

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	3
Princip útoku	3
Implementace.....	4
Scanner	4
Spoofing	4
Interceptor.....	4
Chooser.....	4
Testování aplikace	5
Skenování sítě.....	5
Otrávení cache obětí	5
Přeposílání provozu	6
Přeposílání provozu mezi obětí a routerem.....	7
Závěr	8
Literatura	9

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.

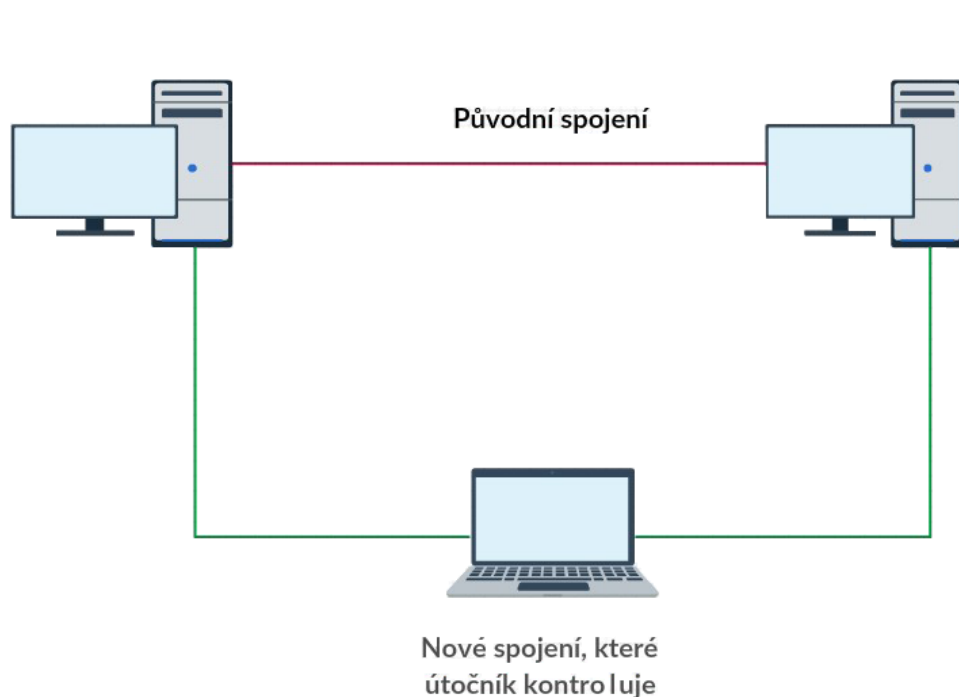
Většina moderních šifrovaných protokolů již poskytuje mechanismus pro ověření na koncové straně právě proto, aby zabránilo útoku Man in the Middle. Například protokol Secure Sockets Layer (SSL) je používán pro ověření jedné nebo obou stran [1].

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 zjištění 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 vektoru. 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 vektoru. Nakonec aplikace vygeneruje výchozí XML soubor, který obsahuje všechna nalezená zařízení včetně jejich adres.

Spoofers

Aplikace jako vstupní parametr obdrží 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.

Interceptor

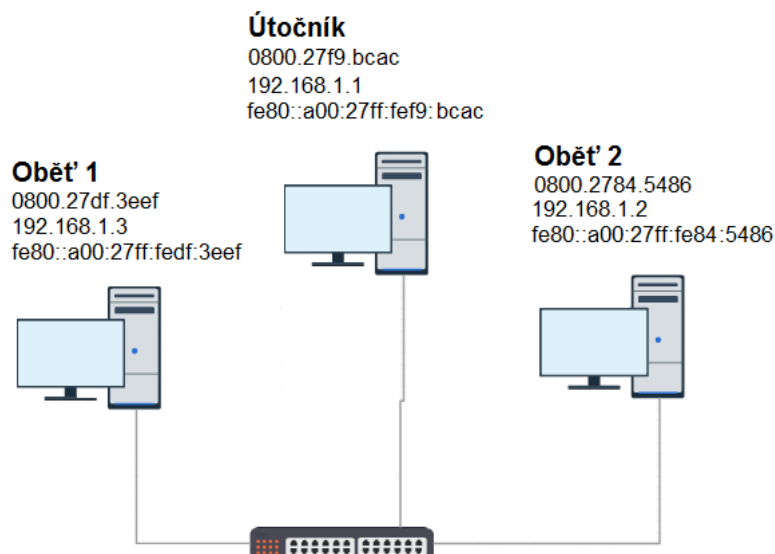
Vstupem aplikace je XML soubor, jenž obsahuje spárovaná zařízení, mezi kterými má přeposílání paketů probíhat. Po zpracování XML souboru si aplikace uloží dvojice do paměti a následně vyčkává v nekonečné smyčce a přijímá pakety. Pokud daný paket náleží nějaké dvojici zařízení, tak cílová linková adresa 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 spáruje 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 Interceptor.

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	1202.7821500	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.

1211	391.09729200	192.168.1.2	192.168.1.3	ICMP	98 Echo (ping) request	id=0x08f9, seq=32/8192, ttl=64 (reply in 1212)
1212	391.09756100	192.168.1.3	192.168.1.2	ICMP	98 Echo (ping) reply	id=0x08f9, seq=32/8192, ttl=64 (request in 1211)
1213	391.96485400	192.168.1.3	192.168.1.2	ICMP	98 Echo (ping) request	id=0x08ce, seq=377/30977, ttl=64 (reply in 1214)
1214	391.96488600	192.168.1.2	192.168.1.3	ICMP	98 Echo (ping) reply	id=0x08ce, seq=377/30977, ttl=64 (request in 1213)
1215	392.09732100	192.168.1.2	192.168.1.3	ICMP	98 Echo (ping) request	id=0x08f9, seq=33/8448, ttl=64 (reply in 1216)
1216	392.09777100	192.168.1.3	192.168.1.2	ICMP	98 Echo (ping) reply	id=0x08f9, seq=33/8448, ttl=64 (request in 1215)
1217	392.96484800	192.168.1.3	192.168.1.2	ICMP	98 Echo (ping) request	id=0x08ce, seq=378/31233, ttl=64 (request in 1218)
1218	392.96488300	192.168.1.2	192.168.1.3	ICMP	98 Echo (ping) reply	id=0x08ce, seq=378/31233, ttl=64 (request in 1217)
1219	393.09791600	192.168.1.2	192.168.1.3	ICMP	98 Echo (ping) request	id=0x08f9, seq=34/8704, ttl=64 (reply in 1220)
1220	393.09845800	192.168.1.3	192.168.1.2	ICMP	98 Echo (ping) reply	id=0x08f9, seq=34/8704, ttl=64 (request in 1219)
1221	393.96616900	192.168.1.3	192.168.1.2	ICMP	98 Echo (ping) request	id=0x08ce, seq=379/31489, ttl=64 (reply in 1222)
1222	393.96620500	192.168.1.2	192.168.1.3	ICMP	98 Echo (ping) reply	id=0x08ce, seq=379/31489, ttl=64 (request in 1221)

Obrázek 5: Komunikace mezi oběťmi

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

1738	6360.4686470	CadmusCo_f9:bc:ac	CadmusCo_84:54:86	ARP	42 192.168.1.3 is at 08:00:27:f9:bc:ac
1739	6360.4686980	CadmusCo_f9:bc:ac	CadmusCo_df:3e:ef	ARP	42 192.168.1.2 is at 08:00:27:f9:bc:ac
1740	6361.0128970	192.168.1.3	192.168.1.2	ICMP	98 Echo (ping) request id=0x08ce, seq=615/26370, ttl=64
1741	6361.0327500	192.168.1.2	192.168.1.3	ICMP	98 Echo (ping) request id=0x08f9, seq=271/3841, ttl=64
1742	6362.0157280	192.168.1.3	192.168.1.2	ICMP	98 Echo (ping) request id=0x08ce, seq=616/26626, ttl=64
1743	6362.0345560	192.168.1.2	192.168.1.3	ICMP	98 Echo (ping) request id=0x08f9, seq=272/4097, ttl=64

Obrázek 6: Odeslání potvrzených 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.

301	88.06005600	fe80::a00:27ff:fedf:3eef	fe80::a00:27ff:fe84:5486	ICMPv6	118 Echo (ping) request id=0x0904, seq=89, hop limit=64 (reply in 302)
302	88.06007300	fe80::a00:27ff:fe84:5486	fe80::a00:27ff:fedf:3eef	ICMPv6	118 Echo (ping) reply id=0x0904, seq=89, hop limit=64 (request in 301)
303	88.63561500	fe80::a00:27ff:fe84:5486	fe80::a00:27ff:fedf:3eef	ICMPv6	118 Echo (ping) request id=0x0943, seq=61, hop limit=64 (reply in 304)
304	88.63616500	fe80::a00:27ff:fedf:3eef	fe80::a00:27ff:fe84:5486	ICMPv6	118 Echo (ping) reply id=0x0943, seq=61, hop limit=64 (request in 303)
305	89.06081200	fe80::a00:27ff:fedf:3eef	fe80::a00:27ff:fe84:5486	ICMPv6	118 Echo (ping) request id=0x0904, seq=90, hop limit=64 (reply in 306)
306	89.06082800	fe80::a00:27ff:fe84:5486	fe80::a00:27ff:fedf:3eef	ICMPv6	118 Echo (ping) reply id=0x0904, seq=90, hop limit=64 (request in 305)
307	89.63505100	fe80::a00:27ff:fe84:5486	fe80::a00:27ff:fedf:3eef	ICMPv6	118 Echo (ping) request id=0x0943, seq=62, hop limit=64 (reply in 308)
308	89.63539800	fe80::a00:27ff:fedf:3eef	fe80::a00:27ff:fe84:5486	ICMPv6	118 Echo (ping) reply id=0x0943, seq=62, hop limit=64 (request in 307)

Obrázek 7: IPv6 Komunikace mezi oběťmi

2503	7018.9436400	fe80::a00:27ff:fedf:3eef	fe80::a00:27ff:fe84:54	ICMPv6	86 Neighbor Advertisement fe80::a00:27ff:fedf:3eef (sol, ovr) is at 08:00:27:f9:bc:ac
2504	7018.9437200	fe80::a00:27ff:fe84:54	fe80::a00:27ff:fedf:3eef	ICMPv6	86 Neighbor Advertisement fe80::a00:27ff:fe84:5486 (sol, ovr) is at 08:00:27:f9:bc:ac
2505	7019.2172890	fe80::a00:27ff:fedf:3eef	fe80::a00:27ff:fe84:54	ICMPv6	118 Echo (ping) request id=0x0904, seq=117, hop limit=64
2506	7019.7879440	fe80::a00:27ff:fe84:54	fe80::a00:27ff:fedf:3eef	ICMPv6	118 Echo (ping) request id=0x0943, seq=89, hop limit=64
2507	7020.2189600	fe80::a00:27ff:fedf:3eef	fe80::a00:27ff:fe84:54	ICMPv6	118 Echo (ping) request id=0x0904, seq=118, hop limit=64
2508	7020.7903120	fe80::a00:27ff:fe84:54	fe80::a00:27ff:fedf:3eef	ICMPv6	118 Echo (ping) request id=0x0943, seq=90, hop limit=64

Obrázek 8: Odeslání potvrzených 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).

5770	8162.3804330	192.168.1.3	192.168.1.2	ICMP	98 Echo (ping) request	id=0x092d, seq=26/6656, ttl=64
5771	8162.3806220	192.168.1.3	192.168.1.2	ICMP	98 Echo (ping) request	id=0x092d, seq=26/6656, ttl=64 (reply in 5772)
5772	8162.3811760	192.168.1.2	192.168.1.3	ICMP	98 Echo (ping) reply	id=0x092d, seq=26/6656, ttl=64 (request in 5771)
5773	8162.3812680	192.168.1.2	192.168.1.3	ICMP	98 Echo (ping) reply	id=0x092d, seq=26/6656, ttl=64
5774	8162.4540830	192.168.1.2	192.168.1.3	ICMP	98 Echo (ping) request	id=0x0959, seq=33/8448, ttl=64
5775	8162.4541730	192.168.1.2	192.168.1.3	ICMP	98 Echo (ping) request	id=0x0959, seq=33/8448, ttl=64 (reply in 5776)
5776	8162.4544510	192.168.1.3	192.168.1.2	ICMP	98 Echo (ping) reply	id=0x0959, seq=33/8448, ttl=64 (request in 5775)
5777	8162.4544810	192.168.1.3	192.168.1.2	ICMP	98 Echo (ping) reply	id=0x0959, seq=33/8448, ttl=64

Obrázek 9: Přeposílání komunikace ze strany útočníka

5086 8051.5431256 fe80::a00:27ff:fedf:3e fe80::a00:27ff:fe84:54 ICMPv6	118 Echo (ping) request id=0x0904, seq=1144, hop limit=64
5087 8051.5432216 fe80::a00:27ff:fedf:3e fe80::a00:27ff:fe84:54 ICMPv6	118 Echo (ping) request id=0x0904, seq=1144, hop limit=64 (reply in 5088)
5088 8051.5435976 fe80::a00:27ff:fe84:54 fe80::a00:27ff:fedf:3e ICMPv6	118 Echo (ping) reply id=0x0904, seq=1144, hop limit=64 (request in 5087)
5089 8051.5436446 fe80::a00:27ff:fe84:54 fe80::a00:27ff:fedf:3e ICMPv6	118 Echo (ping) reply id=0x0904, seq=1144, hop limit=64
5090 8052.4376106 fe80::a00:27ff:fe84:54 fe80::a00:27ff:fedf:3e ICMPv6	118 Echo (ping) request id=0x0943, seq=1117, hop limit=64
5091 8052.4376656 fe80::a00:27ff:fe84:54 fe80::a00:27ff:fedf:3e ICMPv6	118 Echo (ping) request id=0x0943, seq=1117, hop limit=64 (reply in 5092)
5092 8052.4380046 fe80::a00:27ff:fedf:3e fe80::a00:27ff:fe84:54 ICMPv6	118 Echo (ping) reply id=0x0943, seq=1117, hop limit=64 (request in 5091)
5093 8052.4380406 fe80::a00:27ff:fedf:3e fe80::a00:27ff:fe84:54 ICMPv6	118 Echo (ping) reply id=0x0943, seq=1117, hop limit=64

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.

4512 3011.2074956 192.168.1.3	188.92.41.17	TCP	74 42202 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1876287 TSecr=0 WS=12
4513 3011.2075706 192.168.1.3	188.92.41.17	TCP	74 [TCP Out-Of-Order] 42202 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1876287 TSecr=0 WS=12
4514 3011.2250416 188.92.41.17	192.168.1.3	TCP	60 http > 42202 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
4515 3011.2253836 188.92.41.17	192.168.1.3	TCP	58 [TCP Out-Of-Order] http > 42202 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
4516 3011.2258406 192.168.1.3	188.92.41.17	TCP	60 42202 > http [ACK] Seq=1 Ack=1 Win=29200 Len=0
4517 3011.2258826 192.168.1.3	188.92.41.17	TCP	54 [TCP Dup ACK 4516#1] 42202 > http [ACK] Seq=1 Ack=1 Win=29200 Len=0
4518 3011.2261296 192.168.1.3	188.92.41.17	HTTP	342 GET / HTTP/1.1
4519 3011.2261846 192.168.1.3	188.92.41.17	HTTP	342 [TCP Retransmission] GET / HTTP/1.1
4520 3011.2266246 188.92.41.17	192.168.1.3	TCP	60 http > 42202 [ACK] Seq=1 Ack=289 Win=65535 Len=0
4521 3011.2266756 188.92.41.17	192.168.1.3	TCP	54 [TCP Dup ACK 4520#1] http > 42202 [ACK] Seq=1 Ack=289 Win=65535 Len=0
4522 3011.2508866 188.92.41.17	192.168.1.3	TCP	1474 [TCP segment of a reassembled PDU]
4523 3011.2509246 188.92.41.17	192.168.1.3	HTTP	235 HTTP/1.1 501 Not Implemented (text/html)
4524 3011.2510446 188.92.41.17	192.168.1.3	TCP	1474 [TCP Out-Of-Order] http > 42202 [ACK] Seq=1 Ack=289 Win=65535 Len=1420
4525 3011.2511026 188.92.41.17	192.168.1.3	TCP	235 [TCP Retransmission] http > 42202 [PSH, ACK] Seq=1421 Ack=289 Win=65535 Len=181[Reassembled]
4526 3011.2513206 192.168.1.3	188.92.41.17	TCP	60 42202 > http [ACK] Seq=289 Ack=1602 Win=31240 Len=0
4527 3011.2513506 192.168.1.3	188.92.41.17	TCP	54 [TCP Dup ACK 4526#1] 42202 > http [ACK] Seq=289 Ack=1602 Win=31240 Len=0

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-middle-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