Vysoké učení technické v Brně
Fakulta informačních technologií

Síťové aplikace a spáva sítí

Dokumentace k projektu - Tunelování datových přenosů přes DNS dotazy

11. listopadu 2022 David Kocman, xkocma08

Obsah

1 Úvod

Cílem tohoto projektu je posílání dat na námi vytvořený server přes DNS tunel.Tunelování přes DNS dotazy patří mezi mnoho možností exfiltrace dat.

2 Princip

DNS je protokol, který slouží pro překlad URL adres do IP adres. Není to protokol určen pro přenos dat, z tohoto důvodu nebývá tok DNS packetů monitorován a firewally nechávjí packetům vždy volný průběh dovnitř i ven. Proto se zneužívá způsobem, že se data zabalí do DNS paketu a přes port 53 se budou odesílat na útočníkův server.

Útok většinou probíhá takto: [5]

- Útočník zaregistruje doménu. Name server domény odkazuje na server útočníka, kde je nainstalován tunelovací malware program.
- Útočník napadne počítač a posílá DNS dotazy na DNS překladač. Překladač DNS je server, který předává požadavky na adresy IP kořenovým serverům a doménám nejvyšší úrovně.Překladač DNS směruje dotaz na server útočníka, kde je nainstalován tunelovací program.
- Mezi útočníkem a obětí je navázán „tunel“, přes který je možné např. exfiltrovat data nebo tzv. „Command and control“, což znamená provádění jednoduchých příkazů u oběti. Protože neexistuje přímé spojení mezi útočníkem a obětí, je obtížnější vysledovat počítač útočníka.

3 Návrh

Projekt se skládá ze 2 hlavních zdrojových souborů `dns_sender.c` a `dns_receiver.c` pracující na principu klient-server [2]. Jako komunikační protokol mezi serverem a klientem jsem zvolil UDP protokol.

3.1 `dns_sender.c`

`dns_sender.c` posílá zakódovaná data buď přímo na útočníkův server nebo na vzdálený DNS server. Data jsou kódována v base64, jehož funkce jsou definovány v souboru `base64.c` [3]. Poté je vytvořena DNS query hlavička [1], do které jsou vložena data, která byla převedena do DNS formátu funkcí `ChangetoDnsNameFormat` [1], která oddělá znaky které DNS query nepodporuje.

Sender jako první pošle paket s názvem souboru, do kterého se mají data uložit na serveru. Dále se začnou načítat data ze souboru, kde každých 45 znaků se pošle hotový paket. Důvodem je, že zakódované data v base64 jsou delší než nezakódované a subdoména v DNS paketu může mít jen 63 znaků. Po načtení všech dat ze souboru sender pošle ukončující packet a ukončí se.

Na socket je nastaven timeout na 1,5 sekundy [4]. Retransmit ale není podporován.
Důležité funkce:

- `get_args`: získá argumenty z příkazové řádky
- `get_address`: získá adresu DNS serveru ze souboru /etc/resolv.conf, pokud upstream_dns_ip není zadána.
- `ChangetoDnsNameFormat`: převede adresu na DNS formát.[1]
- `send_packet`: funkce odešle hotový packet

3.2 dns_receiver.c

dns_receiver.c naslouchá na implicitním portu pro DNS komunikaci, tedy portu číslo 53. Po zachycení packetů je z „nultého“ dekódován [3] název souboru.

Dále v cyklu jsou data v paketu převedena na formát ve kterém se mezi doménami vyskytují tečky, od paketů je uříznut base_host a data jsou dekódována a zapsána do souboru, který je uložen ve složce dst_filepath. Pokud je base_host různý od zadaného, receiver nepřijmé žádný packet z příchozí adresy. Po přijetí a zapsání odešle DNS response, který dává senderu znamení, že packet přišel.

Data jsou zapisována způsobem, že první se program zeptá jestli dokáže otvřit zadanou složku, resp. zda-li existuje. Pokud ne, je vytvořena. V obou případech pak jsou data do souboru postupně zapisována.

Receiver přijme ukončující packet, udávající konec přenosu jednoho souboru a čeká na přijetí dalšího. V této chvíli je možné stiskem CTRL+C vypnout receiver bez jakéhokoli ztráty paměti.
Důležité funkce:

- pretty_print: funkce nahradí netisknutelné znaky v datech tečkami (inverzní funkce k funkci ChangetoDnsNameFormat popsána nahoře) [6].
- write_data: funkce sloužící k zapsání dekódovaných dat do souboru.
- get_qname: funkce k nahrání dat z DNS query do lokální proměnné.

3.3 base64.c

Zdrojový soubor obsahující base64 kódovací a dekódovací funkce. Kód jsem převzal, nejsem autorem [3].

3.4 custom_dns.h

Hlavičkový soubor obsahující hlavičku odesílaných DNS packetů. Obsahuje struktury DNS_HEADER, QUESTION obsahující typ a třídu DNS query a QUERY obsahující výše zmíněné data a navíc i text obsahující samotnou otázku [1].

3.5 sender.h a receiver.h

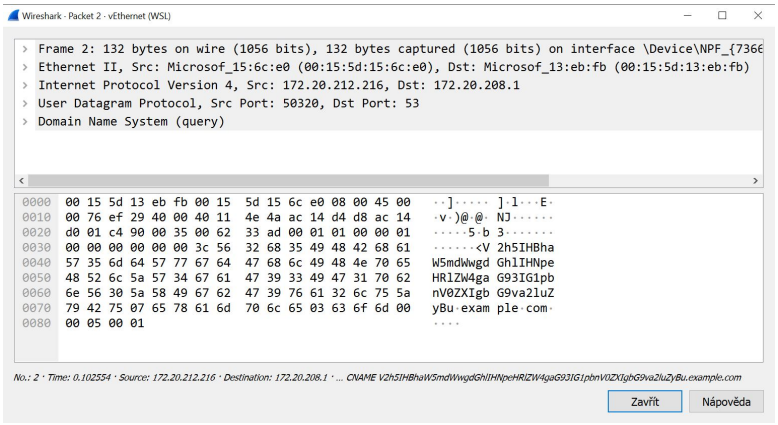
Hlavičkové soubory obsahující definice funkcí, knihovny a konstanty použity ve zdrojových souborech.

4 Testování a měření

Sender i receiver testován na Evě, ale s jiným číslem portu!

Testovaní mého programmu probíhalo vytvářením různých souborů s daty a jejich posíláním na server. Po dekódování a zapsání jsem přístrojem diff porovnal obsah obou souborů. Při práci jsem i pozorně porovnával velikost odeslaného souboru.
Velikost souboru není omezena, protože načítám data postupně a hned posílám packet. Je tu ale možnost, že UDP packet se cestou ztratí a nedojde na stranu příjemce. Valgrind neukazoval žádné chyby a všechna alokace je uvolněna.

Program odesílá platné DNS packety.



Zde je příklad z testování:

```
[INIT] 127.0.0.1
[SENT] data.txt         0 8B to 127.0.0.1
[ENCD] data.txt         1 'SGUgYXMgY29tcGxpbWVudCB1bnJlc2VydmVkIHByb2plY3RpbmcuIEJldHdl.example.com'
[SENT] data.txt         1 45B to 127.0.0.1
[ENCD] data.txt         2 'ZW4gaGFkIG9ic2VydmUgcHJldGVuZCBkZWxpZ2h0IGZvciBiZWxpZXZlLiBE.example.com'
[SENT] data.txt         2 45B to 127.0.0.1
[ENCD] data.txt         3 'byBuZXdzcGFwZXIgcXVlc3Rpb25zIGNvbnN1bHRlZCBzd2VldG5lc3MgZG8u.example.com'
[SENT] data.txt         3 45B to 127.0.0.1
[ENCD] data.txt         4 'IE91ciBzcG9ydHNtYW4gaGlzIHVud2lsbGluZyBmdWxmaWxsZWQgZGVwYXJ0.example.com'
[SENT] data.txt         4 45B to 127.0.0.1
[ENCD] data.txt         5 'dXJlIGxhdy4gTm93IHdvcmxkIG93biB0b3RhbCBzYXZlZCBhYm92ZSBoZXIg.example.com'
[SENT] data.txt         5 45B to 127.0.0.1
[ENCD] data.txt         6 'Y2F1c2UgdGFibGUuIFdpY2tldCBteXNlbGYgaGVyIHNxdWFyZSByZW1hcmsg.example.com'
[SENT] data.txt         6 45B to 127.0.0.1
[ENCD] data.txt         7 'dGhlIHNob3VsZCBmYXIgc2VjdXJlIHNleC4gU21pbGluZyBjb3VzaW5zIHdh.example.com'
[SENT] data.txt         7 45B to 127.0.0.1
[ENCD] data.txt         8 'cnJhbnQgbGF3IGV4cGxhaW4gZm9yIHdoZXRoZXIu.example.com'
[SENT] data.txt         8 30B to 127.0.0.1
[SENT] data.txt         9 3B to 127.0.0.1
[CMPL] data.txt of 345B
```

Obrázek 1: Odeslání dat ze souboru `file.txt`.

```
[INIT] 127.0.0.1
[PARS] ./data/data.txt 'ZGF0YS50eHQ=.example.com'
[PARS] ./data/data.txt 'SGUgYXMgY29tcGxpbWVudCB1bnJlc2VydmVkIHByb2plY3RpbmcuIEJldHdl.example.com'
[RECV] ./data/data.txt         1 45B from 127.0.0.1
[PARS] ./data/data.txt 'ZW4gaGFkIG9ic2VydmUgcHJldGVuZCBkZWxpZ2h0IGZvciBiZWxpZXZlLiBE.example.com'
[RECV] ./data/data.txt         2 45B from 127.0.0.1
[PARS] ./data/data.txt 'byBuZXdzcGFwZXIgcXVlc3Rpb25zIGNvbnN1bHRlZCBzd2VldG5lc3MgZG8u.example.com'
[RECV] ./data/data.txt         3 45B from 127.0.0.1
[PARS] ./data/data.txt 'IE91ciBzcG9ydHNtYW4gaGlzIHVud2lsbGluZyBmdWxmaWxsZWQgZGVwYXJ0.example.com'
[RECV] ./data/data.txt         4 45B from 127.0.0.1
[PARS] ./data/data.txt 'dXJlIGxhdy4gTm93IHdvcmxkIG93biB0b3RhbCBzYXZlZCBhYm92ZSBoZXIg.example.com'
[RECV] ./data/data.txt         5 45B from 127.0.0.1
[PARS] ./data/data.txt 'Y2F1c2UgdGFibGUuIFdpY2tldCBteXNlbGYgaGVyIHNxdWFyZSByZW1hcmsg.example.com'
[RECV] ./data/data.txt         6 45B from 127.0.0.1
[PARS] ./data/data.txt 'dGhlIHNob3VsZCBmYXIgc2VjdXJlIHNleC4gU21pbGluZyBjb3VzaW5zIHdh.example.com'
[RECV] ./data/data.txt         7 45B from 127.0.0.1
[PARS] ./data/data.txt 'cnJhbnQgbGF3IGV4cGxhaW4gZm9yIHdoZXRoZXIu.example.com'
[RECV] ./data/data.txt         8 30B from 127.0.0.1
[PARS] ./data/data.txt 'ZW5k.example.com'
[RECV] ./data/data.txt         9 3B from 127.0.0.1
[CMPL] ./data of 345B
```

Obrázek 2: Přijetí těchto dat.

Obrázek 3: Porovnání velikostí odeslaných souborů. Můžeme vidět, že velikost odeslaných a přijatých dat je stejná stejně jako velikost souboru.

```
$ diff -s ./file.txt ./data/data.txt
Files ./file.txt and ./data/data.txt are identical
```

Obrázek 4: Finální porovnání obou souborů. Můžeme vidět, že jsou identické.

5 Omezení

- Program nepodporuje přenášení binárních souborů.
- Program nepodporuje retransmit ztracených packetů.

6 Zdroje

Reference

[1] BinaryTides: *DNS Query Code in C with Linux sockets*. [online], [viděno 17.10.2022].
URL <https://www.binarytides.com/dns-query-code-in-c-with-linux-sockets/>

[2] Geeksforgeeks: *UDP Server-Client implementation in C*. [online], [viděno 16.10.2022].
URL <https://www.geeksforgeeks.org/udp-server-client-implementation-c/>

[3] Nachtimwald, J.: *Base64 Encode and Decode in C*. [online], [viděno 17.10.2022].
URL <https://nachtimwald.com/2017/11/18/base64-encode-and-decode-in-c/>

[4] Neal: *UDP Socket Set Timeout*. [online], [viděno 16.10.2022].
URL <https://stackoverflow.com/questions/13547721/udp-socket-set-timeout>

[5] Networks, P. A.: *DNS Tunneling – co je to?* [online], [viděno 19.10.2022].
URL <https:https://nextgenfw.cz/2020/03/10/dns-tunneling-co-je-to/>

[6] user405725: *Removing spaces and special characters from string*. [online], [viděno 17.10.2022].
URL <https://stackoverflow.com/questions/15444567/removing-spaces-and-special-characters-from-string>