ICMP Tunneling

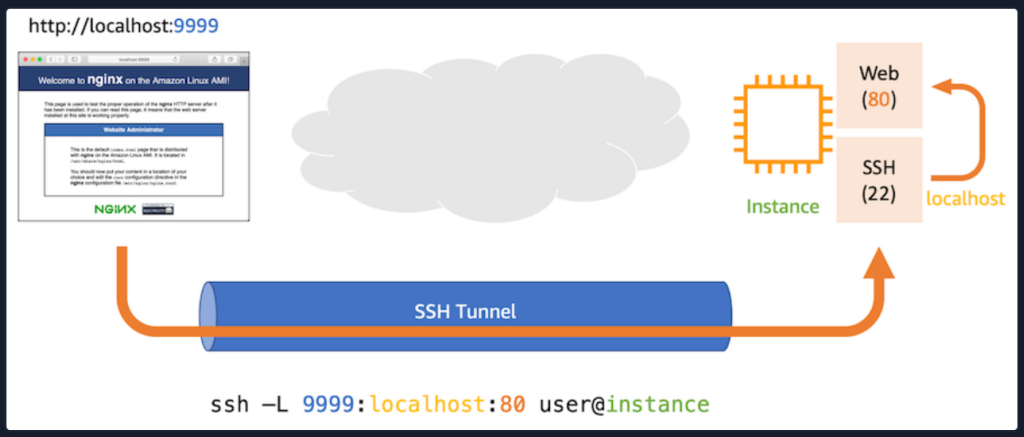
Related PCAP File(s):

- `icmp_tunneling.pcapng`

Tunneling is a technique employed by adversaries in order to exfiltrate data from one location to another. There are many different kinds of tunneling, and each different kind uses a different protocol. Commonly, attackers may utilize proxies to bypass our network controls, or protocols that our systems and controls allow.

Basics of Tunneling

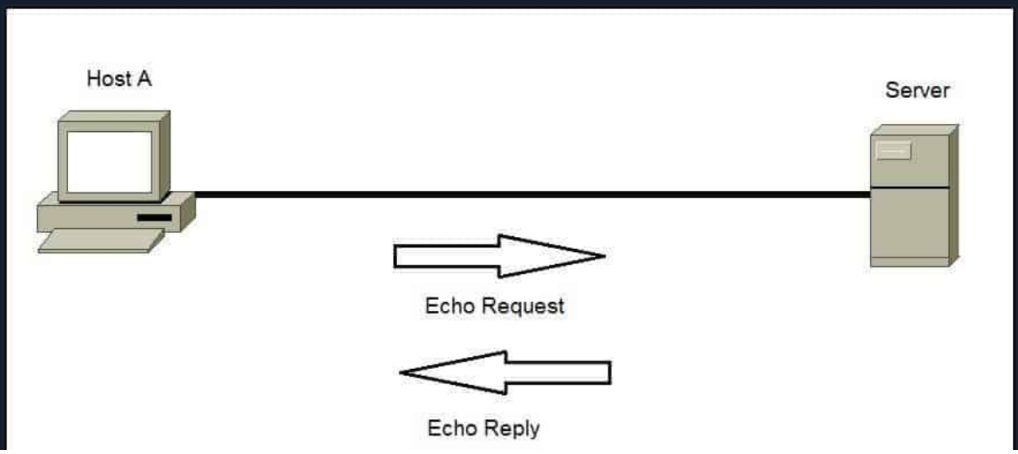
Essentially, when an attacker wants to communicate data to another host, they may employ tunneling. In many cases, we might notice this through the attacker possessing some command and control over one of our machines. As noted, tunneling can be conducted in many different ways. One of the more common types is SSH tunneling. However, proxy-based, HTTP, HTTPs, DNS, and other types can be observed in similar ways.



The idea behind tunneling is that an attacker will be able to expand their command and control and bypass our network controls through the protocol of their choosing.

ICMP Tunneling

In the case of ICMP tunneling an attacker will append data they want to exfiltrate to the outside world or another host in the data field in an ICMP request. This is done with the intention to hide this data among a common protocol type like ICMP, and hopefully get lost within our network traffic.



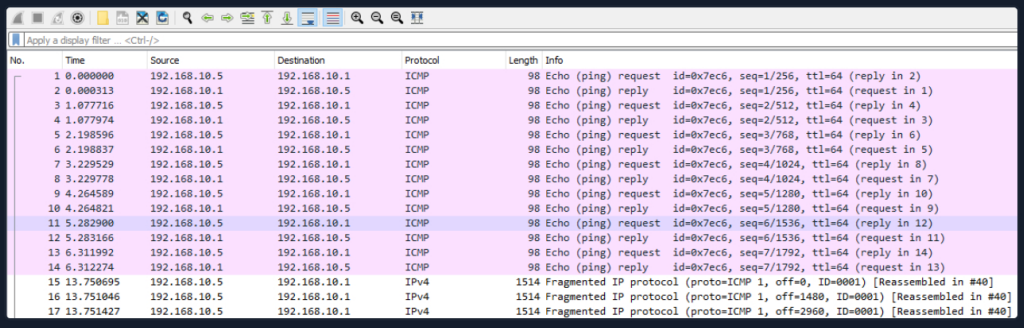
OFFLINE

Start Instance

∞ / 1 spawns left

Finding ICMP Tunneling

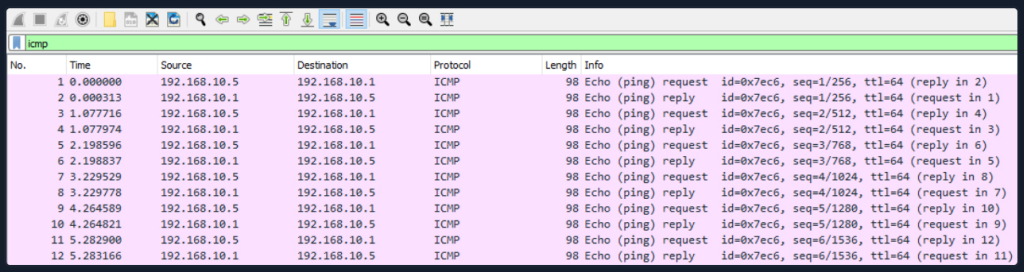
Since ICMP tunneling is primarily done through an attacker adding data into the data field for ICMP, we can find it by looking at the contents of data per request and reply.



Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.10.5	192.168.10.1	ICMP	98	Echo (ping) request id=0x7ec6, seq=1/256, ttl=64 (reply in 2)
2	0.000313	192.168.10.1	192.168.10.5	ICMP	98	Echo (ping) reply id=0x7ec6, seq=1/256, ttl=64 (request in 1)
3	1.077716	192.168.10.5	192.168.10.1	ICMP	98	Echo (ping) request id=0x7ec6, seq=2/512, ttl=64 (reply in 4)
4	1.077974	192.168.10.1	192.168.10.5	ICMP	98	Echo (ping) reply id=0x7ec6, seq=2/512, ttl=64 (request in 3)
5	2.198596	192.168.10.5	192.168.10.1	ICMP	98	Echo (ping) request id=0x7ec6, seq=3/768, ttl=64 (reply in 6)
6	2.198837	192.168.10.1	192.168.10.5	ICMP	98	Echo (ping) reply id=0x7ec6, seq=3/768, ttl=64 (request in 5)
7	3.229529	192.168.10.5	192.168.10.1	ICMP	98	Echo (ping) request id=0x7ec6, seq=4/1024, ttl=64 (reply in 8)
8	3.229778	192.168.10.1	192.168.10.5	ICMP	98	Echo (ping) reply id=0x7ec6, seq=4/1024, ttl=64 (request in 7)
9	4.264589	192.168.10.5	192.168.10.1	ICMP	98	Echo (ping) request id=0x7ec6, seq=5/1280, ttl=64 (reply in 10)
10	4.264821	192.168.10.1	192.168.10.5	ICMP	98	Echo (ping) reply id=0x7ec6, seq=5/1280, ttl=64 (request in 9)
11	5.282900	192.168.10.5	192.168.10.1	ICMP	98	Echo (ping) request id=0x7ec6, seq=6/1536, ttl=64 (reply in 12)
12	5.283166	192.168.10.1	192.168.10.5	ICMP	98	Echo (ping) reply id=0x7ec6, seq=6/1536, ttl=64 (request in 11)
13	6.311992	192.168.10.5	192.168.10.1	ICMP	98	Echo (ping) request id=0x7ec6, seq=7/1792, ttl=64 (reply in 14)
14	6.312274	192.168.10.1	192.168.10.5	ICMP	98	Echo (ping) reply id=0x7ec6, seq=7/1792, ttl=64 (request in 13)
15	13.750695	192.168.10.5	192.168.10.1	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=0001) [Reassembled in #40]
16	13.751046	192.168.10.5	192.168.10.1	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=0001) [Reassembled in #40]
17	13.751427	192.168.10.5	192.168.10.1	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=0001) [Reassembled in #40]

We can filter our wireshark capture to only ICMP requests and replies by entering ICMP into the filter bar.



icmp

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.10.5	192.168.10.1	ICMP	98	Echo (ping) request id=0x7ec6, seq=1/256, ttl=64 (reply in 2)
2	0.000313	192.168.10.1	192.168.10.5	ICMP	98	Echo (ping) reply id=0x7ec6, seq=1/256, ttl=64 (request in 1)
3	1.077716	192.168.10.5	192.168.10.1	ICMP	98	Echo (ping) request id=0x7ec6, seq=2/512, ttl=64 (reply in 4)
4	1.077974	192.168.10.1	192.168.10.5	ICMP	98	Echo (ping) reply id=0x7ec6, seq=2/512, ttl=64 (request in 3)
5	2.198596	192.168.10.5	192.168.10.1	ICMP	98	Echo (ping) request id=0x7ec6, seq=3/768, ttl=64 (reply in 6)
6	2.198837	192.168.10.1	192.168.10.5	ICMP	98	Echo (ping) reply id=0x7ec6, seq=3/768, ttl=64 (request in 5)
7	3.229529	192.168.10.5	192.168.10.1	ICMP	98	Echo (ping) request id=0x7ec6, seq=4/1024, ttl=64 (reply in 8)
8	3.229778	192.168.10.1	192.168.10.5	ICMP	98	Echo (ping) reply id=0x7ec6, seq=4/1024, ttl=64 (request in 7)
9	4.264589	192.168.10.5	192.168.10.1	ICMP	98	Echo (ping) request id=0x7ec6, seq=5/1280, ttl=64 (reply in 10)
10	4.264821	192.168.10.1	192.168.10.5	ICMP	98	Echo (ping) reply id=0x7ec6, seq=5/1280, ttl=64 (request in 9)
11	5.282900	192.168.10.5	192.168.10.1	ICMP	98	Echo (ping) request id=0x7ec6, seq=6/1536, ttl=64 (reply in 12)
12	5.283166	192.168.10.1	192.168.10.5	ICMP	98	Echo (ping) reply id=0x7ec6, seq=6/1536, ttl=64 (request in 11)

Suppose we noticed fragmentation occurring within our ICMP traffic as it is above, this would indicate a large amount of data being transferred via ICMP. In order to understand this behavior, we should look at a normal ICMP request. We may note that the data is something reasonable like 48 bytes.

```
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xd930 [correct]
    [Checksum Status: Good]
    Identifier (BE): 32454 (0x7ec6)
    Identifier (LE): 50814 (0xc67e)
    Sequence Number (BE): 1 (0x0001)
    Sequence Number (LE): 256 (0x0100)
    [Response frame: 2]
    Timestamp from icmp data: Jul 17, 2023 15:43:14.000000000 Mountain Daylight Time
    [Timestamp from icmp data (relative): 0.331815000 seconds]
  Data (48 bytes)
      Data: 341a0500000000001011121314151617181 91a1b1c1d1e1f202122232425262728292a2b…
      [Length: 48]
```

However a suspicious ICMP request might have a large data length like 38000 bytes.

```
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x4ab7 [correct]
    [Checksum Status: Good]
    Identifier (BE): 0 (0x0000)
    Identifier (LE): 0 (0x0000)
    Sequence Number (BE): 0 (0x0000)
    Sequence Number (LE): 0 (0x0000)
    [Response frame: 66]
  Data (38000 bytes)
      Data: 557365726e616d653a20726f6f743b2050617373776f72643a2050617373776f72643132…
      [Length: 38000]
```

If we would like to take a look at the data in transit, we can look on the right side of our screen in Wireshark. In this case, we might notice something like a Username and Password being pinged to an external or internal host. This is a direct indication of ICMP tunneling.

```
0000  08 00 4a b7 00 00 00 00  55 73 65 72 6e 61 6d 65   ··J····· Username
0010  3a 20 72 6f 6f 74 3b 20  50 61 73 73 77 6f 72 64   : root;  Password
0020  3a 20 50 61 73 73 77 6f  72 64 31 32 33 24 55 73   : Passwo rd123$Us
0030  65 72 6e 61 6d 65 3a 20  72 6f 6f 74 3b 20 50 61   ername:  root; Pa
0040  73 73 77 6f 72 64 3a 20  50 61 73 73 77 6f 72 64   ssword:  Password
0050  31 32 33 24 55 73 65 72  6e 61 6d 65 3a 20 72 6f   123$User name: ro
0060  6f 74 3b 20 50 61 73 73  77 6f 72 64 3a 20 50 61   ot; Pass word: Pa
0070  73 73 77 6f 72 64 31 32  33 24 55 73 65 72 6e 61   ssword12 3$Userna
0080  6d 65 3a 20 72 6f 6f 74  3b 20 50 61 73 73 77 6f   me: root ; Passwo
0090  72 64 3a 20 50 61 73 73  77 6f 72 64 31 32 33 24   rd: Pass word123$
00a0  55 73 65 72 6e 61 6d 65  3a 20 72 6f 6f 74 3b 20   Username : root;
00b0  50 61 73 73 77 6f 72 64  3a 20 50 61 73 73 77 6f   Password : Passwo
00c0  72 64 31 32 33 24 55 73  65 72 6e 61 6d 65 3a 20   rd123$Us ername:
00d0  72 6f 6f 74 3b 20 50 61  73 73 77 6f 72 64 3a 20   root; Pa ssword:
00e0  50 61 73 73 77 6f 72 64  31 32 33 24 55 73 65 72   Password 123$User
00f0  6e 61 6d 65 3a 20 72 6f  6f 74 3b 20 50 61 73 73   name: ro ot; Pass
0100  77 6f 72 64 3a 20 50 61  73 73 77 6f 72 64 31 32   word: Pa ssword12
0110  33 24 55 73 65 72 6e 61  6d 65 3a 20 72 6f 6f 74   3$Userna me: root
0120  3b 20 50 61 73 73 77 6f  72 64 3a 20 50 61 73 73   ; Passwo rd: Pass
0130  77 6f 72 64 31 32 33 24  55 73 65 72 6e 61 6d 65   word123$ Username
0140  3a 20 72 6f 6f 74 3b 20  50 61 73 73 77 6f 72 64   : root;  Password
0150  3a 20 50 61 73 73 77 6f  72 64 31 32 33 24 55 73   : Passwo rd123$Us
0160  65 72 6e 61 6d 65 3a 20  72 6f 6f 74 3b 20 50 61   ername:  root; Pa
0170  73 73 77 6f 72 64 3a 20  50 61 73 73 77 6f 72 64   ssword:  Password
0180  31 32 33 24 55 73 65 72  6e 61 6d 65 3a 20 72 6f   123$User name: ro
0190  6f 74 3b 20 50 61 73 73  77 6f 72 64 3a 20 50 61   ot; Pass word: Pa
01a0  73 73 77 6f 72 64 31 32  33 24 55 73 65 72 6e 61   ssword12 3$Userna
01b0  6d 65 3a 20 72 6f 6f 74  3b 20 50 61 73 73 77 6f   me: root ; Passwo
01c0  72 64 3a 20 50 61 73 73  77 6f 72 64 31 32 33 24   rd: Pass word123$
01d0  55 73 65 72 6e 61 6d 65  3a 20 72 6f 6f 74 3b 20   Username : root;
01e0  50 61 73 73 77 6f 72 64  3a 20 50 61 73 73 77 6f   Password : Passwo
01f0  72 64 31 32 33 24 55 73  65 72 6e 61 6d 65 3a 20   rd123$Us ername:
0200  72 6f 6f 74 3b 20 50 61  73 73 77 6f 72 64 3a 20   root; Pa ssword:
0210  50 61 73 73 77 6f 72 64  31 32 33 24 55 73 65 72   Password 123$User
0220  6e 61 6d 65 3a 20 72 6f  6f 74 3b 20 50 61 73 73   name: ro ot; Pass
0230  77 6f 72 64 3a 20 50 61  73 73 77 6f 72 64 31 32   word: Pa ssword12
0240  33 24 55 73 65 72 6e 61  6d 65 3a 20 72 6f 6f 74   3$Userna me: root
```

On the other hand, more advanced adversaries will utilize encoding or encryption when transmitting exfiltrated data, even in the case of ICMP tunneling. Suppose we noticed the following.

```
0000  00 00 d0 ed 00 00 00 00  56 47 68 70 63 79 42 70   ···· ···· VGhpcyBp
0010  63 79 42 68 49 48 4e 6c  59 33 56 79 5a 53 42 72   cyBhIHNl Y3VyZSBr
0020  5a 58 6b 36 49 45 74 6c  65 54 45 79 4d 7a 51 31   ZXk6IEtl eTEyMzQ1
0030  4e 6a 63 34 4f 51 6f 3d  56 47 68 70 63 79 42 70   Njc4OQo= VGhpcyBp
0040  63 79 42 68 49 48 4e 6c  59 33 56 79 5a 53 42 72   cyBhIHNl Y3VyZSBr
0050  5a 58 6b 36 49 45 74 6c  65 54 45 79 4d 7a 51 31   ZXk6IEtl eTEyMzQ1
0060  4e 6a 63 34 4f 51 6f 3d  56 47 68 70 63 79 42 70   Njc4OQo= VGhpcyBp
0070  63 79 42 68 49 48 4e 6c  59 33 56 79 5a 53 42 72   cyBhIHNl Y3VyZSBr
0080  5a 58 6b 36 49 45 74 6c  65 54 45 79 4d 7a 51 31   ZXk6IEtl eTEyMzQ1
0090  4e 6a 63 34 4f 51 6f 3d  56 47 68 70 63 79 42 70   Njc4OQo= VGhpcyBp
00a0  63 79 42 68 49 48 4e 6c  59 33 56 79 5a 53 42 72   cyBhIHNl Y3VyZSBr
00b0  5a 58 6b 36 49 45 74 6c  65 54 45 79 4d 7a 51 31   ZXk6IEtl eTEyMzQ1
00c0  4e 6a 63 34 4f 51 6f 3d  56 47 68 70 63 79 42 70   Njc4OQo= VGhpcyBp
00d0  63 79 42 68 49 48 4e 6c  59 33 56 79 5a 53 42 72   cyBhIHNl Y3VyZSBr
00e0  5a 58 6b 36 49 45 74 6c  65 54 45 79 4d 7a 51 31   ZXk6IEtl eTEyMzQ1
00f0  4e 6a 63 34 4f 51 6f 3d  56 47 68 70 63 79 42 70   Njc4OQo= VGhpcyBp
0100  63 79 42 68 49 48 4e 6c  59 33 56 79 5a 53 42 72   cyBhIHNl Y3VyZSBr
0110  5a 58 6b 36 49 45 74 6c  65 54 45 79 4d 7a 51 31   ZXk6IEtl eTEyMzQ1
0120  4e 6a 63 34 4f 51 6f 3d  56 47 68 70 63 79 42 70   Njc4OQo= VGhpcyBp
0130  63 79 42 68 49 48 4e 6c  59 33 56 79 5a 53 42 72   cyBhIHNl Y3VyZSBr
0140  5a 58 6b 36 49 45 74 6c  65 54 45 79 4d 7a 51 31   ZXk6IEtl eTEyMzQ1
0150  4e 6a 63 34 4f 51 6f 3d  56 47 68 70 63 79 42 70   Njc4OQo= VGhpcyBp
0160  63 79 42 68 49 48 4e 6c  59 33 56 79 5a 53 42 72   cyBhIHNl Y3VyZSBr
0170  5a 58 6b 36 49 45 74 6c  65 54 45 79 4d 7a 51 31   ZXk6IEtl eTEyMzQ1
0180  4e 6a 63 34 4f 51 6f 3d  56 47 68 70 63 79 42 70   Njc4OQo= VGhpcyBp
0190  63 79 42 68 49 48 4e 6c  59 33 56 79 5a 53 42 72   cyBhIHNl Y3VyZSBr
01a0  5a 58 6b 36 49 45 74 6c  65 54 45 79 4d 7a 51 31   ZXk6IEtl eTEyMzQ1
01b0  4e 6a 63 34 4f 51 6f 3d  56 47 68 70 63 79 42 70   Njc4OQo= VGhpcyBp
01c0  63 79 42 68 49 48 4e 6c  59 33 56 79 5a 53 42 72   cyBhIHNl Y3VyZSBr
01d0  5a 58 6b 36 49 45 74 6c  65 54 45 79 4d 7a 51 31   ZXk6IEtl eTEyMzQ1
```

```
01e0  4e 6a 63 34 4f 51 6f 3d  56 47 68 70 63 79 42 70   Njc4OQo= VGhpcyBp
01f0  63 79 42 68 49 48 4e 6c  59 33 56 79 5a 53 42 72   cyBhIHNl Y3VyZSBr
0200  5a 58 6b 36 49 45 74 6c  65 54 45 79 4d 7a 51 31   ZXk6IEtl eTEyMzQ1
0210  4e 6a 63 34 4f 51 6f 3d  56 47 68 70 63 79 42 70   Njc4OQo= VGhpcyBp
0220  63 79 42 68 49 48 4e 6c  59 33 56 79 5a 53 42 72   cyBhIHNl Y3VyZSBr
0230  5a 58 6b 36 49 45 74 6c  65 54 45 79 4d 7a 51 31   ZXk6IEtl eTEyMzQ1
0240  4e 6a 63 34 4f 51 6f 3d  56 47 68 70 63 79 42 70   Njc4OQo= VGhpcyBp
```

We could copy this value out of Wireshark and decode it within linux with the base64 utility.

ICMP Tunneling

```
MisaelMacias@htb[/htb]$ echo 'VGhpcyBpcyBhIHNlY3VyZSBrZXk6IEtleTEyMzQ1Njc4OQo=' | base64 -d
```

This would also be a case where ICMP tunneling is observed. In many cases, if the ICMP data length is larger than 48-bytes, we know something fishy is going on, and should always look into it.

Preventing ICMP Tunneling

In order to prevent ICMP tunneling from occurring we can conduct the following actions.

1. `Block ICMP Requests` - Simply, if ICMP is not allowed, attackers will not be able to utilize it.
2. `Inspect ICMP Requests and Replies for Data` - Stripping data, or inspecting data for malicious content on these requests and replies can allow us better insight into our environment, and the ability to prevent this data exfiltration.

Enable step-by-step solutions for all questions

Questions

Answer the question(s) below to complete this Section and earn cubes!

+ 1 Enter the decoded value of the base64-encoded string that was mentioned in this section as your answer.

This is a secure key: Key123456789

Submit

Previous Next

Mark Complete & Next