Fakulta informačních technologií
Vysoké učení technické v Brně

Přenos dat, počítačové sítě a protokoly

L2 MitM

23. dubna 2017 Daniel Klimaj

Obsah

1 Man-in-the-Middle

Man-in-the-Middle (MitM) je jeden z najčastejších typov útokov na počítačovú sieť. Podstatou útoku je aktívne odpočúvanie komunikácie medzi vybranými obeťami. MitM funguje tak, že útočník vytvorí spojenie medzi sebou a zvolenými obeťami. Obete sú stále v ilúzii, že komunikujú priamo, ale v skutočnosti ich komunikácia prechádza cez zariadenie útočníka. V konečnom dôsledku môže útočník zachytávať citlivé dáta alebo vkladať aebo upravovať dátové prúdy[2]. Tento dokument sa bude ďalej venovať MitM útokom ARP Cache Poisoning a Neighbor Discovery Cache Poisoning (NDP).

1.1 ARP Chache Poisoning

ARP (Address Resolution Protocol) protokol je protokol pracujúci na linkovej vrstve a slúži pre mapovanie sieťových adries (IPv4) na fyzické adresy zariadení (MAC). Toto sa deje pomocou dvochy typov ARP správ - ARP Request a ARP Reply. Zariadenie, ktoré zisťuje vlastníka IP adresy zašle broadcastom správu ARP Request ďalej obsahujúcu aj svoju IP a MAC adresu. Cieľové zariadenie (ak sa v sieti nachádza) odpovie na túto správu pomocou ARP Reply, v ktorej uvedie svoju MAC adresu[2].

Vzhľadom na to, že ARP protokol je nutný pre správne fungovanie IPv4 sietí, musí prijímať všetky správy kedykoľvek. Toto umožňuje útok ARP Cache Poisoning. Útok spočíva v tom, že útočník pošle obeti ARP Reply aj keď žiaden ARP Request nebol vyslaný. Vyslaná správa ARP Reply spôsobí aktualizáciu ARP Cache u obete. V prípade ak, útočníkova správa spôsobila zmenu MAC adresy nejakého zariadenia Z v sieti na útočníkovu MAC adresu, budú správy obete posielané pre zariadenie Z posielané útočníkovi.

1.2 Neighbor Discovery Cache Poisoning

ARP protokol bol vIPv6 nahradený Neighbor Discovery. Rozlišovanie adries pracuje pomocou dvoch ICMPv6 správ Neighbor Solicitation (typ 135) a Neighbor Advertisement (typ 136). Podobne ako u ARP, ani NS a NA výmena nie je vôbec zabezpečená, a tak neexistujú žiadne protiopatrenia proti zasielaniu nevyžiadaných NA správ[1].

1.3 Postup útoku

ARP Chace Poisoning a Neighbor Discovery Cache Poisoning spravidla prebieha v nasledujúcich krokoch[3]:

1. Útočník zmapuje segment siete a vyhľadá potenciálne obete.
2. Útočník otrávi ARP/ND chache obetí - obeti 1 pošle ARP Reply alebo NA, ktorá nahradí MAC adresu obete 2 útočníkovou MAC adresou a obrátene u obete 2. Toto sa periodicky opakuje.
3. Útočník zachytáva komunikáciu obetí a u obetí udržuje ilúziu priamej komunikácie, tým že zachytenú komunikáciu preposiela pôvodne myslenému cieľ.

2 Implementácia MitM útoku

Táto kapitola sa bude venovať popisu implementácie MitM útoku ARP Cahce Poisoning a Neighbor Discovery Cache Poisoning.

2.1 Programy

Výsledkom implementácie sú tri spustiteľné programy, z ktorých každý vykonáva jeden z krokov popísaných v časti 1.3.

- pds-scanner - mapovanie siete
- pds-spoof - periodické zasielanie ARP Reply a NA správ obetiam
- pds-intercept - zachyt8vanie komunikácie obetí

2.1.1 pds-scanner

Program pds-scanner slúži na zistenie aktívnych IPv4 a IPv6 zariadení v lokálnej sieti. Získané adresy sa zapíšu do určeného XML súboru.

Vyhľadávanie IPv4 prebieha nasledovným spôsobom:

1. Získa sa lokálna IPv4 adresa útočníka.
2. Z lokálnej adresy sa na základe masky podsiete zistí celkový rozsah adries.
3. Vytvorí sa std::vector obsahujúci všetky možné adresy v podsieti.
4. Vyfiltruje sa adresa podsiete, routeru abroadcastová adresa.
5. Na všetky adresy sa postupne pošle ARP Request. Tento bod sa opakuje 2x.
6. Zanalyzujú sa prijaté ARP Reply a vytvorí sa zoznam hostov.

Po skončení vyhľadávania IPv4 hostov nasleduje skenovanie Link-Local IPv6 adries a následne Global-scope IPv6 adries. IPv6 adresy sa vyhľadávajú len pre MAC adresy získané pri skenovaní IPv4 adries, ak v tom kroku nebol nájdený žiaden host, nasledujúce kroky sa preskočia. Skenovanie IPv6 pozostáva z:

1. Pošle sa Echo Ping na multicast FF02::1.
2. Zanalyzujú sa prijaté Echo Reply, Neighbor Solicitation a Neighbor Adveritisement pakety.
3. Ak sa MAC adresa odosielateľa zhoduje s MAC adresou v zozname hostov, doplní sa adresa.

Použitie: `pds-scanner -i interface -f file`

- `-i interface` je názov sieťového rozhrania
- `-f file` je cesta ku XML súboru, do ktorého sa zapíše výstup

2.1.2 pds-spoof

Program pds-spoof je najjednoduchší spomedzi implementovaných programov. Na základe zadaných parametrov buď periodicky posiela falošné ARP Reply alebo Neighbor Advertisement pakety dvom zadaným obetiam. Po skončení zašle ARP Reply alebo NA obnovujúci počiatočný stav cache.

Použitie: `pds-spoof -i interface -t sec -p protocol -victim1ip ip -victim1mac mac -victim2ip ip -victim2mac mac`

- `-i interface` je názov sieťového rozhrania
- `-t sec` je interval v milisekundách, v ktorom sa zasielajú pakety
- `-p protocol` je názov protokolu, platné hodnoty sú len *arp* a *ndp*
- `-victimNip ip` je IP adresa obete N, kde $N = \{1, 2\}$, IP adresa musí odpovedať zvolenému protokolu
- `-victimNmac mac` je MAC adresa obete N, kde $N = \{1, 2\}$

2.1.3 pds-intercept

Program pds-intercept zachytáva komunikáciu medzi vybranými obeťami, ktoré sú načítané z určeného XML súboru. Zo XML súboru sa postupne parsujú elementy a hodnoty pre každého hosta sa ukladajú do std::vector¡std::string¿ vo formáte `nazov@obsah`, kde nazov je názov elementu alebo atribútu a obsah je text elementu alebo hodnota atribútu (napr. ipv4@192.168.100.5). Zo získaných vektorov sa potom vytvorí HostGroup (viď. 2.2.5). Samotný interceptor pracuje nasledovne:

1. Zachytáva všetky pakety a kontroluje cieľovú MAC adresu a protokol. Všetky pakety neobsahujúce útočníkovu MAC adresu ako cieľ alebo pakety pre iný protokol ako IPv4 alebo IPv6 preskakuje.
2. Z platného paketu získa zdrojovú a cieľovú IP adresua zdrojovú MAC adresu.
3. Získané adresy porovná so skupinami HostGroup.
4. Ak nájde zhodu, upraví MAC adresy v Ethernetovej hlavičke a paket prepošle.

Použitie: `pds-intercept -i interface -f file`

- `-i interface` je názov sieťového rozhrania
- `-f file` je cesta ku XML súboru, do ktorého sa zapíše výstup

2.2 Triedy

Programy z časti 2.1 využívajú triedy implementujúce rôzne funkcionality:

2.2.1 IPAddr, IPv4Addr, IPv6Addr

Trieda IPAddr slúži ako základ pre IPv4Addr a IPv6Addr, ktoré reprezentujú IP adresy IPv4 a IPv6. Ich inštancie sa vytvárajú načítaním zo zvoleného sieťového rozhrania, z textového reťazca (IPv4 v dotted-decimal formáte a IPv6 vo formáte popísanom v RFC5952) alebo z bytového prúdu. Umožňujú manipuláciu s IP adresami - rozdelenie na časti (oktety, bloky), prevod do bytového prúdu a porovnávanie na úrovni bytov.

2.2.2 MACAddr

MACAddr predstavuje abstrakciu MAC adresy. Podobne ako IPAddr aj MACAddr umožňuje vytvorenie inštancie načítaním zo sieťového rozhrania, z reťazca (vo formáte xx:xx:xx:xx:xx:xx alebo xxxx.xxxx.xxxx) alebo z bytov oktetov.

2.2.3 NetItf

NetItf predstavuje lokálne sieťové rozhranie. Hlavným účelom tejto triedy je získať index vybraného sieťového rozhrania a lokálne adresy - MAC, IPv4 a všetky IPv6.

2.2.4 Packet, ArpPkt, IcmpV6Pkt

Packet je abstraktná trieda poskytujúca základ pre triedy ArpPkt a IcmpV6Pkt. Tento spoločný základ je potrebný pre jednotnú prácu uvedených potomkov so socketmi. Umožňuje zostavenie Ethernetovej hlavičky, nastavenie zdrojovej a cieľovej MAC adresy a poskytuje abstraktné rozhranie pre serializáciu triedy na tok bytov a získanie veľkosti paketu.

ArpPkt predstavuje paket protokolu ARP. Trieda je potomkom abstraktnej triedy Packet. Jej verejné rozhranie tvoria metódy pre nastavenie zdrojovej a cieľovej IPv4 adresy a metóda pre analýzu ARP Reply, z ktorej získa zdrojovú MAC a IPv4 adresu.

Icmpv6Pkt predstavuje ICMPv6 hlavičku. Podobne ako ArpPkt jej verejné rozhranie tvoria metódy pre nastavenie zdrojovej a cieľovej IPv6 adresy alebo príznakov ICMPv6 typu 136 (Neighbor Advertisement). Zaujímavou súkromnou metódou je checksum. Checksum počíta 16-bitový kontrolný súčet IPv6 pseudo hlavičky (zdrojová a cieľová IPv6 adresa, veľkosť ICMPv6 hlavičky a kód nasledujúcej hlavičky (ICMPv6 = 58)) a ICMPv6 hlavičky. Priebežný súčet sa ukladá do 32-bitovej premennej, ktorá sa na záver "oreže"na 16b, spraví sa prenos na LSB a hodnota sa zinvertuje.

2.2.5 Pomocné triedy

Okrem vyššie popísaných tried bolo implementovaných niekoľko pomocných tried:

- Hash - implementácia asociat9vneho poľa nas std::map
- HostGroup - trieda pre uchovávanie informácií o pároch obetí
- Socket - zjednodušené rozhranie nad socket, sendto, recvfrom a setsockopt
- Types - pomocné metódy

3 Testovanie

Testovanie prebiehalo na 3 virtuálnych strojoch *isa2015* v bridged móde, ich parametre boli:

Attacker

MAC	08:00:27:92:43:65
IPv4	10.190.8.38
IPv6 LL	FE80::A00:27FF:FE92:4365
IPv6 G	2001:67C:1220:C1A0:B4E1:7544:4291:9F21

Victim 1

MAC	08:00:27:76:DE:10
IPv4	10.190.8.76
IPv6 LL	FE80::A00:27FF:FE76:DE10
IPv6 G	2001:67C:1220:C1A0:ED4E:80B7:D968:27F

Victim 2

MAC	08:00:27:17:A1:9F
IPv4	10.190.11.57
IPv6 LL	FE80::A00:27FF:FE17:A19F
IPv6 G	2001:67C:1220:C1A0:741C:D41F:51B1:F49D

Priepustnosť siete bola testovaná pomocou utility ping s paketmi o veľkosti 50008 a 1408 bytov, výsledky sú v nasledujúcej tabuľke (stĺpce s Itc v hlavičke značia hodnoty, kedy dochádzalo k MitM, hodnoty sú v Mb/s):

50K	1408	50K Itc	1408 Itc
140,177	30,609	0,882	0,828
71,034	31,909	0,909	0,989
119,422	27,608	0,849	0,859
77,307	68,893	0,871	0,833
104,048	35,421	0,887	0,793

Z tabuľky možno vyvodiť, že pri zachytávaní komunikácie medzi obeťami dochádza k výraznému spomaleniu ich vzájomnej komunikácie.

Literatúra

[1] Packetlife: *IPv6 neighbor spoofing.* [online]. 2009, [cit. 23.4.2017].
URL <http://packetlife.net/blog/2009/feb/2/ipv6-neighbor-spoofing/>

[2] Techgenix: *Understanding Man-in-the-Middle Attacks – ARP Cache Poisoning.* [online]. 2010, [cit. 23.4.2017].
URL <http://techgenix.com/understanding-man-in-the-middle-attacks-arp-part1/>

[3] Veracode: *ARP Spoofing.* [online]. 2017, [cit. 23.4.2017].
URL <https://www.veracode.com/security/arp-spoofing>