadov prostredníctvom makier balíka Microsoft Office (10).

Macro_pack

Macro_pack sa používa na automatizáciu obfuskácie a generovania dokumentov Microsoft Office, skriptov VB, skratiek a iných formátov pre pentestovanie (10).

Worse-PDF

Worse-PDF zmení normálny súbor PDF na škodlivý. To môže byť užitočné na získanie dôvery vašich obetí. Najmä ak by od vás pravdepodobne očakávali legitímny PDF dokument (10).

VBA makrá

Visual Basic for Applications (VBA) je jednoduchý programovací jazyk. Jedná sa o odnož jazyka Visual Basic, počítačového jazyka, ktorý Microsoft vyvinul v 90. rokoch. Tento programovací jazyk je k dispozícii pokročilým používateľom Microsoft Office, aby pomohol automatizovať úlohy. Microsoft Office odhaľuje množstvo udalostí, funkcií a objektov, ktoré možno použiť na manipuláciu s dokumentmi, databázami, formulármi, tabuľkami, prezentáciami a v konečnom dôsledku aj s počítačom. VBA skutočne umožňuje programátorovi vykonávať takmer akúkoľvek operáciu používateľa, ako je kliknutie myšou, stlačenie klávesov, otváranie dialógových okien, ukladanie súborov, zadávanie údajov atď. Primárna výhoda VBA je, že je zabudovaná do Microsoft Office. Takže aj keď sa verzia a funkcie mohli mierne zmeniť, základné funkcie a schopnosti tu čakali na použitie a zneužitie od Office 97 a až po Office 365 (11).

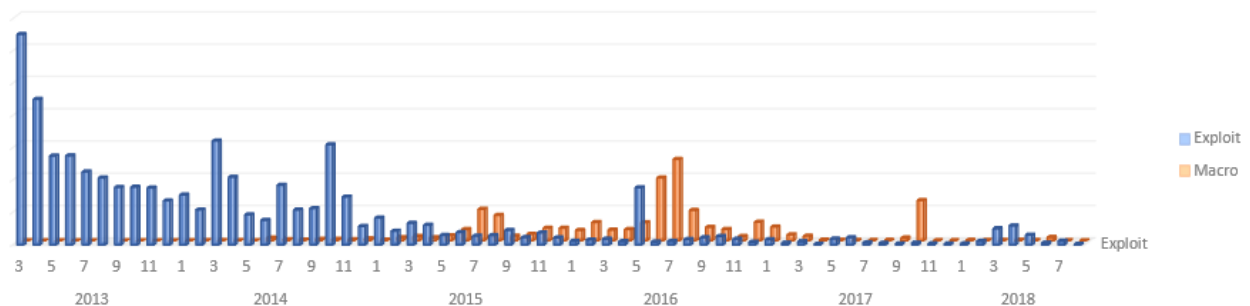
VBA môže vyzeráť ako iné programovacie jazyky, no zároveň to môže byť trochu náročné. VBA je vo svojej podstate objektovo založený jazyk, ako napríklad Python, C++ alebo iné skriptovacie jazyky. Aplikácie Microsoft Office odhaľujú rôzne objekty, metódy, vlastnosti atď., ktoré môže programovací jazyk použiť. VBA interaguje s aplikáciou využívaním výhod týchto objektov a odosielaním pokynov alebo volaním metód. Tento objektový model odhaľuje nielen objekty balíka Microsoft Office, ale aj základné objekty operačného systému, ako aj ďalšie knižnice a komponenty. Objekty predstavujú rôzne časti dokumentu alebo samotnej aplikácie, napríklad aktuálny dokument, strana, zošit, pracovný hárok, bunka a snímka. Metódy sú funkcie spojené s konkrétnym objektom, napr. Uložiť a vybrať. Vlastnosti sú údaje alebo premenné spojené s konkrétnymi objektmi, napr. hodnota a obsah. Okrem všeobecného objektového modelu existujú ďalšie všeobecné funkcie, ktoré môžu prijímať argumenty a vracať hodnoty. Tieto argumenty môžu byť objekty alebo jednoduchšie typy (11).

Akonáhle je makro spustené pod rúškom „oficiálneho“ dokumentu Microsoft Office, má voľnú kontrolu nad systémom. Makrá môžu infikovať súbory, poškodiť iné časti systému, stiahnuť a nainštalovať softvér alebo robiť čokoľvek iné, čo si útočník želá (12).

Keďže škodlivý kód je súčasťou makra, nie všetky antivírusové programy to považovali za hrozbu. Nakoniec Microsoft a bezpečnostná komunita začali tvrdo zasahovať. V predvolenom nastavení zakázali makrá počnúc Microsoft Office 2007, čo prinútilo používateľa povoliť makrá v

dokumente. Tým boli útoky trochu zložitejšie. Antivírusové programy začali skenovať dokumenty na makrá s otvoreným textom, keďže VBA je, koniec koncov, skriptovací jazyk. Keď útočníci začali obfuskovať ich kód, Microsoft predstavil Antimalware Scan Interface (AMSI), ktorý bol schopný skenovať skriptovacie funkcie volané za behu namiesto zahmlených textových reťazcov v dokumentoch. Táto zvýšená bezpečnosť prinútila útočníkov hľadať alternatívne metódy na spustenie kódu, konkrétne exploits, ale teraz, keď je čoraz ťažšie nájsť exploits a bezpečnostné spoločnosti sa proti nim zlepšujú, makrá sa vracajú (12).

Na Obrázok 2 Exploit vs Macro môžeme vidieť množstvo exploit a makro útokov za posledné roky.

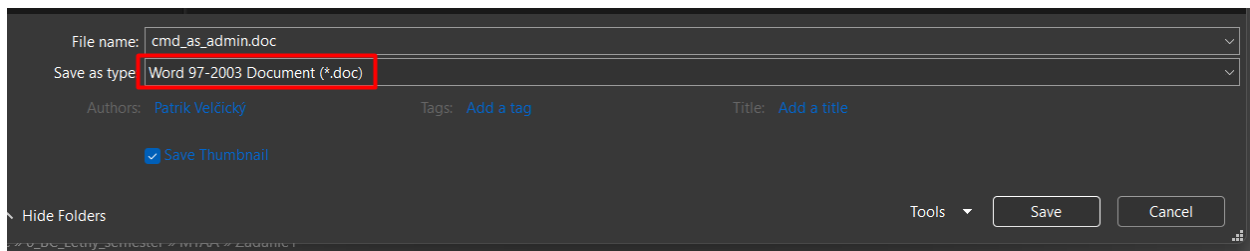


Obrázok 2 Exploit vs Macro²

Vytváranie škodlivého dokumentu

Vytvorenie dokumentu typu .doc

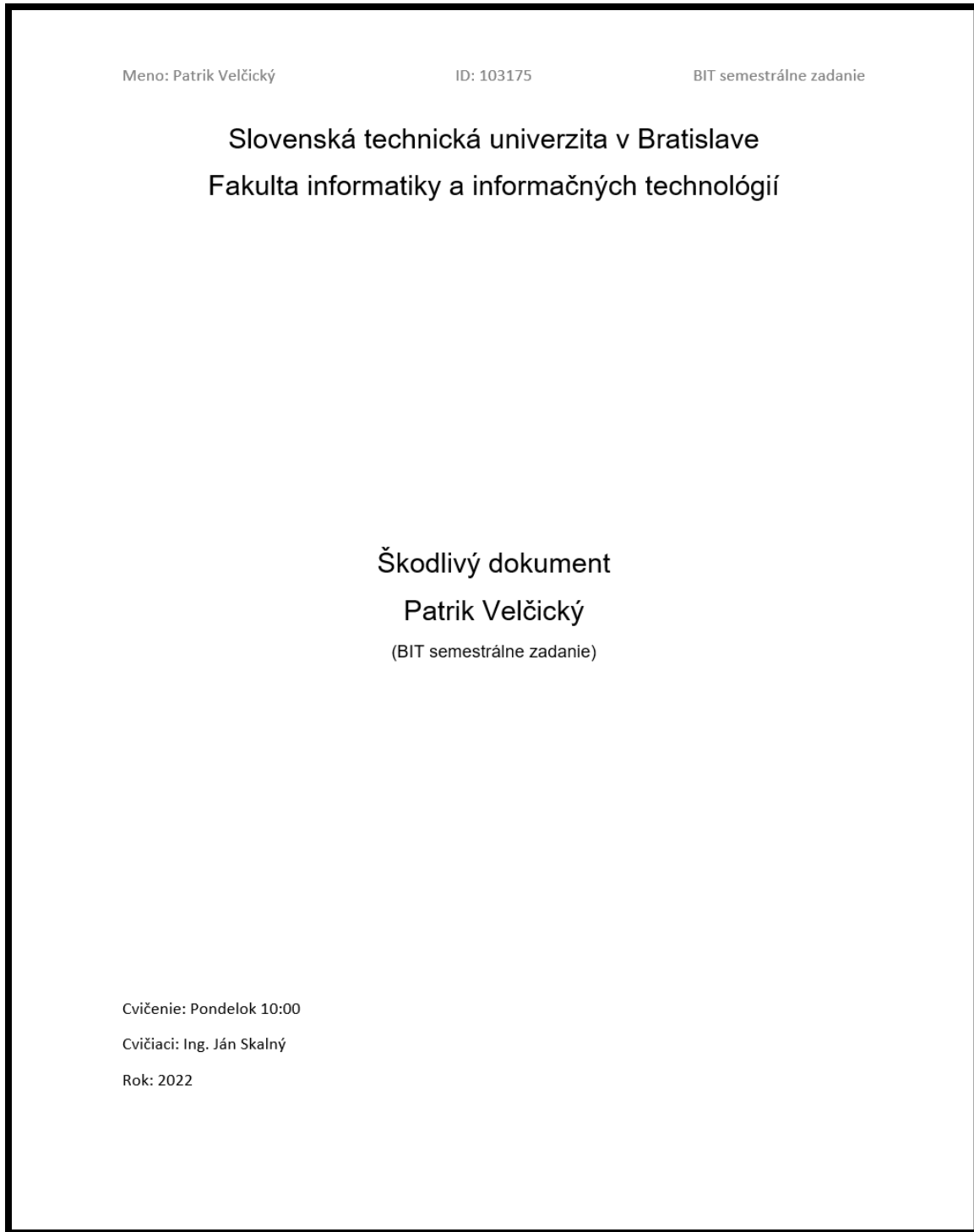
Ako prvé si musíme **vytvoriť Microsoft Word dokument typu .doc**. Dodržanie .doc formátu je nutné z dôvodu vloženia VBA makra. Novšie typy ako napríklad .docx neumožnia uložiť dokument, ktorý obsahuje makrá a požadujú uloženie dokumentu s makrami pod špeciálnou príponou .docm, ktorá má úplne inú ikonku ako klasický Microsoft Word dokument.



Obrázok 3 Uloženie dokumentu ako .doc

Nasledujúci krok je vytvoriť obsah škodlivého dokumentu. Obsah môjho škodlivého dokumentu môžete vidieť na Obrázok 4 Obsah škodlivého dokumentu.

² <https://www.microsoft.com/security/blog/uploads/securityprod/2018/09/fig1-prevalence-of-exploit-vs-macro-2.png>

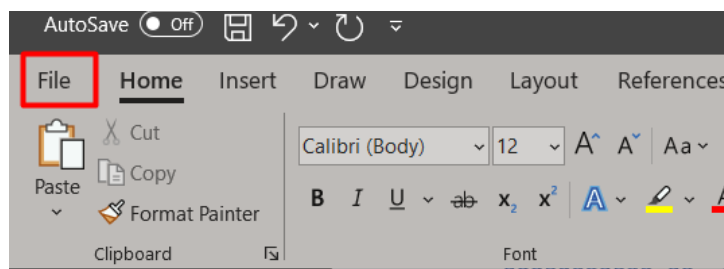


Obrázok 4 Obsah škodlivého dokumentu

Zapnutie Developer tabu

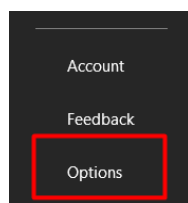
Pre vloženie VBA makra do dokumentu je potrebné **zapnúť tab s názvom Developer**. Ten zapneme nasledovne:

1. V ľavom hornom rohu obrazovky klikneme na **File**.



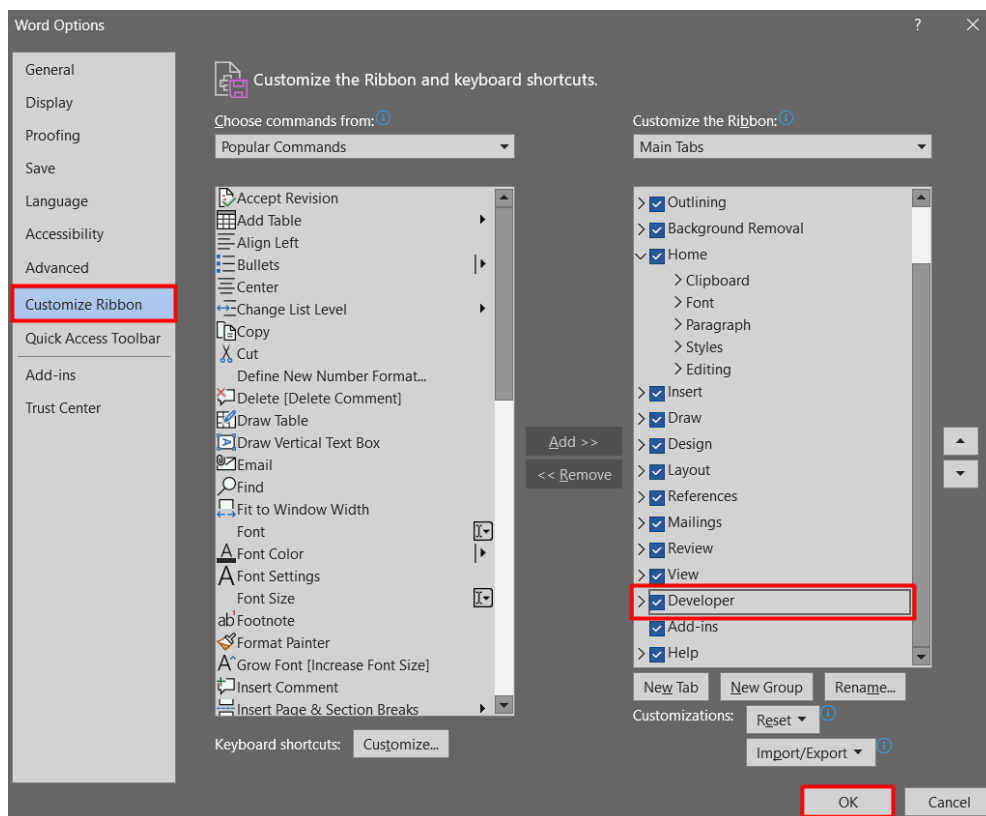
Obrázok 5 Zapnutie Developer tab Krok-1

2. V ľavom dolnom rohu obrazovky klikneme na **Options**.



Obrázok 6 Zapnutie Developer tab Krok-2

3. Ako posledné pod záložkou **Customize Ribbon** nájdeme a zaškrtneme **Developer**. Následne túto voľbu uložíme stlačením **OK**.

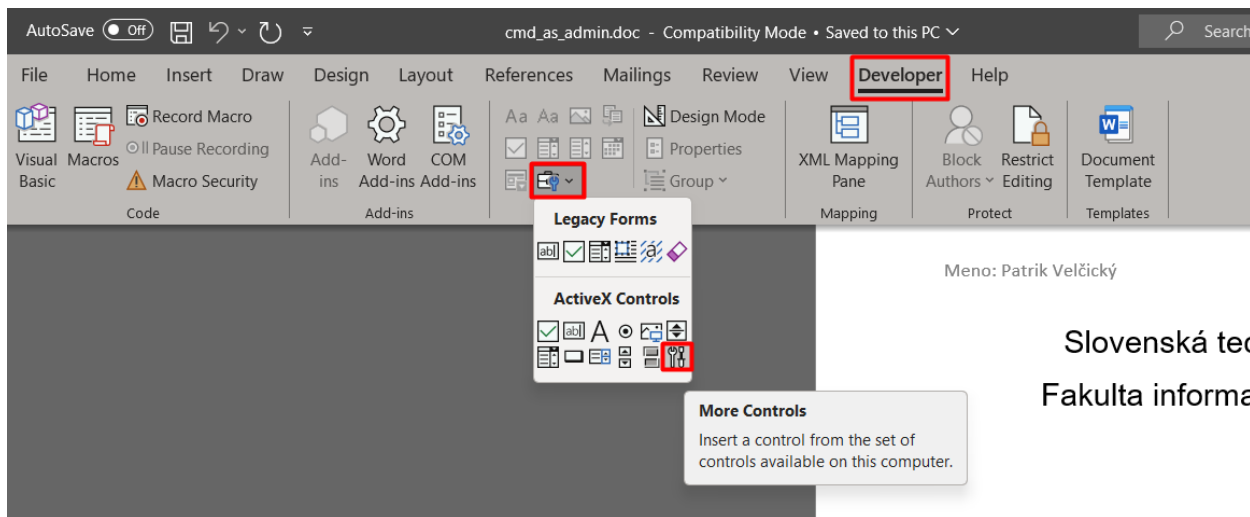


Obrázok 7 Zapnutie Developer tab Krok-3

Vytvorenie VBA makra

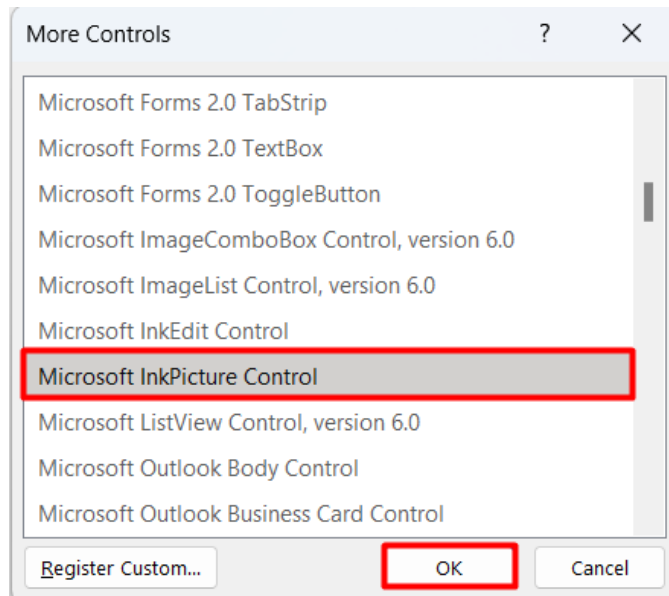
Spustenie VBA makra naviažem na funkciu **InkPicture1_Painted**, na to potrebujeme vložiť InkPicture do škodlivého dokumentu. InkPicture vložíme do dokumentu nasledovne:

1. V **Developer** tabe klikneme na **Legacy tools** a následne na **More Controls**.



Obrázok 8 Vloženie InkPicture Krok-1

2. Zo zoznamu vyberieme **Microsoft InkPicture Control** a klikneme **OK**.



Obrázok 9 Vloženie InkPicture Krok-2

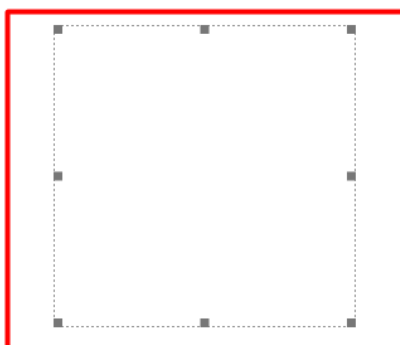
3. Následne **InkPicture** umiestnime do dokumentu. Odporúčam nemeniť veľkosť InkPicture, lebo po vykonaní skriptu sa zmení jeho veľkosť na pôvodnú. To môže pozmeniť cieľový škodlivý dokument.

Meno: Patrik Velčický

ID: 103175

BIT semestrálne zadanie

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií



Škodlivý dokument

Patrik Velčický

(BIT semestrálne zadanie)

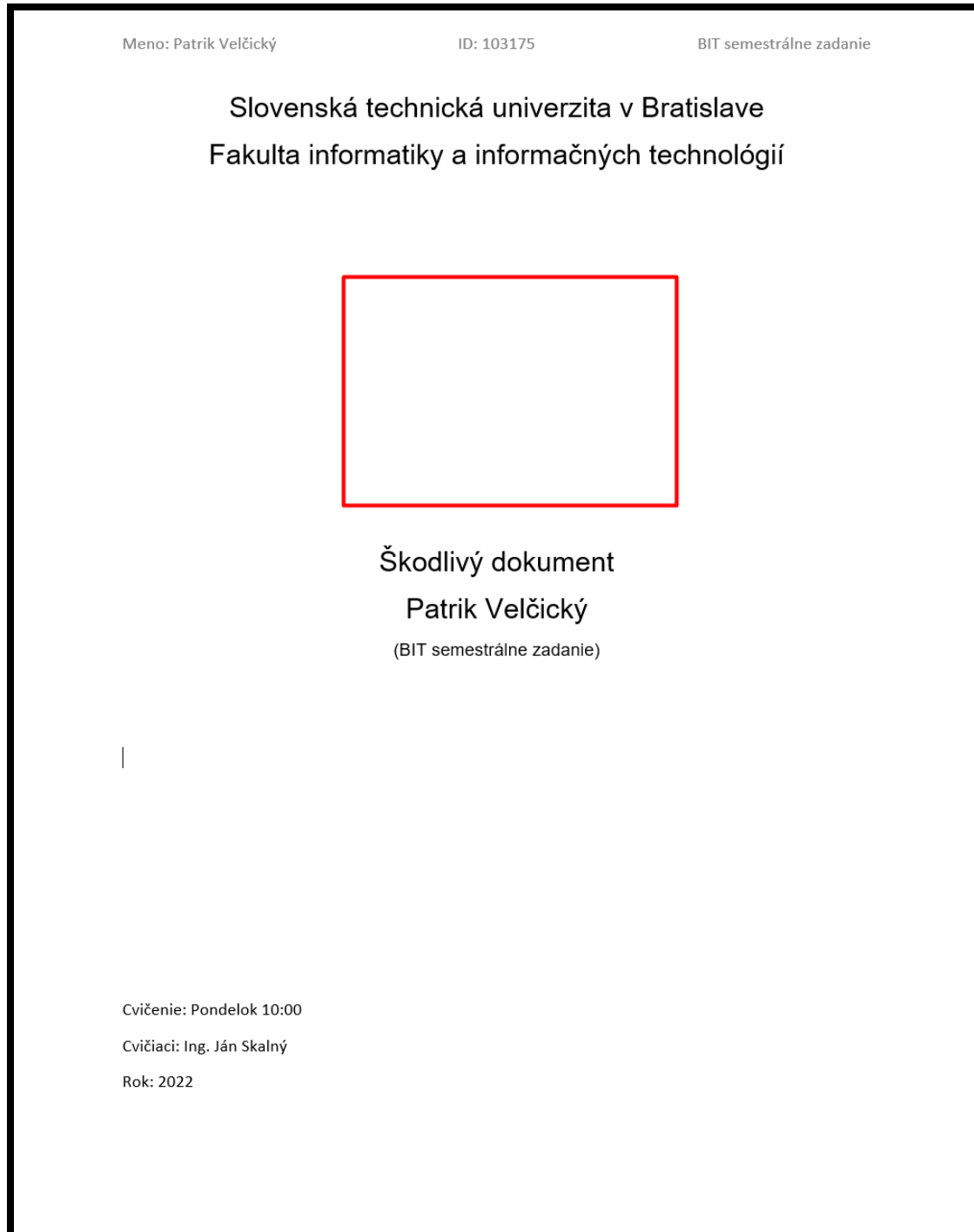
Cvičenie: Pondelok 10:00

Cvičiaci: Ing. Ján Skalný

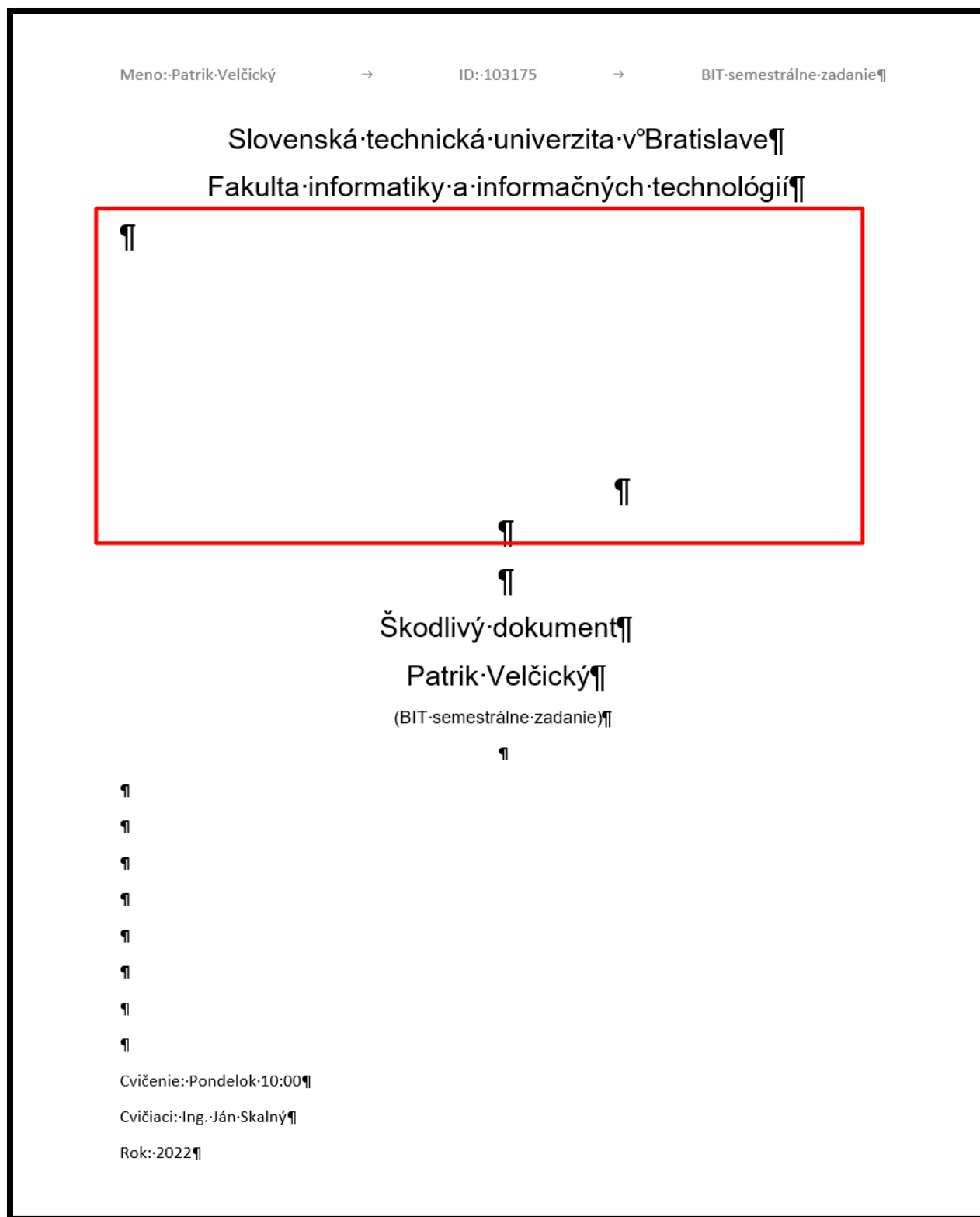
Rok: 2022

Obrázok 10 Vloženie InkPicture Krok-3

Na Obrázok 11 Vzhľad dokumentu po vložení InkPicture môžeme vidieť, že pôvodný vzhľad dokumentu nebol pozmenený z pohľadu „voľného oka“. Pri zapnutí Show/Hide zobrazenom na Obrázok 12 Vzhľad dokumentu po vložení InkPicture Show/Hide môžeme vidieť, že sa tam niečo nachádza (riadky nie sú odentrovane tak, ako by bežne boli).



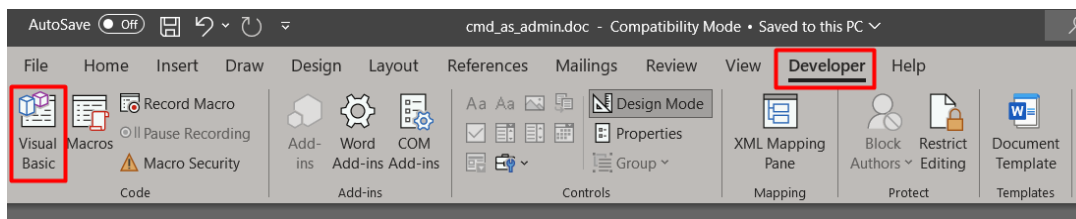
Obrázok 11 Vzhľad dokumentu po vložení InkPicture



Obrázok 12 Vzhľad dokumentu po vložení InkPicture Show/Hide

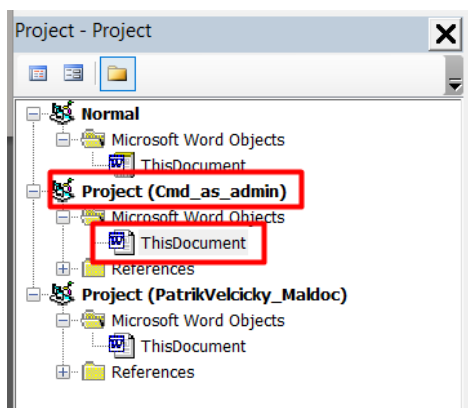
V nasledujúcich krokoch vložíme VBA makro do dokumentu.

4. Prepne sa do **Developer** tabu a na ľavej strane klikneme na **Visual Basic**.



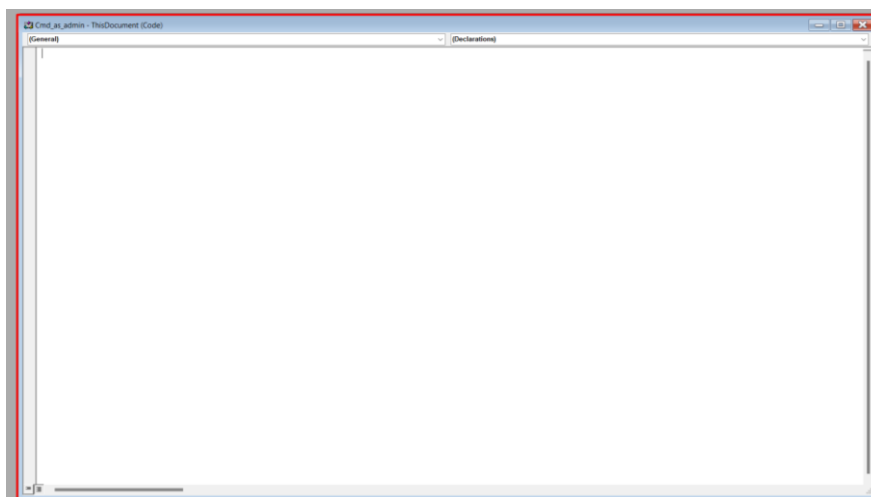
Obrázok 13 Vloženie VBA makra Krok-1

5. Otvorí sa nové okno. Na jeho ľavej strane nájdeme názov dokumentu (v mojom prípade Cmd_as_admin) a dvojklikom otvoríme **ThisDocument**.

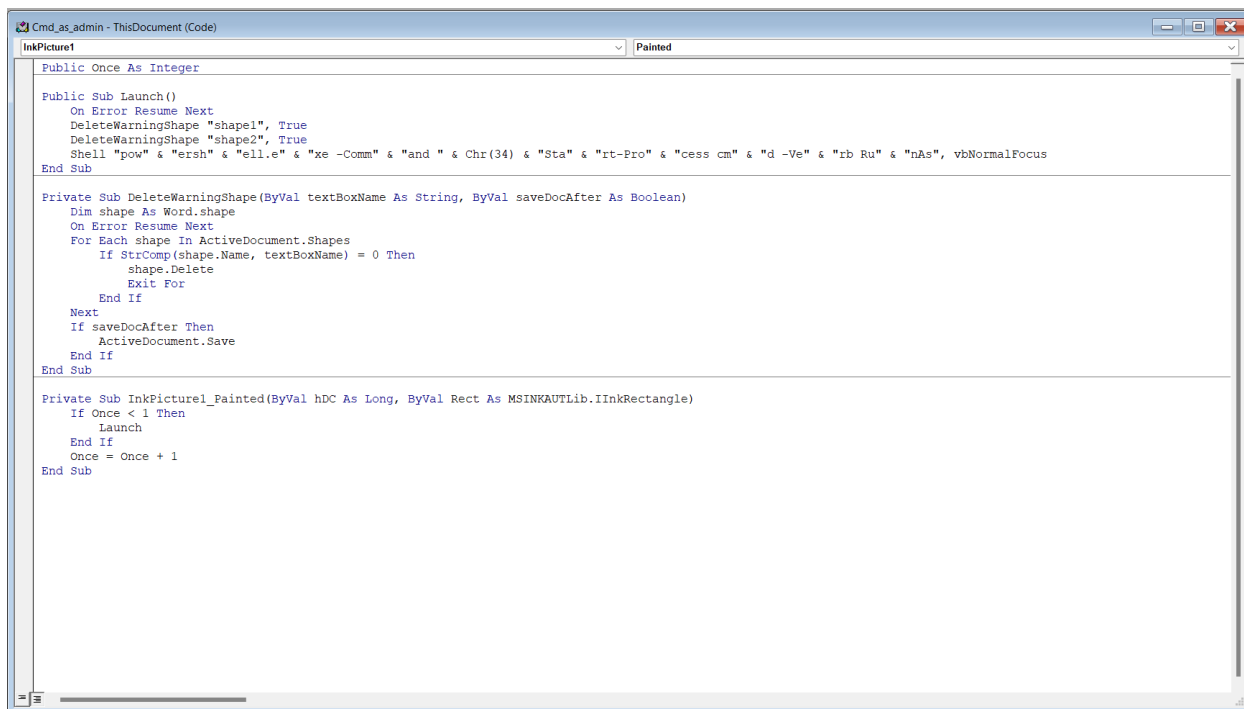


Obrázok 14 Vloženie VBA makra Krok-2

6. Následne sa otvorí okno, ktoré môžeme vidieť na Obrázok 15 Vloženie VBA makra Krok-3. Do tohto okna napíšeme alebo vložíme škodlivé VBA makro ako môžeme vidieť na Obrázok 16 Vložené VBA makro.



Obrázok 15 Vloženie VBA makra Krok-3

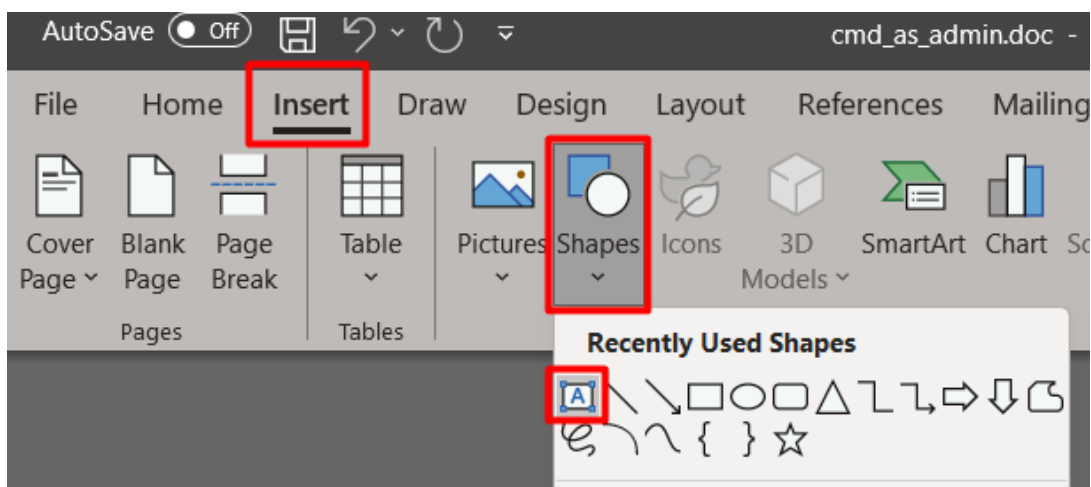


Obrázok 16 Vložené VBA makro

Sociálne inžinierstvo

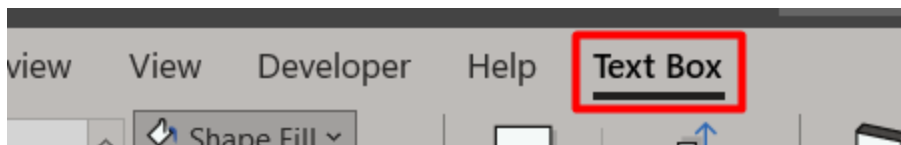
Následne je potrebné, aby používateľ po obdržaní takéhoto dokumentu zvolil **Enable Editing** a **Enable Content**, aby som toto dosiahol použijem sociálne inžinierstvo. Pomocou dvoch **TextBoxov**, jeden obsahujúci obrázok a druhý obsahujúci text, napodobní fiktívnu správu od Microsoft Office. Pre vytvorenie takejto správy je postup nasledovný:

1. Na tabe **Insert** klikneme na **Shapes** a následne vyberieme **TextBox**.



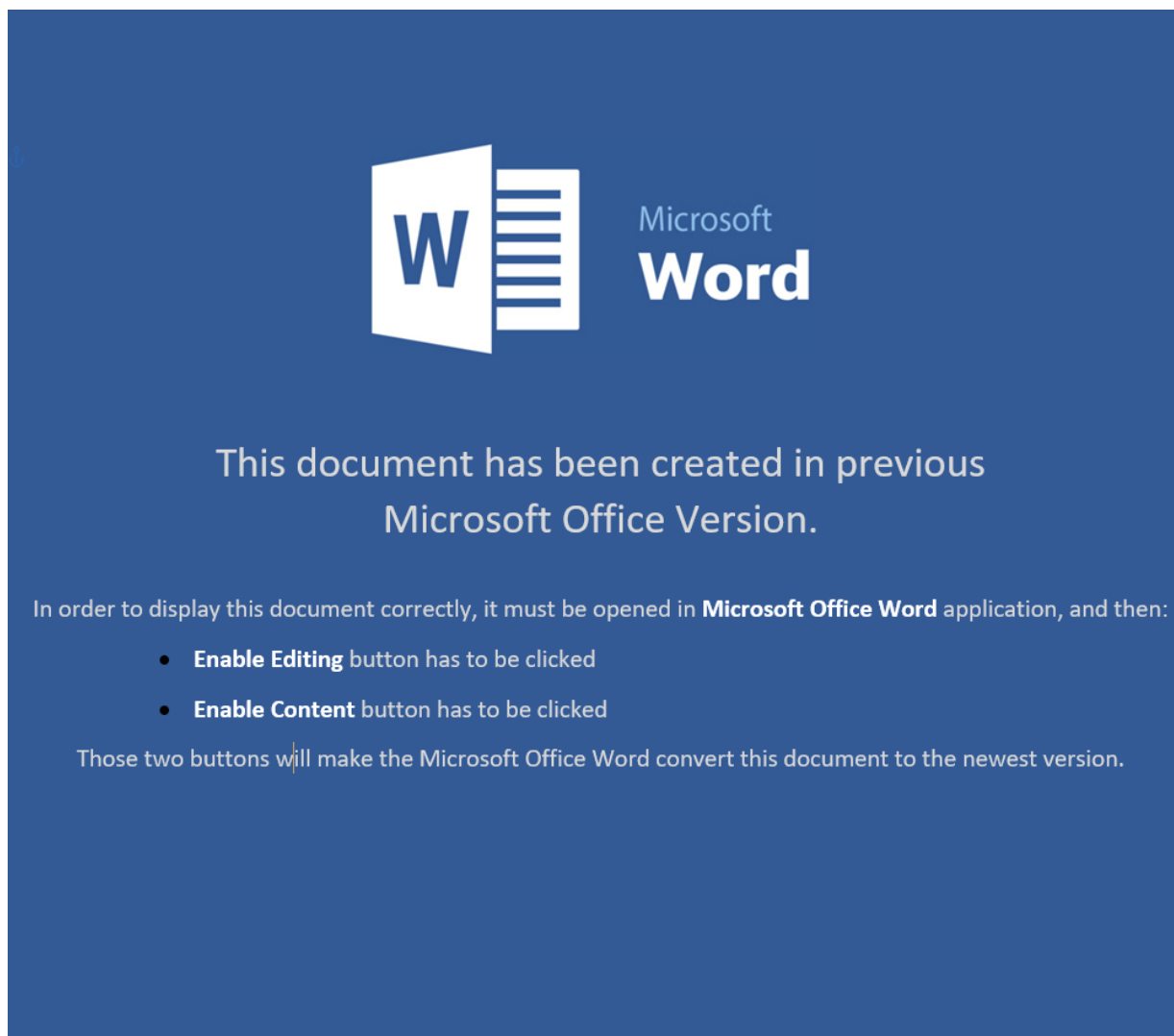
Obrázok 17 Vloženie TextBox

2. Po kliknutí na **TextBox** sa nám zobrazí tab s názvom **Text Box**. V tomto tabe vieme nastavovať rôzne vlastnosti TextBoxu. Ja som pomocou Shape Fill vložil obrázok do prvého TextBoxu a zmenil farbu druhého TextBoxu na mnou požadovanú pomocou Hex hodnoty. Následne som tieto TextBoxy umiestnil na obrazovku podľa potreby a do druhého TextBoxu napísal fiktívnu správu.



Obrázok 18 TextBox tab

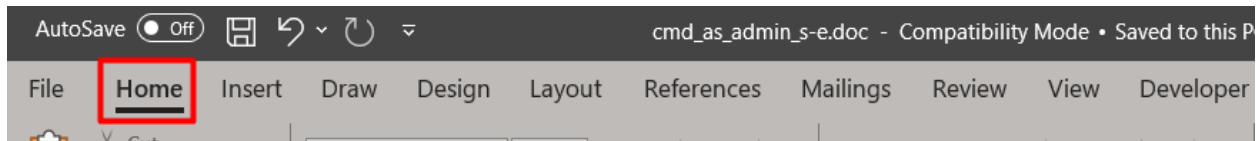
Finálny vzhľad dokumentu po aplikovaní sociálneho inžinierstva môžeme vidieť na Obrázok 19 Vzhľad dokumentu po vložení TextBox.



Obrázok 19 Vzhľad dokumentu po vložení TextBox

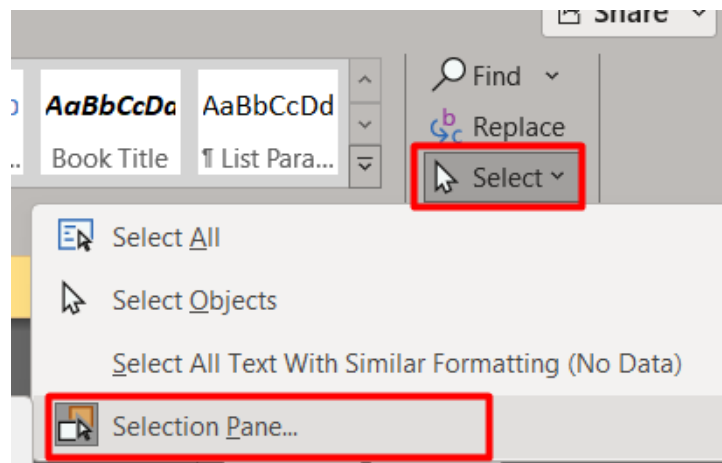
V rámci VBA skriptu je aj funkcia na odstránenie Shapeov, do tejto funkcie vstupujú dve premenné. Prvá premenná určuje názov Shapu, ktorý chceme vymazať a druhá premenná určuje, či sa dokument po vymazaní Shapu uloží alebo nie. Z tohto dôvodu potrebujeme zmeniť názvy Text Boxou na nami požadované. To urobíme nasledovne:

1. Prejdeme na tab **Home**.



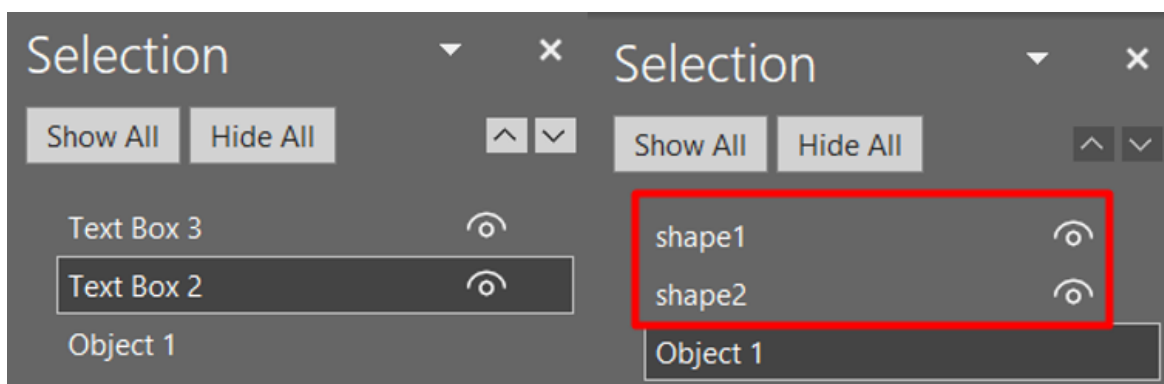
Obrázok 20 Premenovanie TextBox Krok-1

2. Na pravej strane klikneme na **Select** a následne klikneme na **Selection Pane**.



Obrázok 21 Premenovanie TextBox Krok-2

3. Na pravej strane obrazovky sa zobrazí **Selection** v mojom prípade sa tam nachádzajú tri objekty. Objekty s názvom Text Box premenujem dvojklikom na shape1 a shape2 (podľa VBA skriptu).



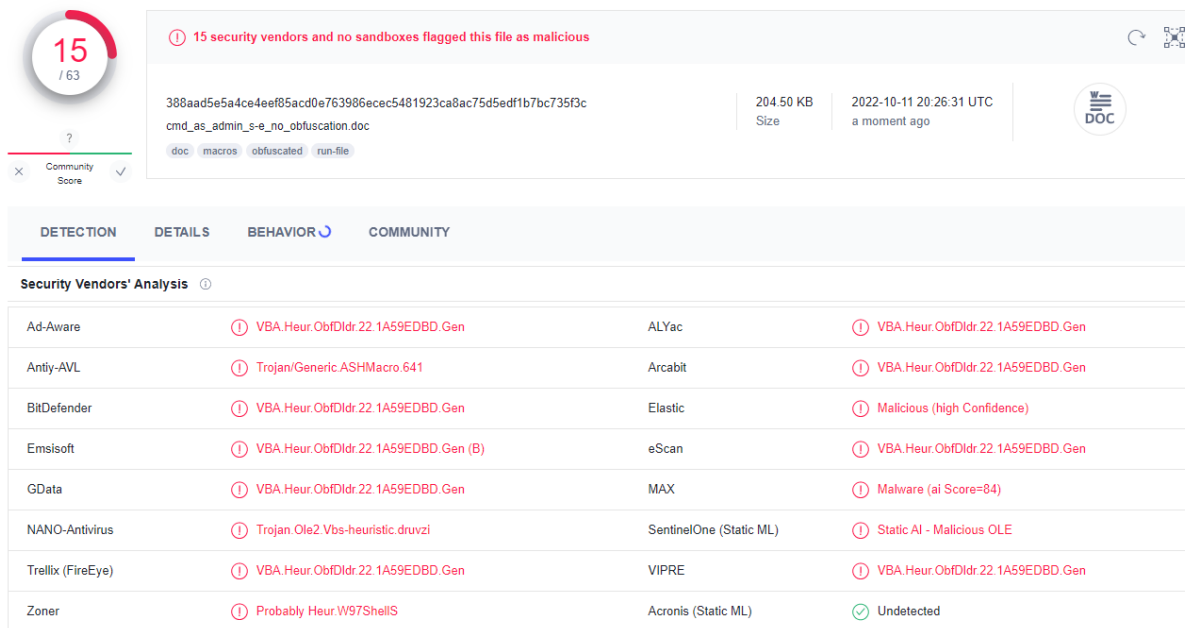
Obrázok 22 Premenovanie TextBox Krok-3

Obfuskácia VBA makra

Na obfuskáciu VBA makra som použil Python kód z githubu s názvom obfuscate.py (<https://github.com/mgeeky/VisualBasicObfuscator/blob/master/obfuscate.py>). Po obfuskácii VBA makra som znovu vložil tento VBA kód do dokumentu.

Sken škodlivého dokumentu pomocou VirusTotal

Aby som zistil detekciu škodlivého kódu pomocou AV použil som online nástroj VirusTotal (<https://www.virustotal.com/gui/home/upload>).



The screenshot shows the VirusTotal interface for a document file. The file is identified as 'cmd_as_admin_s-e_no_obfuscation.doc' with a size of 204.50 KB, uploaded on 2022-10-11 20:26:31 UTC. The file is categorized as 'doc', 'macros', 'obfuscated', and 'run-file'. The scan results show that 15 security vendors and no sandboxes flagged this file as malicious. The 'Security Vendors' Analysis table is as follows:

Vendor	Detection	Vendor	Detection
Ad-Aware	VBA.Heur.ObfDldr.22.1A59EDBD.Gen	ALYac	VBA.Heur.ObfDldr.22.1A59EDBD.Gen
Antiy-AVL	Trojan.Generic.ASHMacro.641	Arcabit	VBA.Heur.ObfDldr.22.1A59EDBD.Gen
BitDefender	VBA.Heur.ObfDldr.22.1A59EDBD.Gen	Elastic	Malicious (high Confidence)
Emsisoft	VBA.Heur.ObfDldr.22.1A59EDBD.Gen (B)	eScan	VBA.Heur.ObfDldr.22.1A59EDBD.Gen
GData	VBA.Heur.ObfDldr.22.1A59EDBD.Gen	MAX	Malware (ai Score=84)
NANO-Antivirus	Trojan.Ole2.Vbs-heuristic.druzzi	SentinelOne (Static ML)	Static AI - Malicious OLE
Trellix (FireEye)	VBA.Heur.ObfDldr.22.1A59EDBD.Gen	VIPRE	VBA.Heur.ObfDldr.22.1A59EDBD.Gen
Zoner	Probably Heur.W97ShellS	Acronis (Static ML)	Undetected

Obrázok 23 Sken dokumentu bez obfuskácie

The screenshot shows the VirusShare analysis interface for a document file. At the top, a red circle indicates a score of 26 out of 62. A warning message states: "26 security vendors and no sandboxes flagged this file as malicious". The file details include a long hash, a size of 54.50 KB, and a timestamp of 2022-10-13 19:05:40 UTC. The file is identified as "cmd_as_admin_s-e_obfuscation.doc" and has tags for "doc", "macros", "obfuscated", and "run-file". The "DETECTION" tab is active, showing a table of security vendors' analysis results.

Security Vendors' Analysis			
Acronis (Static ML)	ⓘ Suspicious	Ad-Aware	ⓘ VBA.Heur.Ursnif.2.1AFA4A54.Gen
ALYac	ⓘ VBA.Heur.Ursnif.2.1AFA4A54.Gen	Antiy-AVL	ⓘ Trojan/Generic.ASHMacro.654
Arcabit	ⓘ VBA.Heur.Ursnif.2.1AFA4A54.Gen	Avast	ⓘ SNH.Script [Dropper]
AVG	ⓘ SNH.Script [Dropper]	Avira (no cloud)	ⓘ HEUR/Macro.Downloader.MRABK.Gen
BitDefender	ⓘ VBA.Heur.Ursnif.2.1AFA4A54.Gen	ClamAV	ⓘ Doc.Malware.Chronos-6897935-0
Cynet	ⓘ Malicious (score: 99)	Cyren	ⓘ W97M/Agent.OJ.gen!Eldorado
Elastic	ⓘ Malicious (high Confidence)	Emsisoft	ⓘ VBA.Heur.Ursnif.2.1AFA4A54.Gen (B)
eScan	ⓘ VBA.Heur.Ursnif.2.1AFA4A54.Gen	ESET-NOD32	ⓘ VBA/TrojanDownloader.Agent.HEF
Fortinet	ⓘ VBA/Agent.HEFtr	GData	ⓘ VBA.Heur.Ursnif.2.1AFA4A54.Gen
Google	ⓘ Detected	Ikarus	ⓘ Trojan-Downloader.VBA.Agent
MAX	ⓘ Malware (ai Score=85)	Microsoft	ⓘ Trojan.Script/Wacatac.Blml
Sangfor Engine Zero	ⓘ VBA.Sus.Ofb	SentinelOne (Static ML)	ⓘ Static AI - Malicious OLE
Trellix (FireEye)	ⓘ VBA.Heur.Ursnif.2.1AFA4A54.Gen	VIPRE	ⓘ VBA.Heur.Ursnif.2.1AFA4A54.Gen

Obrázok 24 Sken dokumentu s obfuskáciou

Ako môžeme vidieť, tak počet detegovaní škodlivého dokumentu, ktorý obsahuje obfuskovaný VBA script je takmer dvojnásobne vyšší ako počet detegovaní dokumentu s neobfuskovaným VBA skriptom. Z týchto zistení mi vyplýva, že AV v súčasnosti ľahšie detegujú obfuskovaný VBA skript.

Bibliografia

1. cyborgsecurity. <https://www.cyborgsecurity.com/>. [Online] 16. November 2021. [https://www.cyborgsecurity.com/threats/emerging-threats/maldoc-execution-chain/#:~:text=Maldoc%20\(Malicious%20Documents\)%20are%20documents,document%20with%20embedded%20VBA%E2%80%8D%20macros%E2%80%8D..](https://www.cyborgsecurity.com/threats/emerging-threats/maldoc-execution-chain/#:~:text=Maldoc%20(Malicious%20Documents)%20are%20documents,document%20with%20embedded%20VBA%E2%80%8D%20macros%E2%80%8D..)
2. *Classifying Malicious Documents on the Basis of Plain-Text Features: Problem, Solution, and Experiences*. Jiwon Hong, Dongho Jeong, Sang-Wook Kim. Advances in Big Data and Machine Learning, Seoul 04763, Korea : Applied Sciences, 2022. 12084088.
3. Anderson, Ross J. *Security engineering: a guide to building dependable distributed systems*. Indianapolis : John Wiley & Sons, 2008. 978-0-470-06852-6.
4. Yadav, Neelam and Panda, Supriya P. Feature selection for email phishing detection using machine learning. *International Conference on Innovative Computing and Communications*. Singapore : Springer, 2022.
5. Lutkevich, Ben. techtarget. <https://www.techtarget.com/>. [Online] April 2021. <https://www.techtarget.com/searchsecurity/definition/obfuscation>.
6. *Malicious web content detection by machine learning*. Hou, Yung-Tsung and Chang, Yimeng and Chen, Tsuhan and Lai, Chi-Sung and Chen, Chia-Mei. s.l. : Elsevier, 2010, Zv. 37. 1.
7. *Attribution is in the object: Using RTF object dimensions to track APT phishing weaponizers*. Saad, Ghareeb and Raggi, Michael A. London : Virus Bull, 2020, Zv. 12.
8. Yang, Shaojie and Chen, Wenbo and Li, Shanxi and Xu, Qingxiang. Approach using transforming structural data into image for detection of malicious MS-DOC files based on deep learning models. *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. s.l. : IEEE, 2019.
9. Smutz, Charles and Stavrou, Angelos. Malicious PDF detection using metadata and structural features. *Proceedings of the 28th annual computer security applications conference*. 2012.
10. Smith, Ryan. bestestredteam. <https://bestestredteam.com/>. [Online] 19. Marec 2019. <https://bestestredteam.com/2019/03/19/7-tools-for-malicious-document-creation/>.
11. Todd, Adam. trustedsec. <https://www.trustedsec.com/>. [Online] 3. Marec 2020. <https://www.trustedsec.com/blog/intro-to-macros-and-vba-for-script-kiddies/>.
12. —. trustedsec. <https://www.trustedsec.com/>. [Online] 4. August 2020. https://www.trustedsec.com/blog/malicious-macros-for-script-kiddies/#_ftn3.