VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DOKUMENTACE K PROJEKTU DO PŘEDMĚTU PDS

Man in the Middle

Bc. Lukáš Pelánek, xpelan03@stud.fit.vutbr.cz

Obsah

Útoky Man in the Middle

Jak je z názvu patrné, při tomto druhu útoku útočník stojí mezi dvěma stranami, které se domnívají, že komunikují přímo mezi sebou. Mezitím celá konverzace je útočníkem monitorována a ovládána. Pro úspěšný útok je nezbytné, aby útočník vstoupil do komunikace mezi oběma stranami.

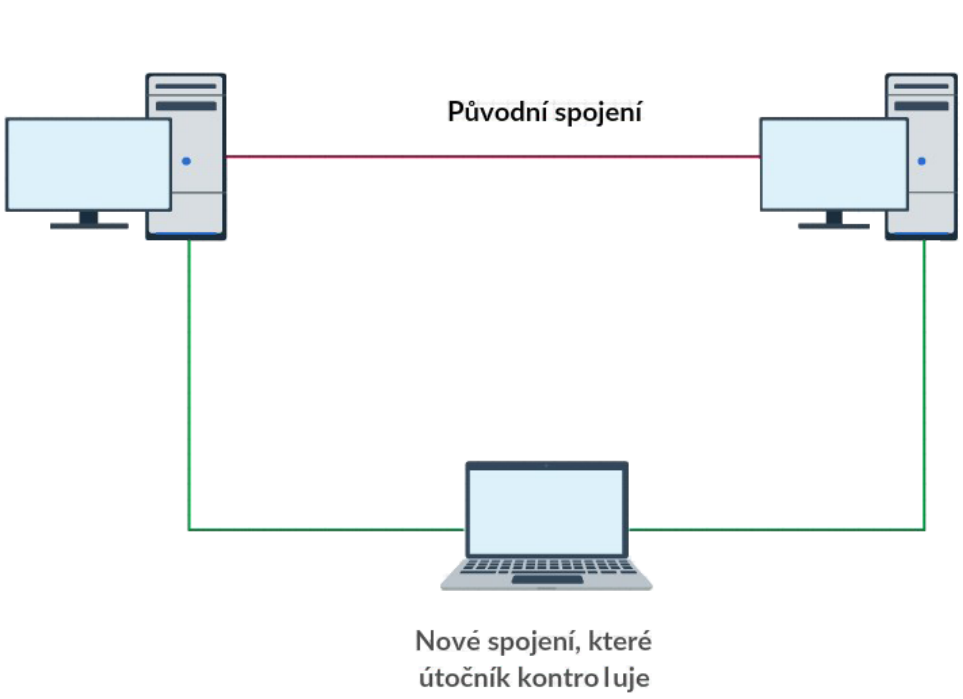
Většina moderních šifrovaných protokolů již poskytuje mechanismus pro ověření na koncové straně právě proto, aby zabránily útoku Man in the Middle. Například protokol Secure Sockets Layer (SSL) je používán pro ověření jedné nebo obou stran [1].

Princip útoku

Útok spočívá v otrávení cache, která obsahuje IP adresy a příslušné linkové adresy lokálních zařízení. Pro protokol IPv4 se používá ARP cache a pro protokol ipv6 se používá NDP cache. Útočník v první fázi musí pomocí protokolů ARP [2] a ICMP [3] oskenovat lokální síť. Počet lokálních zařízení v síti s IPv4 není vysoký, a tak útočník může poslat ARP dotaz na všechny možné adresy v síti. Pro protokol IPv6 je počet zařízení příliš vysoký, a tak pro zjistění zařízení v síti je vhodné použít zprávu ping request na speciální multicast adresu a zařízení v síti sama odpoví.

V druhé fázi útočník odešle zprávu, že IP adresa vybraných zařízení náleží útočníkovi. Obě strany si tuto informaci uloží do cache a útočník se stane prostředníkem mezi oběma stranami.

Ve třetí fázi musí útočník veškerou komunikaci, která je směřována mezi zařízeními dále přeposílat, či modifikovat, aby napadená zařízení nepoznala, že se staly oběťmi útoku Man in the Middle. Útočník v tuto chvíli získává plnou kontrolu nad komunikací mezi napadenými zařízeními.



Obrázek 1: Útok Man in the Middle

Implementace

Aplikace je implementována v jazyce C++ a je přeložitelná na platformě Linux. Řešení zahrnuje několik samostatných aplikací, jejichž implementace bude rozebrána v této kapitole. Každá aplikace pro síťovou komunikaci pracuje se sockety. Při implementaci byl kladen důraz na objektově orientovaný model. Návod pro obsluhu jednotlivých aplikací je přiložen v souboru readme.txt. Každá aplikace se korektně ukončí při příchodu signálu SIGINT.

Scanner

Tato aplikace slouží ke skenování lokální sítě. Aplikace nejprve pomocí protokolu ARP rozešle dotaz na adresu všech možných zařízení v síti. Následně aplikace vyčkává 2,5 vteřiny na odpovědi. Všechny odpovědi zpracuje a nalezená zařízení si uloží do vectoru. Následně pomocí protokolu ICMPv6 provede skenování ipv6 adres. Na multicast adresu **ff02::1**, jenž zahrnuje všechny uzly v síti, odešle ping request. Následně 2,5 vteřiny vyčká a všechny odpovědi zpracuje a adresy vloží do vectoru. Nakonec aplikace vygeneruje výchozí XML soubor, který obsahuje všechna nalezená zařízení včetně jejich adres.

Spoofer

Aplikace jako vstupní parametr obdrž protokol, na němž bude probíhat podvrhování adres. Současně i linkovou a síťovou adresu vybraných zařízení. V případě protokolu ARP aplikace odešle ARP odpověď, ve které stojí, že IP adresa obětí má stejnou linkovou adresu jako útočník. Pro podvrhnutí IPv6 provozu aplikace odešle Neighbor Advertisement zprávu, ve které opět stojí, že daná IPV6 adresa náleží linkové adrese útočníka. Tyto zprávy jsou periodicky odesílány v časovém intervalu, který je rovněž specifikován jako vstupní parametr.

Intercepter

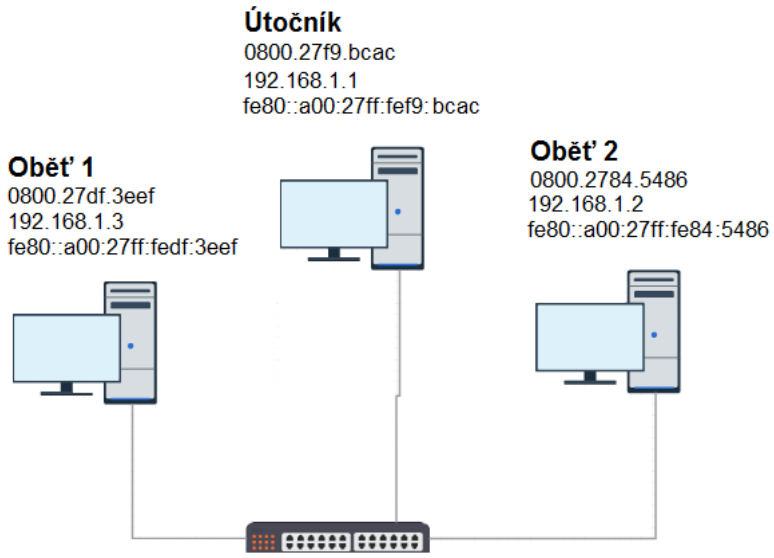
Vstupem aplikace je XML soubor, jenž obsahuje spárovaná zařízení, mezi kterýma má přeposílání paketů probíhat. Po zpracování XML souboru si aplikace uloží dvojice do paměti a následně vyčkává v nekonečné smyčce a přijímá pakety. Pokud daný paket náleží nějaké dvojici zařízení, tak cílová linková adresa paketu je přepsána na druhé zařízení v dané dvojici. Tím je zajištěno, že komunikace mezi oběťmi útoku je funkční.

Chooser

Implementoval jsem bonusovou aplikaci, která načte XML soubor, jenž je výstupem Scanneru. Následně vypíše seznam zařízení do terminálu včetně příslušných indexů. Uživatel následně interaktivně páruje zařízení a buď příkazem end a nebo signálem SIGINT ukončí aplikaci. Aplikace při ukončení vygeneruje obohacený XML soubor, jenž obsahuje spárované dvojice. Tento soubor je vstupem aplikace Intercepter.

Testování aplikace

Aplikace byla testována ve virtuální síti, která obsahovala 3 stanice. Jedna stanice prováděla útok a další dvě stanice byly oběťmi útoku. Topologie sítě a adresy zařízení jsou zobrazeny na následujícím obrázku.



Obrázek 2: Topologie virtuální sítě

Skenování sítě

Nejprve provedu skenování lokální sítě pomocí scanneru. Aplikace odešle ARP a ICMP zprávy všem stanicím v síti.

```
3 0.000079000 CadmusCo_f9:bc:ac     Broadcast           ARP          42 Who has 192.168.1.4?  Tell 192.168.1.1
4 0.000102000 CadmusCo_f9:bc:ac     Broadcast           ARP          42 Who has 192.168.1.5?  Tell 192.168.1.1
5 0.000120000 CadmusCo_f9:bc:ac     Broadcast           ARP          42 Who has 192.168.1.6?  Tell 192.168.1.1
6 0.000399000 CadmusCo_df:3e:ef     CadmusCo_f9:bc:ac   ARP          60 192.168.1.3 is at 08:00:27:df:3e:ef
7 0.000402000 CadmusCo_84:54:86     CadmusCo_f9:bc:ac   ARP          60 192.168.1.2 is at 08:00:27:84:54:86
8 0.000498000 CadmusCo_f9:bc:ac     Broadcast           ARP          42 Who has 192.168.1.7?  Tell 192.168.1.1
```

Obrázek 3: Skenování sítě pomocí ARP

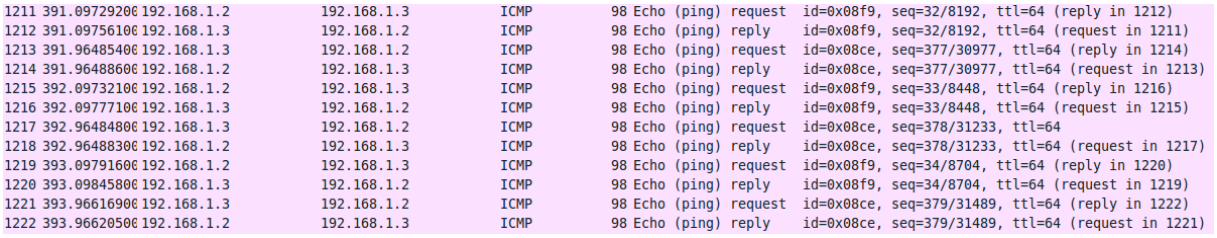
```
1066 1202.7813610 fe80::a00:27ff:fef9:bc ff02::1             ICMPv6       73 Echo (ping) request id=0x0045, seq=0, hop limit=255
1067 1202.7821010 fe80::a00:27ff:fedf:3e fe80::a00:27ff:fef9:bc ICMPv6    73 Echo (ping) reply id=0x0045, seq=0, hop limit=64
1068 1202.7821506 fe80::a00:27ff:fe84:54 fe80::a00:27ff:fef9:bc ICMPv6    73 Echo (ping) reply id=0x0045, seq=0, hop limit=64
```

Obrázek 4: Skenování sítě pomocí ICMPv6

Skenování bylo úspěšné a útočník objevil všechna zařízení v síti.

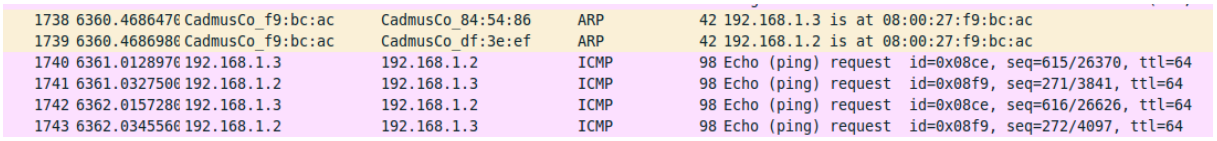
Otrávení cache obětí

V další fázi je nutné, aby došlo k otrávení cache na základě použitého protokolu. Mezi oběťmi probíhá komunikace. Cílem útočníka je otrávit cache oběti tak, aby pakety pro druhou oběť směřovaly k útočníkovi.



Obrázek 5: Komunikace mezi oběťmi

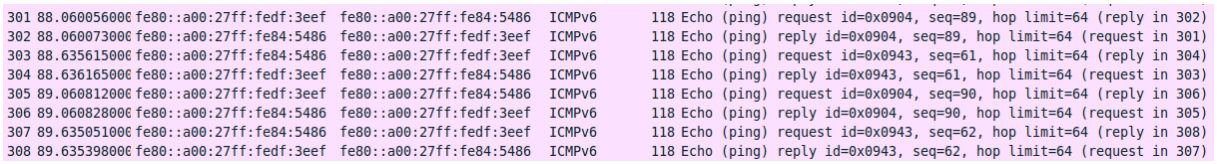
Stanice spolu komunikují pomocí protokolu ICMP (Obr. 5). Následně útočník odešle podvržené ARP odpovědi (Obr. 6), aby oběti aktualizovaly svoje cache. Následně veškerá komunikace je směrována k útočníkovi.



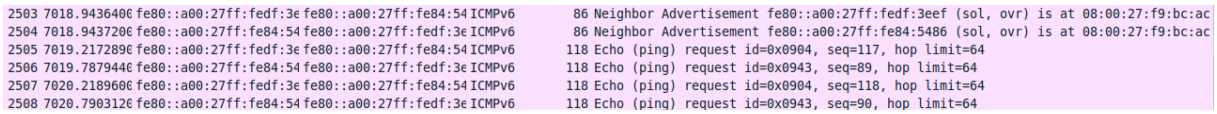
Obrázek 6: Odeslání podvržených odpovědí a přesměrování komunikace z pohledu útočníka

Útočník přijímá pouze zprávy ping request, protože útočník paket nepřepošle a oběť tedy nemůže odpovědět.

V případě protokolu IPv6 je postup totožný. Místo paketu ARP se odešle ICMPv6 Neighbor Advertisement zpráva s linkovou a síťovou adresou.



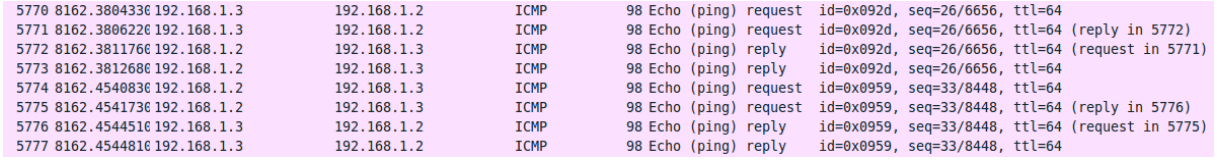
Obrázek 7: IPv6 Komunikace mezi oběťmi



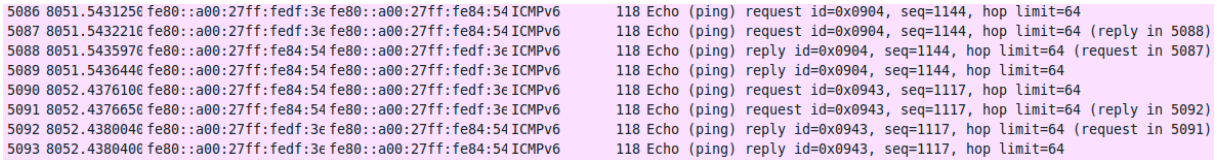
Obrázek 8: Odeslání podvržených odpovědí a přesměrování IPv6 komunikace z pohledu útočníka

Přeposílání provozu

Jako poslední krok je potřeba zajistit přeposílání provozu mezi oběťmi. Útočník spustí aplikaci pro přesměrování příchozího provozu. Příchozí pakety jsou zpracovány a přeposlány správným stranám (Obr. 8).



Obrázek 9: Přeposílání komunikace ze strany útočníka



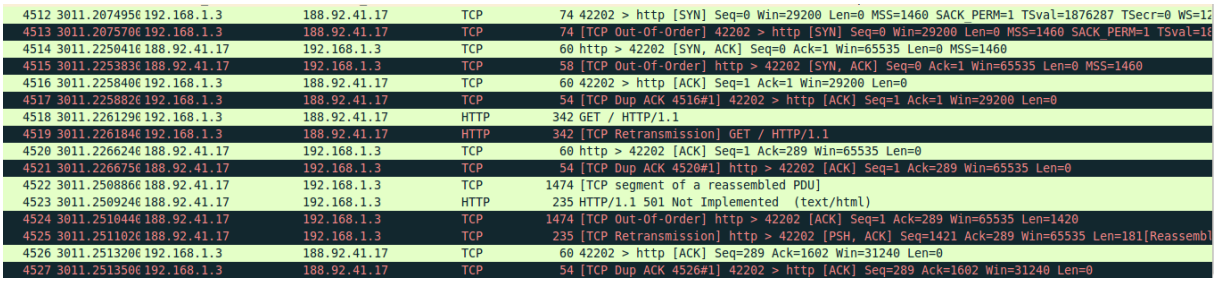
Obrázek 10: Přeposílání IPv6 komunikace ze strany útočníka

Z obrázku (Obr. 9) je patrné, že útočník dostává pakety ping reply. To proto, že oběť obdržela předchozí ping request zprávu, a tak druhé straně odpovídá. Útočník se tedy stal prostředníkem a nyní může celou komunikaci monitorovat a ovládat.

Přeposílání provozu mezi obětí a routerem

Při tomto testu jsem nakonfiguroval zařízení oběti 2 jako router, pomocí kterého se zařízení v síti připojují k internetu. Útočník vstoupí do komunikace mezi obětí a routerem a pokusí se tak ovládnout skutečný provoz.

Útočník otráví cache obětí a spustí aplikaci pro přeposílání provozu. Následně se oběť pokusí dostat na webovou stránku pomocí prohlížeče.



Obrázek 11: Komunikace z pohledu útočníka

Veškerá komunikace probíhá skrze útočníka, který zachytává a přeposílá provoz mezi routerem a obětí toku. Uživatel se přihlásil na webovou stránku, která se mu následně zobrazila v prohlížeči.

Závěr

Cílem projektu bylo nastudovat a implementovat síťový útok Man in the Middle. Aplikace je použitelná pro IPv4 i IPv6 provoz. Testování dílčích částí ukázalo, že není těžké se stát obětí takového útoku. Všechny testy dopadly ve prospěch útočníka. Aplikace je schopna vygenerovat útok Man in the Middle. Z bonusových částí jsem implementoval interaktivní výběr dvojic obětí (chooser).

Literatura

[1] How to defend yourself against MITM or Man-in-the-middle attack [online]. http://hackerspace.kinja.com/how-to-defend-yourself-against-mitm-or-man-in-the-middl-1461796382, [cit. 2017-04-22].

[2] Plummer, D.: An Ethernet Address Resolution Protocol. RFC 826, Listopad 1982

[3] Narten, T.: Neighbor Discovery for IP version 6 (IPv6). RFC 4861, Září 2007