



ALL POSTS

NETWORKING

CYBER SECURITY

MORE ▾



Login / Sign up

Brandon Hitzel · Sep 25 · 13 min read

⋮

# 5 Easy Router Protection Techniques - includes Attack and Packet Analysis

Updated: Sep 27

Executing attacks against a router using Kali Linux along with Wireshark and the easy steps to protect against them.

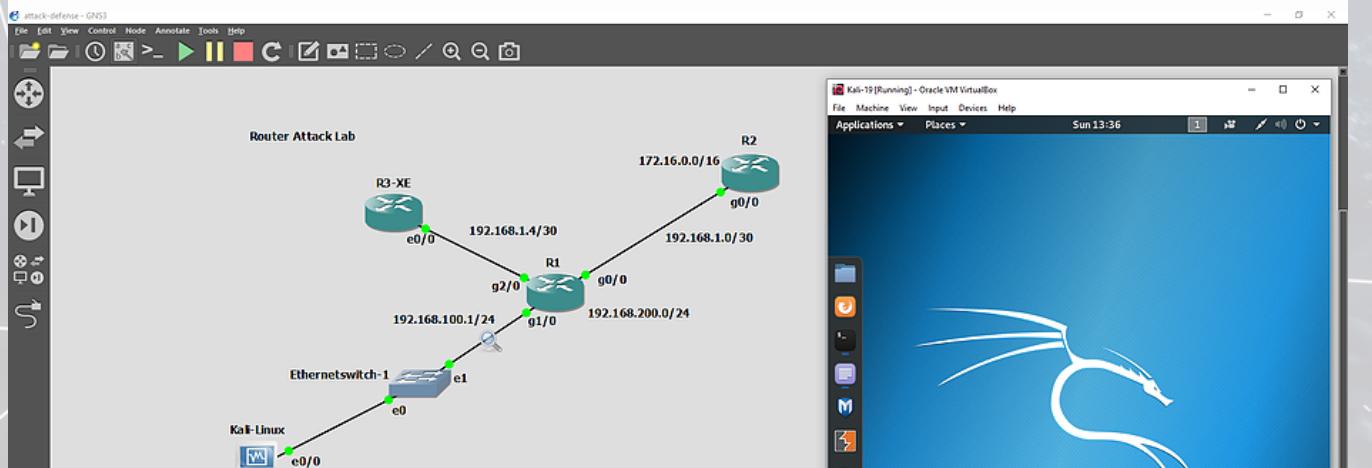
---

I'm sure along your networking or security studies you've read about or came across articles about device hardening techniques or best practices. Doing things like disabling unused services and applying certain configurations in order to better protect your management/control/data planes etc. But how do the attacks look against the router or protocol you are protecting?

My main goal was to execute a few attacks and look at them in [Wireshark](#) to analyze, along with checking out what tools are easily available to accomplish this, as opposed to just showing some configuration and saying this is what you need to do.

One of the main objectives of a network and cyber security is **availability**, so you'll notice a theme of Denial of Service (DoS) attempts throughout the post. If a DoS condition is achieved against a router or switch then connection availability is affected. Thus, one of the main points of Network Defense shall be to harden your devices from such DoS attacks!

So for those that don't know you can run operating systems in [Oracle Virtual Box](#) like [Kali Linux](#) and then integrate that into your GNS3 topology which is what I did here. All the programs shown below were installed by default, except for Loki which is an older software that focuses on routing protocols. Loki was a challenge to install due to its dependencies and since most guides are from 2013 but if I can figure it out I'm sure others can as well. I actually went to one of my old virtual box VDI VHDs from 2017 which had Kali w/ Loki on it to refresh my memory. That is one reason its good to save items from your labs because if you ever have to go back it's serves as a good refresher.





Router Attack Lab Topology in GNS3

Here is the GNS3 lab topology. I have 3 routers, 2x Cisco IOS and 1x Cisco IOS-XE and a virtual box with the newest Kali Linux 2019 distro. What I will probably do in the future is replace one of the virtual IOS routers with a Juniper router to perform some more testing with a different vendor. Likely I will have 1 or 2 more posts related to this. In this post I mainly focused on the local router (aka default gateway or first hop) for my target. The magnifying glass is where the packet captures were taken.

During my research and in the process I found most of the out-of-the-box tools and exploits specific to what I was looking to do lack luster and aged (i.e. pre-2010) and didn't experience stellar results, but that isn't to say there are limited options. Nevertheless, in this article I'll go through a few simple attacks and the easy steps you should take to protect your router or other networking devices against them.

#### TL;DR

1. Stop the enabled protocols you are using from broadcasting information
  2. Secure the protocols you *are* using
  3. Evaluate what services you are using and disable unused services
  4. Secure everything with access control lists
  5. Keep your software/firmware/OS up to date
- Bonus: Be aware of what information your device is sharing

---

#### CDP/LLDP

Your first step is to disable Cisco Discover Protocol (CDP) and Link Layer Discovery Protocol (LLDP) on unnecessary interfaces such as those of your guest networks/VLANs or on user facing subnets etc. These protocols are commonly enabled by default and send traffic out to find information about directly connected neighbors. Although very helpful, these two protocols can easily cause DoS conditions for your network devices.

This was the easiest and most successful attack of the few I looked at. You basically have a choice of to just recon the packets and obtain information, or gain the information and then perform a flooding disruption. Since these protocols are Layer 2, by just connecting to a LAN and running Wireshark you are able to see these packets without even obtaining an IP address (so the infiltrator can be STEALTH).

No.	Time	Source	Src port	Destination	Dst Port	Protocol	Length	Info
8	210.962539	ca:01:2b:18:00:1c		ca:01:2b:18:00:1c		LOOP	60	Reply
9	214.841973	192.168.100.1		224.0.0.5		OSPF	90	Hello Packet
10	223.728116	ca:01:2b:18:00:1c		ca:01:2b:18:00:1c		LOOP	60	Reply
11	226.942281	192.168.100.1		224.0.0.5		OSPF	90	Hello Packet
12	229.252022	ca:01:2b:18:00:1c		CDP/VTTP/DTP/PAgP/-		CDP	397	Device ID: Router1.networkdefenseblog.com Port ID: GigabitEthernet1/0
13	236.441423	ca:01:2b:18:00:1c		ca:01:2b:18:00:1c		LOOP	60	Reply
14	239.622050	192.168.100.1		224.0.0.5		OSPF	90	Hello Packet
15	248.657074	ca:01:2b:18:00:1c		ca:01:2b:18:00:1c		LOOP	60	Reply
16	250.515224	192.168.100.1		224.0.0.5		OSPF	90	Hello Packet
17	260.431127	ca:01:2b:18:00:1c		ca:01:2b:18:00:1c		LOOP	60	Reply

```

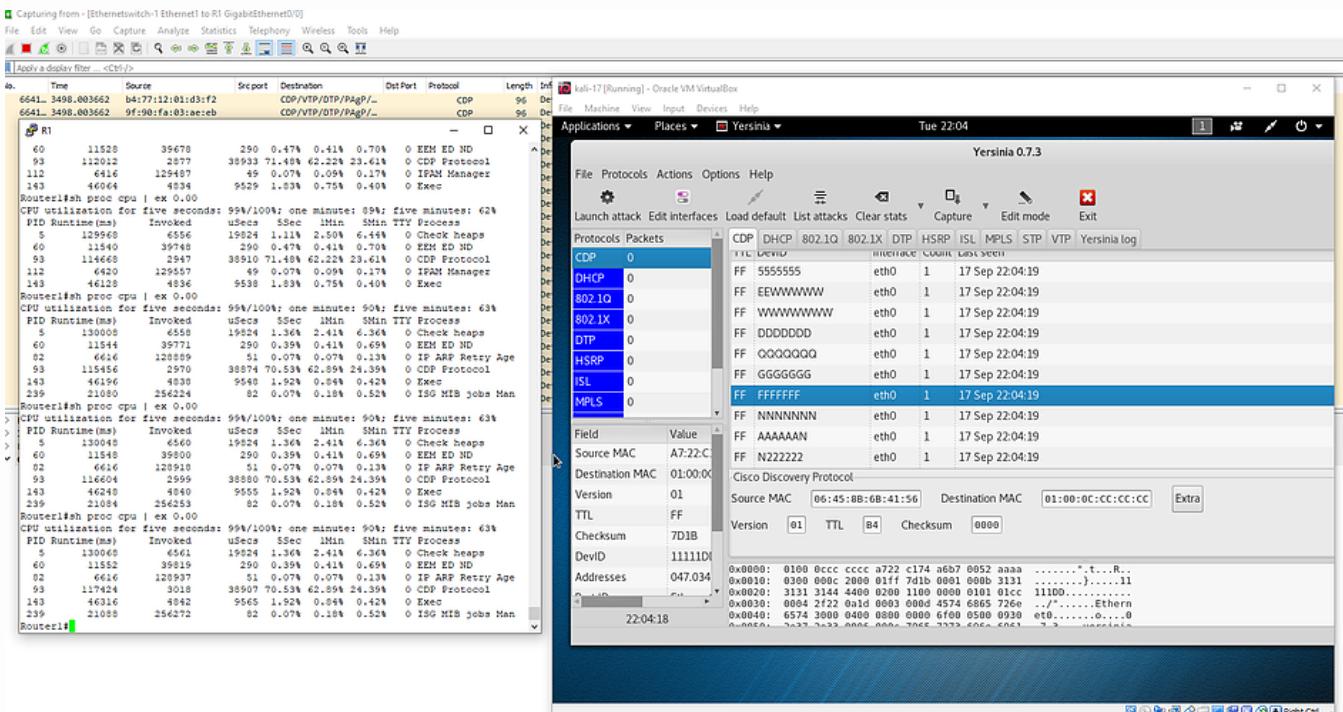
> IEEE 802.3 Ethernet
> Logical-Link Control
> Cisco Discovery Protocol
  Version: 2
  TTL: 180 seconds
  Checksum: 0xd005 [correct]
  [Checksum Status: Good]
  > Device ID: Router1.networkdefenseblog.com
> Software Version
  Type: Software version (0x0005)
  Length: 251
  Software version: Cisco IOS Software, 7200 Software (C7200-ADVENTERPRISEK9-M), Version 15.2(4)S2, RELEASE SOFTWARE (fc1)
  Software version: Technical Support: http://www.cisco.com/techsupport
  Software version: Copyright (c) 1986-2012 by Cisco Systems, Inc.
  Software version: Compiled Tue 11-Dec-12 13:32 by prod_rel_team
> Platform: Cisco 7200VXR
> Addresses
> Port ID: GigabitEthernet1/0
> Capabilities
> Duplex: Full
> Management Addresses
  Type: Management Address (0x0016)
  Length: 17
  Number of addresses: 1
  > IP address: 192.168.100.1

```

### CDP Packet, Notice Make, Model, IP address, Software version

The flood was easy to execute using yersinia. After about a minute or 2 the router was hosed at 100% CPU and then started throwing random logging messages (see below). Remember this is a virtual router under no load, so I'm sure one of your average production devices under some load would have faster results.

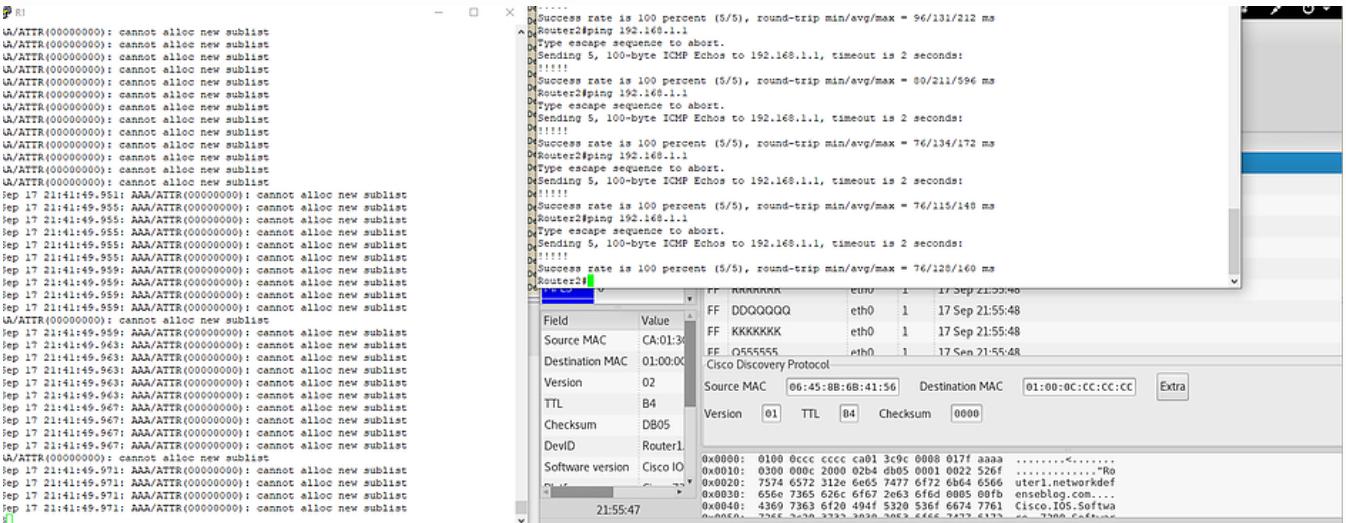
The defensive measure is to go under the interfaces and disable CDP or LLDP so its stops broadcasting and so the device stops processing the requests. Although I did see the overall CPU slightly elevated about 5-10% after disabling CDP both globally or on the interface while still running the attack, the CPU did not increment up to 100% which is good. However I did see the 5 second CPU spike a few times. I speculate this is due to the amount of broadcasts being seen on the interface since the flood attack has some volume.



CDP Layer 2 Flood Attack with Yersinia, also the random CDP messages being sent

Ping tests from one of the other routers showed responses from the victim at around 150+ milliseconds, and the router took a minute to recover even after I stopped sending the CDP messages and disabled CDP globally.

Notice the log entries in the left terminal screen after running the script for a few minutes.



### CDP Layer 2 Flood Attack with Yersinia 2

## Internal Routing Protocols

The next step is to stop sending IGP (EIGRP, RIP, OSPF) multicast messages on your guest or user facing LANs. The easiest way is to *passive-interface default* which is supported by most vendors in my research and then no-passive the interfaces you need to have a neighbor relationship on. This will prevent the messages from being sent out that link and also prevent neighbors from forming. Moreover, as you've probably read before use cryptographic authentication like MD5/SHA to minimize the chance of forming neighbors with rogue routers.

**Beware**, do not perform this mitigation on active production L3 devices unless you have a "commit confirm" type of device, can execute via a script, or have local console access etc. This is because your protocol adjacency will bounce when implemented which could kick you out :) - depending how you're connecting.

First we look at the OSPF multicast Hello packet on the local LAN to begin our reconnaissance. You'll see common attributes within the packet shown.

Capturing from - [Ethernetswitch-1 Ethernet1 to R1 GigabitEthernet0/0]						
No.	Time	Source	Src port	Destination	Dst Port	Protocol
1	0.000000	192.168.100.1	224.0.0.5			OSPF
2	0.762238	ca:01:3c:9c:00:08		ca:01:3c:9c:00:08		LOOP
3	12.301638	192.168.100.1	224.0.0.5			OSPF
4	13.547196	ca:01:3c:9c:00:08		ca:01:3c:9c:00:08		LOOP
5	13.955166	ca:01:3c:9c:00:08		CDP/VTPL/PTP/PoGP/-		CDP
6	24.614712	192.168.100.1	224.0.0.5			OSPF
7	26.288181	ca:01:3c:9c:00:08		ca:01:3c:9c:00:08		LOOP
8	36.475549	192.168.100.1	224.0.0.5			OSPF
9	39.062069	ca:01:3c:9c:00:08		ca:01:3c:9c:00:08		LOOP
10	48.628931	192.168.100.1	224.0.0.5			OSPF
11	51.031994	ca:01:3c:9c:00:08		ca:01:3c:9c:00:08		LOOP
12	59.939347	192.168.100.1	224.0.0.5			OSPF

```
> Frame 1: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface 0
> Ethernet II, Src: ca:01:3c:9c:00:08 (ca:01:3c:9c:00:08), Dst: IPv4mcast_05 (01:00:5e:00:00:05)
> Internet Protocol Version 4, Src: 192.168.100.1, Dst: 224.0.0.5
```

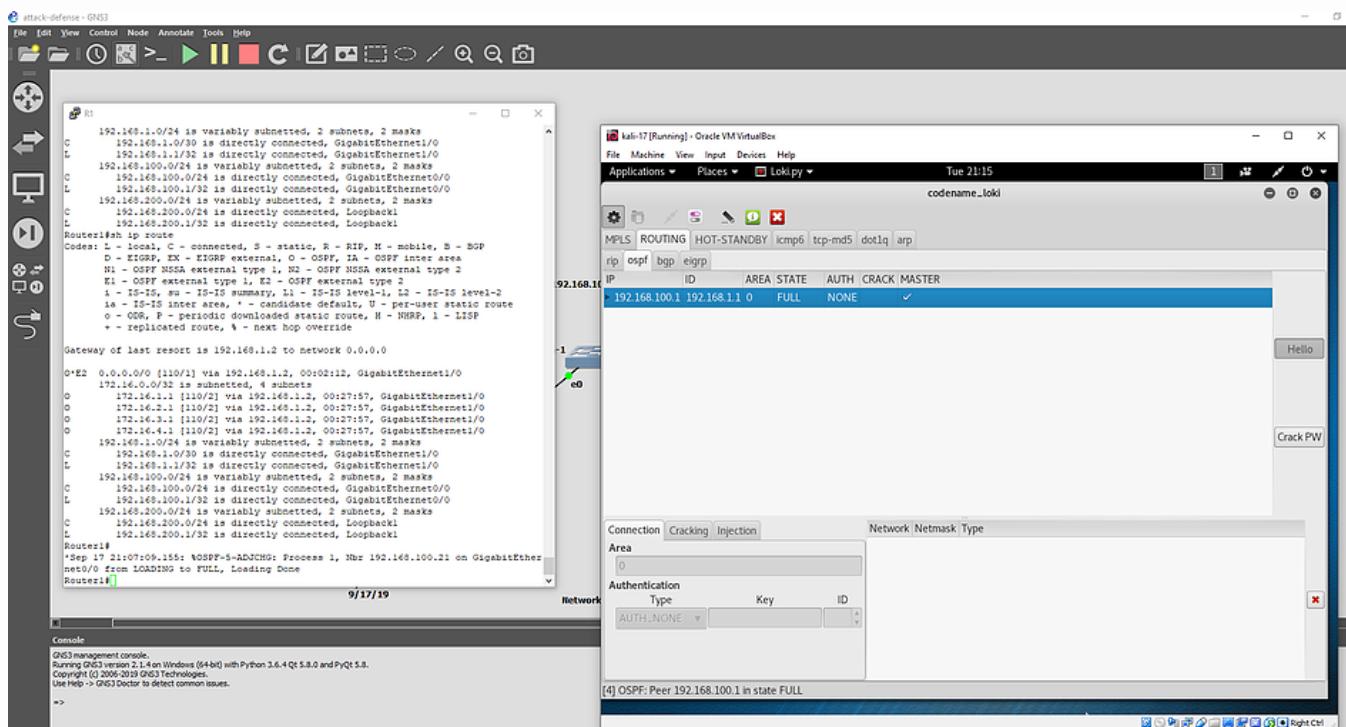
```

    v Open Shortest Path First
    v OSPF Header
        Version: 2
        Message Type: Hello Packet (1)
        Packet Length: 44
        Source OSPF Router: 192.168.1.1
        Area ID: 0.0.0.0 (Backbone)
        Checksum: 0x064b [correct]
        Auth Type: Null (0)
        Auth Data (none): 0000000000000000
    v OSPF Hello Packet
        Network Mask: 255.255.255.0
        Hello Interval [sec]: 10
        > Options: 0x12, (L) LLS Data block, (E) External Routing
        Router Priority: 1
        Router Dead Interval [sec]: 40
        Designated Router: 192.168.100.1
        Backup Designated Router: 0.0.0.0
    v OSPF LLS Data Block
        Checksum: 0xffff6
        LLS Data Length: 12 bytes
        > Extended options TLV

```

OSPF Hello Packet, Notice the Area ID, DR priority, and Hello interval

I mentioned using the program Loki earlier and that it was old. I recall when I first used it a few years ago I was able to perform a malicious route injection pretty easily, but for some reason this time I could not get it to work. It could be because my GNS3/Virtual box has changed and is on a newer version or because my IOS is different (I think I had an older IOS last time). So your mileage may vary with this, plus I noticed the new 0.3.0 version did not have OSPF anymore like the older 0.2.4 version. Nevertheless, it's probably easier to just bridge your NIC adapter into GNS3 and have a virtual router perform this attack vs the showcased programs. Plus, you'd have more features at your disposal using that method.



Forming an OSPF Adjacency to gather info and perform a routing attack

We establish a neighbor-ship with the router first to learn the routing table. The concept would be to inject certain subnets to have them route to your location, or perform the holy grail of IGP Denial of Service attacks and inject a preferred default route into the routing table.

Moving on, a best practice with EIGRP for remote location routers (think hub and spoke) is to limit the query domain using the "stub" feature. This feature can also help prevent malicious route injection because even with a connection to a rogue router, the stub router will not forward the routes learned from it due to the behavior of the stub feature (you can configure it to forward, but by default only connected and summary routes are sent).

configure it to forward, but by default only connected and summary routes are sent.

Therefore, the upstream WAN or core will not learn of the bad path.

Similarly with OSPF a common practice is to use "stub areas" for remote locations or non-core routers which could help mitigate this malicious route injection. This is due to Type-5 LSAs not being generated inside stub areas, so the DoS would likely only effect the stub area (if at all). You'd probably have to generate a Type-3 LSA default route in order to propagate within the stub area, which would still not be preferred inside the backbone (Area 0).

Sequence Number	Link ID	Data	Type	Metric	Description
463	1560.175876	192.168.100.1		192.168.100.21	OSPF 78 DB Description
464	1561.143942	192.168.100.21		192.168.100.1	OSPF 66 DB Description
465	1561.151762	192.168.100.1		192.168.100.21	OSPF 178 DB Description
466	1562.146892	192.168.100.21		192.168.100.1	OSPF 86 DB Description[Malformed Packet]
467	1562.150808	192.168.100.1		192.168.100.21	OSPF 78 DB Description
468	1562.794299	192.168.100.1		224.0.0.5	OSPF 122 LS Update
469	1562.805041	192.168.100.1		224.0.0.5	OSPF 94 LS Update
470	1562.987541	192.168.100.1		224.0.0.5	OSPF 94 Hello Packet
471	1563.149654	192.168.100.21		192.168.100.1	OSPF 70 LS Request
472	1563.151607	192.168.100.21		192.168.100.1	OSPF 70 LS Request
473	1563.153560	192.168.100.21		192.168.100.1	OSPF 70 LS Request
474	1563.155513	192.168.100.21		192.168.100.1	OSPF 70 LS Request
475	1563.157466	192.168.100.21		192.168.100.1	OSPF 70 LS Request
476	1563.159419	192.168.100.1		192.168.100.21	OSPF 122 LS Update
477	1563.159419	192.168.100.1		192.168.100.21	OSPF 146 LS Update
478	1563.159419	192.168.100.1		192.168.100.21	OSPF 110 LS Update
479	1563.159419	192.168.100.1		192.168.100.21	OSPF 94 LS Update
480	1563.159419	192.168.100.1		192.168.100.21	OSPF 98 LS Update
481	1564.159444	192.168.100.21		192.168.100.1	OSPF 98 LS Update

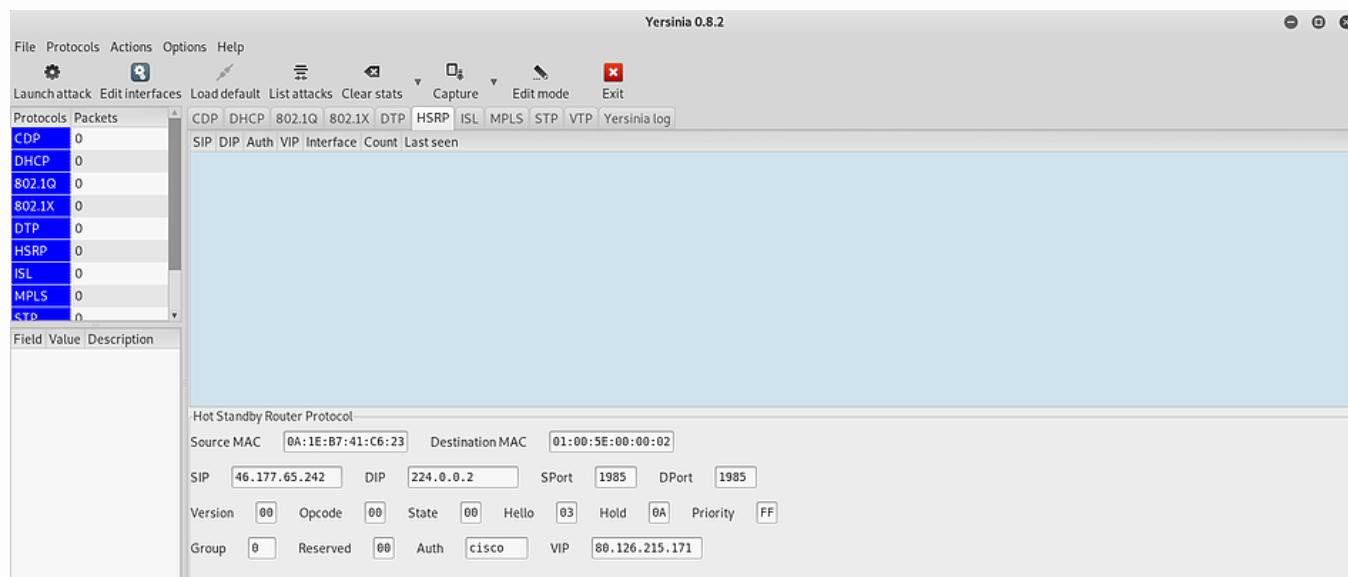
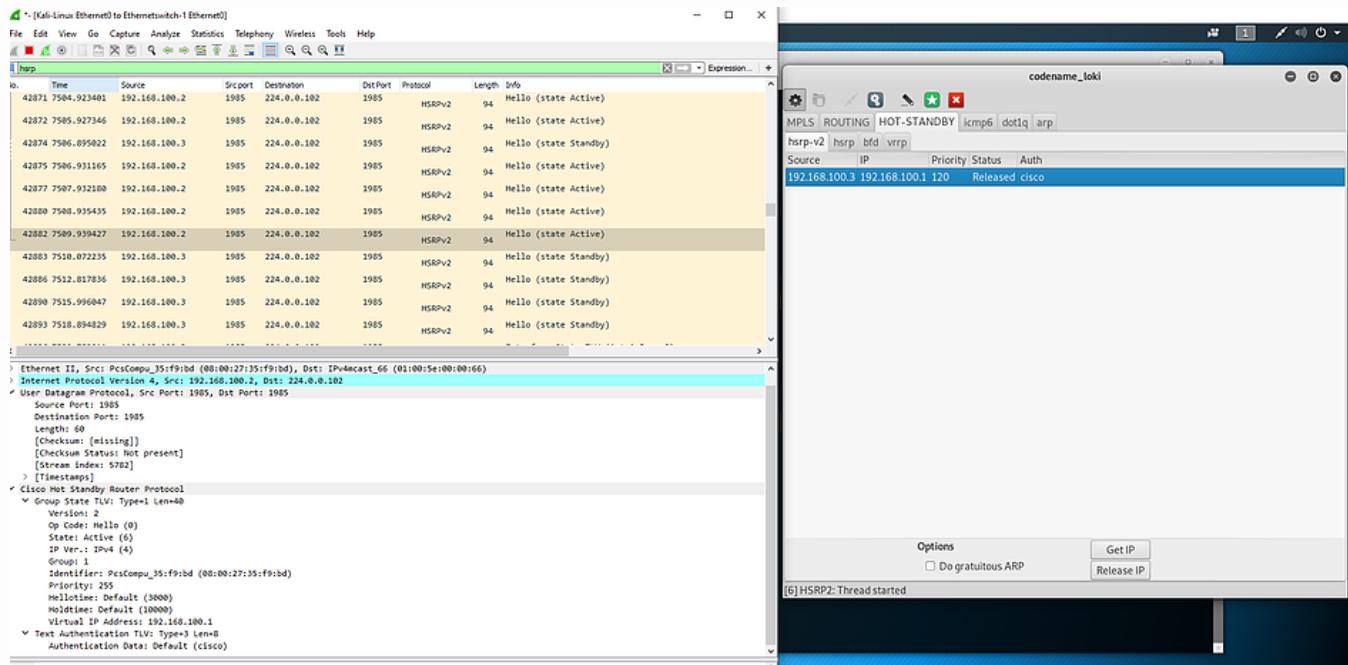
```
> Flags: 0x02, (E) AS boundary router
  Number of Links: 5
  ▾ Type: Stub    ID: 172.16.1.1      Data: 255.255.255 Metric: 1
    Link ID: 172.16.1.1 - IP network/subnet number
    Link Data: 255.255.255.255
    Link Type: 3 - Connection to a stub network
    Number of Metrics: 0 - TOS
    0 Metric: 1
  ▾ Type: Stub    ID: 172.16.2.1      Data: 255.255.255 Metric: 1
    Link ID: 172.16.2.1 - IP network/subnet number
    Link Data: 255.255.255.255
    Link Type: 3 - Connection to a stub network
    Number of Metrics: 0 - TOS
    0 Metric: 1
  ▾ Type: Stub    ID: 172.16.3.1      Data: 255.255.255 Metric: 1
    Link ID: 172.16.3.1 - IP network/subnet number
    Link Data: 255.255.255.255
    Link Type: 3 - Connection to a stub network
    Number of Metrics: 0 - TOS
    0 Metric: 1
  ▾ Type: Stub    ID: 172.16.4.1      Data: 255.255.255 Metric: 1
    Link ID: 172.16.4.1 - IP network/subnet number
    Link Data: 255.255.255.255
    Link Type: 3 - Connection to a stub network
    Number of Metrics: 0 - TOS
    0 Metric: 1
  ▾ Type: Transit ID: 192.168.1.2      Data: 192.168.1.2      Metric: 1
```

OSPF Database Exchange

Finally, you can see the OSPF database exchange after becoming neighbors. Notice the routes being exchanged and shown in the packets using wireshark. These routes were displayed in the routing table of our victim router in the second screen shot of this section. If one would have chosen to inject a bad route - as soon as this process was completed - if done correctly the bad route would have propagated immediately.

A raider could use this information to help minimize excessive discovery scanning on the target network, or use it to hone in on certain IPv4 subnets to selectively scan. However, if cryptographic authentication or the passive-interface command was implemented for this LAN, then the injection wouldn't have been possible.

Loki and yersinia both support HSRP hijacking which would cause black holing of LAN traffic or enable a man-in-the-middle. HSRP is a first-hop redundancy protocol commonly used in scenarios where there are two routers acting as default gateways. The protocol supports MD5 authentication which would help mitigate this vector. Yersinia seemed to have more options but I performed the testing with Loki as MITM attacks were beyond the scope of this lab. 192.168.100.2 is Kali and 100.3 is the router. Note the virtual IP listed inside the packet of 100.1. I just wanted to mention this in order to further highlight the theme of knowing what can be performed when the protocols your router is utilizing are unsecured.



Yersinia's HSRP Hijack options

## Access Control Lists and Services

Use Access Control Lists (ACLs) to protect router management access and access to services on your network devices. Use them for accessing SSH, FTP, SCP, HTTPS etc. Use

the method of least privilege and only allow those certain management subnets that need access. Restricting access greatly reduces the attack surface and is very easy to implement from the basic home routers to the large service provider boxes (please tell me you have ACLs protecting management access!).

Below are 2 screenshots of a Nmap scan of a router with a before and after implementing an ACL to protect some of the services. Basically a scan was performed with the main services enabled and the second is with ACLs implemented on the VTY lines, NTP, SNMP, and HTTP services. I did perform many different scans to identify the open ports and enabled services which took minutes or less each to perform, so this illustrates how quick the attack can develop from recon to exploitation once an unprotected appliance is found.

```

Scan Tools Profile Help
Target: 192.168.100.1
Profile: Intense scan, no ping
Command: nmap -T4 -A -v -Pn 192.168.100.1
Hosts Services Nmap Output Ports/Hosts Topology Host Details Scans
OS Host OS 192.168.100.1
Initiating NSE at 14:26
Completed NSE at 14:26, 41.19s elapsed
Initiating NSE at 14:26
Completed NSE at 14:27, 60.05s elapsed
Initiating NSE at 14:27
Completed NSE at 14:27, 0.00s elapsed
Nmap scan report for 192.168.100.1
Host is up (0.01s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          Cisco SSH 1.25 (protocol 1.5)
| ssh-hostkey:
|_ 2048 2d:b2:ea:65:0f:a0:dd:6e:82:43:2d:a5:2f:a4:ee:d8 (RSA1)
|_sshv1: Server supports SSHv1
23/tcp    open  telnet        Cisco router telnetd
80/tcp    open  http          Cisco IOS http config
|_http-auth:
|_HTTP/1.1 401 Unauthorized
|_ Basic realm=level_15_access
|_http-methods:
|_ Supported Methods: POST
|_http-server-header: cisco-IOS
|_http-title: Site doesn't have a title.
443/tcp   open  ssl/https?
MAC Address: CA:01:2B:18:00:1C (Unknown)
Device type: router
Running: Cisco IOS 12.X
OS CPE: cpe:/hi:cisco:7600_router cpe:/o:cisco:ios:12.2
OS details: Cisco 7600 Router (IOS 12.2)
Network-Delay: 0.000 ms
TCP Sequence Prediction: Difficulty=258 (Good luck!)
IP ID Sequence Generation: Randomized
Service Info: OS: IOS; Device: router; CPE: cpe:/o:cisco:ios

```

Nmap Scan before implementing ACLs and disabling services

```

Scanning 192.168.100.1 [100 ports]
Discovered open port 443/tcp on 192.168.100.1
Discovered open port 22/tcp on 192.168.100.1
Discovered open port 23/tcp on 192.168.100.1
Discovered open port 80/tcp on 192.168.100.1
Completed SYN Stealth Scan at 20:37, 2.28s elapsed (100 total ports)
Initiating UDP Scan at 20:37
Scanning 192.168.100.1 [100 ports]
Discovered open port 123/udp on 192.168.100.1
Discovered open port 161/udp on 192.168.100.1
Completed UDP Scan at 20:37, 1.71s elapsed (100 total ports)

```

Nmap results showing open ports on the Cisco router

```

Scan Tools Profile Help
Target: 192.168.100.1
Profile: Intense scan, no ping
Command: nmap -T4 -A -v -Pn 192.168.100.1
Hosts Services Nmap Output Ports/Hosts Topology Host Details Scans
OS Host OS 192.168.100.1
Initiating ARP Ping Scan at 14:36
Scanning 192.168.100.1 [1 port]
Completed ARP Ping Scan at 14:36, 0.06s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 14:36
Completed Parallel DNS resolution of 1 host. at 14:36, 13.00s elapsed
Initiating SYN Stealth Scan at 14:36
Scanning 192.168.100.1 [1000 ports]
Completed SYN Stealth Scan at 14:36, 2.02s elapsed (1000 total ports)

```

```

Completed SYN Stealth Scan at 14:39, 2.655 elapsed (1000 total ports)
Initiating Service scan at 14:36
Initiating OS detection (try #1) against 192.168.100.1
Retrying OS detection (try #2) against 192.168.100.1
NSE: Script scanning 192.168.100.1.
Initiating NSE at 14:36
Completed NSE at 14:36, 0.00s elapsed
Initiating NSE at 14:36
Completed NSE at 14:36, 0.00s elapsed
Initiating NSE at 14:36
Completed NSE at 14:36, 0.00s elapsed
Nmap scan report for 192.168.100.1
Host is up (0.014s latency).
All 1000 scanned ports on 192.168.100.1 are closed
MAC Address: CA:01:28:18:00:1C (Unknown)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

TRACEROUTE
HOP RTT      ADDRESS
1  14.26 ms  192.168.100.1

```

Nmap scan showing 0 results after implementing ACLs and disabling services

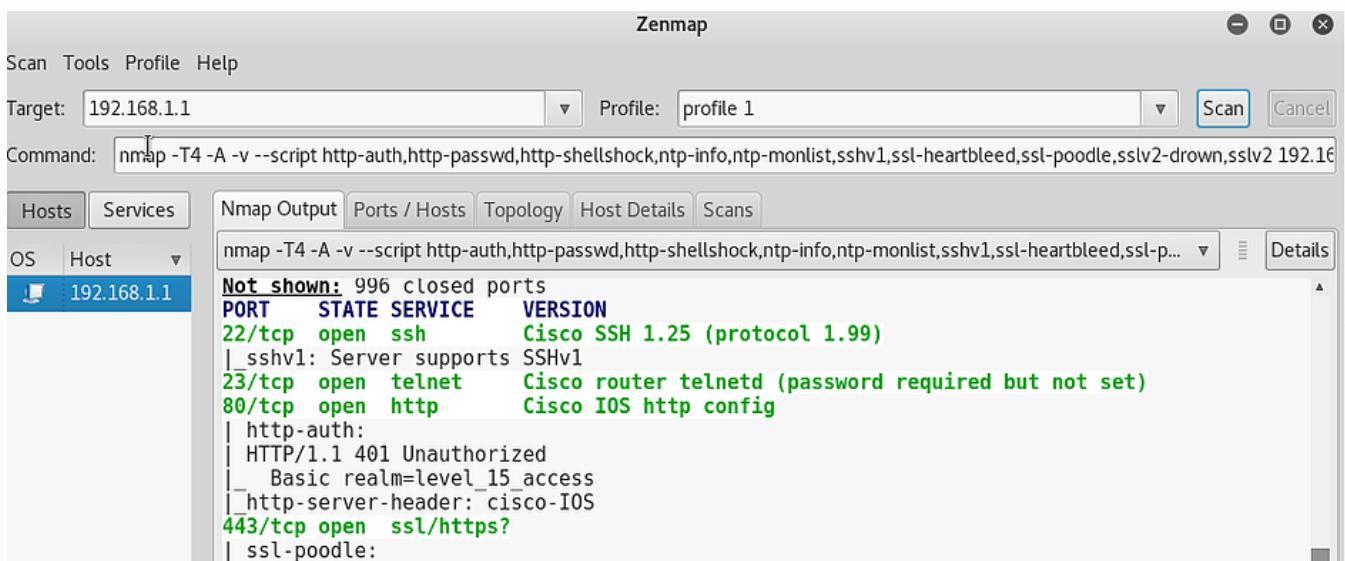
Also consider reviewing your router configurations to ensure there are no unnecessary services enabled and that the services that are enabled are protected. Do you use the web GUI at all? or is there a need to have the HTTP server enabled? if not then consult your vendor's configuration guide to disable it, simple.

I noticed a few of the recent CVEs being released for Cisco IOS are [HTTP based attacks](#), therefore having that service disabled mitigates the risk for those particular vulnerabilities - same thing applies for other exploits against their corresponding vulnerabilities. By having exposed services (i.e. open ports) it allows the attacker to pick and choose and attempt different methods of offense.

Consider which protocols are being used for your routers and switches as well - perhaps you need to change them. For instance, SCP and SFTP are the preferred secure protocols for downloading configurations while most people I see still use TFTP and FTP to perform these actions (even over the internet). TFTP and FTP are cleartext and are less secure than their SSH-based counterparts.

Likewise ensure for your RSA keys you are generating have strong key lengths like 2048. Its been well known keys under 1024 can easily be cracked for a while now. Enable SSH version 2 and set a max number of retries in addition to logging failed logins and setting a minimum accepted key size. This is an example of securing a service you use on your network device.

Here is a Nmap that was ran against a router with weak ciphers and SSH/HTTPS on defaults. I selected a bunch of scripts to check for vulnerable cryptography and such. Note: the router didn't have an ACL applied.



PORT	STATE	SERVICE	VERSION
22/tcp	open	ssh	Cisco SSH 1.25 (protocol 1.99)
		[_sshv1:	Server supports SSHv1
23/tcp	open	telnet	Cisco router telnetd (password required but not set)
80/tcp	open	http	Cisco IOS http config
		http-auth:	
		HTTP/1.1 401 Unauthorized	
		_ Basic realm=level_15_access	
		_http-server-header: cisco-IOS	
443/tcp	open	ssl/https?	
		ssl-poodle:	

```
VULNERABLE:  
SSL POODLE information leak  
State: VULNERABLE  
IDs: CVE-2014-3566 OSVDB:113251  
The SSL protocol 3.0, as used in OpenSSL through 1.0.1i and  
other products, uses nondeterministic CBC padding, which makes it easier  
for man-in-the-middle attackers to obtain cleartext data via a  
padding-oracle attack, aka the "POODLE" issue.  
Disclosure date: 2014-10-14  
Check results:  
TLS_RSA_WITH_3DES_EDE_CBC_SHA  
References:  
https://www.imperialviolet.org/2014/10/14/poodle.html  
http://osvdb.org/113251  
https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-3566  
https://www.openssl.org/~bodo/ssl-poodle.pdf
```

Filter Hosts

Nmap showing SSL vulnerability on a Cisco router

Some other services that are typically disabled are source routing, proxy arp, and IP redirects. Source routing would allow a end host to choose its routed path, proxy arp lets hosts communicate to other IP ranges without a default gateway, or if their subnet mask is configured in correctly. Moreover, redirects not only send [ICMP packets](#) but also notify the end host of a better path - which you should have already selected based on your topology.

A side note as well is to secure your Auxiliary/Modem or AUX ports (Cisco = "line aux 0") Because even with the VTY lines protected you need to apply a similar configuration like an ACL or "*transport input none*" in order to protect or disable the ports 6001 and 9001 etc. which I have seen still open on different models of Cisco devices. By having that service open it could provide telnet or other access leaving you vulnerable to potential offensive action. A good practice is to scan your router and see if there are any non-standard ports open.

---

## ICMP

One point I want to bring to your attention is the information Internet control message protocol (ICMP) can give away. Essentially depending on the situation the router will send ICMP replies letting the sending device know certain information. On some vendor's routers you have the opportunity to disable *ip unreachables* (or type 3 codes) on certain interfaces of your choosing so the device will not send such replies. Though, from my research you need to be tactical on where you disable these [ICMP type 3 replies](#).

For example Path MTU discovery depends on an ICMP type 3 code 4 message to let the end host know fragmentation is needed. Furthermore, type 3 messages are used for traceroute - destination port unreachable (code 3). So by placing *no ip unreachables* on the end destination of the traceroute you will break it, however if you place it on the local LAN interface to protect the first hop device or on intermediate hops traceroute should still work in most cases. Same thing goes for disabling unreachables on an interface that has a lower MTU other than the default 1500 which would need to send the code 4 messages. Therefore, you need to think about which interfaces it would be best to disable this feature on.

As far as I know it is possible for an excessive amount of ICMP unreachables to cause a DoS condition due to excessive CPU usage. So consider rate-limiting them either with control plane policing, specific commands (e.g. "ip icmp rate-limit unreachable <#>"), or by disabling it all together on an interface. ([check this, old but educational article on ICMP crafted messages](#))

Proceeding on, if you are using an ACL to restrict access or for traffic in/out of an interface,

when a block entry is hit the device will respond with a type 3 code 13 "Communication administratively prohibited". This might be unwanted behavior as an attacker could utilize this to their advantage to find open ports or certain types of allowed flows. Here are the results of the first ping test which is dropped via an access-list.

No.	Time	Source	Src port	Destination	Dst Port	Protocol	Length	Info
193	598.997703	192.168.100.1		192.168.100.2		ICMP	78	Destination unreachable (Communication administratively filtered)
194	599.998810	192.168.100.1		192.168.100.1		ICMP	98	Echo (ping) request id=0x0771, seq=3/768, ttl=64 (no response found!)
195	599.995663	192.168.100.1		192.168.100.2		ICMP	78	Destination unreachable (Communication administratively filtered)
196	601.030067	192.168.100.2		192.168.100.1		ICMP	98	Echo (ping) request id=0x0771, seq=4/1024, ttl=64 (no response found!)
197	601.036903	192.168.100.1		192.168.100.2		ICMP	78	Destination unreachable (Communication administratively filtered)
198	602.030419	192.168.100.2		192.168.100.1		ICMP	98	Echo (ping) request id=0x0771, seq=5/1280, ttl=64 (no response found!)
199	602.033365	192.168.100.1		192.168.100.2		ICMP	78	Destination unreachable (Communication administratively filtered)
200	603.031381	192.168.100.2		192.168.100.1		ICMP	98	Echo (ping) request id=0x0771, seq=6/1536, ttl=64 (no response found!)
201	603.053843	192.168.100.1		192.168.100.2		ICMP	78	Destination unreachable (Communication administratively filtered)
202	603.170047	PcsCompu_35:f9:bd	ca:01:2b:18:00:1c			ARP	60	I who has 192.168.100.1? Tell 192.168.100.2
203	603.182742	ca:01:2b:18:00:1c				ARP	60	192.168.100.1 is at ca:01:2b:18:00:1c
204	609.890350	192.168.100.1		224.0.0.5		OSPF	98	Hello Packet
205	621.870752	192.168.100.1		224.0.0.5		OSPF	98	Hello Packet
206	633.528158	192.168.100.1		224.0.0.5		OSPF	98	Hello Packet

```
> Internet Protocol Version 4, Src: 192.168.100.2, Dst: 192.168.100.2
└─ Internet Control Message Protocol
  └─ Type: 3 (Destination unreachable)
    └─ Code: 13 (Communication administratively filtered)
      └─ Checksum: 0x78e4 [correct]
        └─ [Checksum Status: Good]
        └─ Unused: 00000000
  ─ Internet Protocol Version 4, Src: 192.168.100.2, Dst: 192.168.100.1
    ── 0100 .... = Version: 4
    ── 0101 = Header Length: 20 bytes (5)
    ── Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    ── Total Length: 84
    ── Identification: 0xb4b6 (46262)
    ── Flags: 0x4000, Don't fragment
    ── Time to live: 63
    ── Protocol: ICMP (1)
    ── Header checksum: 0x3d9e [validation disabled]
    ── [Header checksum status: Unverified]
    ── Source: 192.168.100.2
    ── Destination: 192.168.100.1
  ─ Internet Control Message Protocol
    ── Type: 8 (Echo (ping) request)
    ── Code: 8
    ── Checksum: 0x7497 [unverified] [in ICMP error packet]
    ── [Checksum Status: Unverified]
    ── Identifier (BE): 1905 (0x0771)
    ── Identifier (LE): 28935 (0x7107)
    ── Sequence number (BE): 6 (0x0006)
    ── Sequence number (LE): 1536 (0x0600)
```

ICMP code 13 packet after pinging traffic restricted by an Access Control List

As mentioned you can disable the ICMP type 3 replies on an interface, which might be a good consideration for a edge router that typically has an inbound access list screening some of the internet traffic as it reaches the network.

Below you can observe that after disabling unreachables there are no ping responses in the second ping test, so scanners would likely not even detect there is a device active if their traffic was dropped; however in the previous scenario the scanner would detect there was a device there even if its traffic was dropped by the access list because of the ICMP reply.

No.	Time	Source	Src port	Destination	Dst Port	Protocol	Length	Info
276	706.228288	192.168.100.1	57691	192.168.100.2	33435	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
277	706.228288	192.168.100.1	33191	192.168.100.2	33436	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
278	706.229239	192.168.100.2	43457	192.168.100.1	53	DNS	86	Standard query 0x6ccb PTR 1.100.168.192.in-addr.arpa
279	706.249757	192.168.100.2	52192	192.168.100.2	33437	ICMP	70	Destination unreachable (Port unreachable)
280	706.249757	192.168.100.2	41843	192.168.100.2	33438	ICMP	70	Destination unreachable (Port unreachable)
281	706.249757	192.168.100.2	49996	192.168.100.2	33439	ICMP	70	Destination unreachable (Port unreachable)
282	706.249757	192.168.100.2	50841	192.168.100.2	33440	ICMP	70	Destination unreachable (Port unreachable)
283	706.249757	192.168.100.2	49134	192.168.100.2	33441	ICMP	70	Destination unreachable (Port unreachable)
284	706.249757	192.168.100.2	50952	192.168.100.2	33442	ICMP	70	Destination unreachable (Port unreachable)
285	706.249757	192.168.100.2	49307	192.168.100.2	33443	ICMP	70	Destination unreachable (Port unreachable)
286	706.250725	192.168.100.2	49050	192.168.100.2	33444	ICMP	70	Destination unreachable (Port unreachable)
287	706.250725	192.168.100.2	44194	192.168.100.2	33445	ICMP	70	Destination unreachable (Port unreachable)
288	706.250725	192.168.100.2	46448	192.168.100.2	33446	ICMP	70	Destination unreachable (Port unreachable)
289	706.250725	192.168.100.2	51544	192.168.100.2	33447	ICMP	70	Destination unreachable (Port unreachable)
290	706.250725	192.168.100.2	55893	192.168.100.2	33448	ICMP	70	Destination unreachable (Port unreachable)
291	711.230490	192.168.100.2	43457	192.168.100.1	53	DNS	86	Standard query 0x6ccb PTR 1.100.168.192.in-addr.arpa
293	711.404331	PcsCompu_35:f9:bd	ca:01:2b:18:00:1c			ARP	60	I who has 192.168.100.1? Tell 192.168.100.2
294	711.423867	ca:01:2b:18:00:1c				ARP	60	192.168.100.1 is at ca:01:2b:18:00:1c
295	716.281588	192.168.100.2	40046	172.16.1.1	33450	UDP	70	40046 + 33450 Len=32
296	716.281588	192.168.100.2	55997	192.168.100.1	35	DNS	84	Standard query 0x6ccb PTR 2.1.100.168.192.in-addr.arpa
297	716.366083	192.168.100.2	40046	192.168.100.2	33450	ICMP	70	Destination unreachable (Port unreachable)
298	717.034902	192.168.100.1	22470.0.3			OSPF	98	Hello Packet
299	717.937197	CDP/VTTP/DTP/PAGP/-				CDP	397	Device ID: Router1.com Port ID: GigabitEthernet1/0
300	721.307980	192.168.100.2	55947	192.168.100.1	53	DNS	84	Standard query 0x60d1 PTR 2.1.100.168.192.in-addr.arpa
301	728.585581	192.168.100.1	224.0.0.5			OSPF	98	Hello Packet
302	736.506846	192.168.100.2		192.168.100.1		ICMP	98	Echo (ping) request id=0x0778, seq=1/256, ttl=64 (no response found!)
303	737.598546	192.168.100.2		192.168.100.1		ICMP	98	Echo (ping) request id=0x0778, seq=2/512, ttl=64 (no response found!)
304	738.622421	192.168.100.2		192.168.100.1		ICMP	98	Echo (ping) request id=0x0778, seq=3/768, ttl=64 (no response found!)

First traceroute

Second traceroute

Second ping test

```

305 739.645522 192.168.100.2      192.168.100.1      ICMP      98  Echo (ping) request id=0x0778, seq=4/1024, ttl=64 (no response found!)
306 740.452225 192.168.100.1      224.0.0.5        OSPF      90  Hello Packet
307 740.669026 192.168.100.2      192.168.100.1      ICMP      98  Echo (ping) request id=0x0778, seq=5/1280, ttl=64 (no response found!)
> Frame 276: 78 bytes on wire (600 bits), 78 bytes captured (600 bits) on interface 0
> Ethernet II, Src: cas01:2b:18:00:1c (ca:01:2b:18:00:1c), Dst: PcsCompu_35:f9:bd (08:00:27:35:f9:bd)
> Internet Protocol Version 4, Src: 192.168.100.1, Dst: 192.168.100.2
  Internet Control Message Protocol
    Type: 11 (Time-to-live exceeded)
    Code: 0 (Time to live exceeded in transit)
    Checksum: 0xb0fa [correct]
    [Checksum Status: Good]
    Unused: 00000000
  Internet Protocol Version 4, Src: 192.168.100.2, Dst: 172.16.1.1
    0100 ... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
Type (cmp.type), 1 byte
Packets: 321 • Displayed: 321 (100.0%)
Profile: Default ...

```

Ping and Traceroute tests as viewed in wireshark before and after IP unreachables are disabled

Additionally, notice the difference between the first and second traceroutes where the second traceroute did not have a packet from the first hop router due to the ICMP type 3 replies being disabled. The final hop still had them enabled though.

Here is how the tests looked from the end host perspective. From the top down:

1. Ping without ACL - successful
2. Ping with ACL - Code 13 reply from first hop router
3. Traceroute normal
4. Traceroute with "no IP unreachables" on first hop interface
5. Ping with ACL but with no ICMP replies - unsuccessful

```

root@kali: ~
File Edit View Search Terminal Help
64 bytes from 192.168.100.1: icmp_seq=3 ttl=255 time=9.17 ms
64 bytes from 192.168.100.1: icmp_seq=4 ttl=255 time=6.22 ms
64 bytes from 192.168.100.1: icmp_seq=5 ttl=255 time=5.24 ms
^C
--- 192.168.100.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 1.222/7.451/15.404/4.720 ms
root@kali:~# ping 192.168.100.1
PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data.
From 192.168.100.1 icmp_seq=1 Packet filtered
From 192.168.100.1 icmp_seq=2 Packet filtered
From 192.168.100.1 icmp_seq=3 Packet filtered
From 192.168.100.1 icmp_seq=4 Packet filtered
From 192.168.100.1 icmp_seq=5 Packet filtered
From 192.168.100.1 icmp_seq=6 Packet filtered
^C
--- 192.168.100.1 ping statistics ---
6 packets transmitted, 0 received, +6 errors, 100% packet loss, time 5050ms

root@kali:~# traceroute 172.16.1.1
traceroute to 172.16.1.1 (172.16.1.1), 30 hops max, 60 byte packets
 1 _gateway (192.168.100.1) 21.626 ms 21.644 ms 21.699 ms
 2 192.168.1.2 (192.168.1.2) 31.450 ms 31.390 ms 31.474 ms
root@kali:~# traceroute 172.16.1.1
traceroute to 172.16.1.1 (172.16.1.1), 30 hops max, 60 byte packets
 1 _gateway (192.168.100.1) 14.026 ms 14.002 ms 14.091 ms
 2 192.168.1.2 (192.168.1.2) 34.975 ms 34.893 ms 34.905 ms
root@kali:~# ping 192.168.100.1
PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data.
^C
--- 192.168.100.1 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4084ms

```

## Out of the Box Review

Kali linux has a variety of tools that are shipped with it that can literally exploit and devastate a variety of targets. However in my testing I found the tools had more exploits for Soho routers than for Cisco routers. The programs tested were cisco-torch, cisco global exploit (CGE), and metasploit framework. Moreover, a lot of the vulnerabilities seemed aged based on the listed date (if there was one). This is why its important to keep your software up to date, because as these issues with the code arise the vendors (hopefully) patch and protect against them. You wouldn't want to be taken down from an old vulnerability that was patched years ago because you neglected to update.

but it's been years ago because you're going to update.

Back to the aged scripts, for example the cisco global exploit showed one option to be for catOS which is old and probably wouldn't be seen by most people and it had a few old HTTP CVEs from the early 2000s. I found one funny in that it sent a HTTP GET with a tons of AAAAAAAAs which I assume did something once upon a time. The router responded with a 401 bad request and seemed unchanged after a few attempts (this could also be because it is a virtual device but I doubt it). I'm not really familiar with that CVE either.

HTTP exploit ran against a Cisco Router

Most the assaults I tried with cisco-torch and CGE were ineffective, except for the telnet buffer overflow when left unchecked (but like we discussed - disable unused services, Telnet should not be used because it is not secure). I even tried some NTP vulnerabilities which I expected my IOS to effected by like ntp\_overflow and ntpd\_reserved\_dos but no dice.

This is how the Syslog #9 vulnerability looked from GCE.

```
root@kali: ~
[10] - CiscoSecure ACS for Windows NT Server Denial of Service Vulnerability
[11] - Cisco Catalyst Memory Leak Vulnerability
[12] - Cisco Catos CiscoView HTTP Server Buffer Overflow Vulnerability
[13] - 0 Encoding IDS Bypass Vulnerability (UTF)
[14] - Cisco IOS TFTP Denial of Service Vulnerability
root@kali: ~# cge.pl 192.168.100.1 2
socket sent ...
ow checking server's status ...
ulnerability unsuccessful exploited. Target server is still up ...
root@kali: ~# cge.pl 192.168.100.1 4
socket sent ...
ow checking server's status ...
ulnerability unsuccessful exploited. Target server is still up ...
root@kali: ~# cge.pl 192.168.100.1 8
socket sent ...
erver response :
HTTP/1.1 404 Not Found
date: Sun, 22 Sep 2019 14:45:38 GMT
server: cisco-IOS
Content-Ranges: none

04 Not Found
root@kali: ~# cge.pl 192.168.100.1 5
socket sent ...
root@kali: ~# cge.pl 192.168.100.1 9
input packets size : 1024

ackets sent ...
lease enter a server's open port : 514

ow checking server status ...
ulnerability successful exploited. Target server is down ...

root@kali: ~# cge.pl 192.168.100.1 9
input packets size : 5000

ackets sent ...
lease enter a server's open port : 23

[10] - CiscoSecure ACS for Windows NT Server Denial of Service Vulnerability
[11] - Cisco Catalyst Memory Leak Vulnerability
[12] - Cisco Catos CiscoView HTTP Server Buffer Overflow Vulnerability
[13] - 0 Encoding IDS Bypass Vulnerability (UTF)
[14] - Cisco IOS TFTP Denial of Service Vulnerability
root@kali: ~# cge.pl 192.168.100.1 2
socket sent ...
ow checking server's status ...
ulnerability unsuccessful exploited. Target server is still up ...
root@kali: ~# cge.pl 192.168.100.1 4
socket sent ...
ow checking server's status ...
ulnerability unsuccessful exploited. Target server is still up ...
root@kali: ~# cge.pl 192.168.100.1 8
socket sent ...
erver response :
HTTP/1.1 404 Not Found
date: Sun, 22 Sep 2019 14:45:38 GMT
server: cisco-IOS
Content-Ranges: none

04 Not Found
root@kali: ~# cge.pl 192.168.100.1 5
socket sent ...
root@kali: ~# cge.pl 192.168.100.1 9
input packets size : 1024

ackets sent ...
lease enter a server's open port : 514

ow checking server status ...
ulnerability successful exploited. Target server is down ...

root@kali: ~# cge.pl 192.168.100.1 9
input packets size : 5000

ackets sent ...
lease enter a server's open port : 23

[10] - CiscoSecure ACS for Windows NT Server Denial of Service Vulnerability
[11] - Cisco Catalyst Memory Leak Vulnerability
[12] - Cisco Catos CiscoView HTTP Server Buffer Overflow Vulnerability
[13] - 0 Encoding IDS Bypass Vulnerability (UTF)
[14] - Cisco IOS TFTP Denial of Service Vulnerability
root@kali: ~# cge.pl 192.168.100.1 2
socket sent ...
ow checking server's status ...
ulnerability unsuccessful exploited. Target server is still up ...
root@kali: ~# cge.pl 192.168.100.1 4
socket sent ...
ow checking server's status ...
ulnerability unsuccessful exploited. Target server is still up ...
root@kali: ~# cge.pl 192.168.100.1 8
socket sent ...
erver response :
HTTP/1.1 404 Not Found
date: Sun, 22 Sep 2019 14:45:38 GMT
server: cisco-IOS
Content-Ranges: none

04 Not Found
root@kali: ~# cge.pl 192.168.100.1 5
socket sent ...
root@kali: ~# cge.pl 192.168.100.1 9
input packets size : 1024

ackets sent ...
lease enter a server's open port : 514

ow checking server status ...
ulnerability successful exploited. Target server is down ...

root@kali: ~# cge.pl 192.168.100.1 9
input packets size : 5000

ackets sent ...
lease enter a server's open port : 23

[10] - CiscoSecure ACS for Windows NT Server Denial of Service Vulnerability
[11] - Cisco Catalyst Memory Leak Vulnerability
[12] - Cisco Catos CiscoView HTTP Server Buffer Overflow Vulnerability
[13] - 0 Encoding IDS Bypass Vulnerability (UTF)
[14] - Cisco IOS TFTP Denial of Service Vulnerability
root@kali: ~# cge.pl 192.168.100.1 2
socket sent ...
ow checking server's status ...
ulnerability unsuccessful exploited. Target server is still up ...
root@kali: ~# cge.pl 192.168.100.1 4
socket sent ...
ow checking server's status ...
ulnerability unsuccessful exploited. Target server is still up ...
root@kali: ~# cge.pl 192.168.100.1 8
socket sent ...
erver response :
HTTP/1.1 404 Not Found
date: Sun, 22 Sep 2019 14:45:38 GMT
server: cisco-IOS
Content-Ranges: none

04 Not Found
root@kali: ~# cge.pl 192.168.100.1 5
socket sent ...
root@kali: ~# cge.pl 192.168.100.1 9
input packets size : 1024

ackets sent ...
lease enter a server's open port : 514

ow checking server status ...
ulnerability successful exploited. Target server is down ...

root@kali: ~# cge.pl 192.168.100.1 9
input packets size : 5000

ackets sent ...
lease enter a server's open port : 23
```

```
ow checking server status ...
ulnerability unsuccessful exploited. Target server is still up ... Ready to load or capture Packets: 73619 - Displayed: 73619 (100.0%) Profile: Default
```

### Syslog DoS attempt from CGE

You can see on metasploit there are a lot of options to choose from, but like with the previous examples the attacks I tried didn't do much against a more up to date (although still older IOS) protected device; although I only tested on virtual instances of Cisco routers. I'm sure some of those Soho scripts for D-links or the new Cisco RV small business devices are potent though.

```
msfs > search router
Matching Modules
=====
# Name Disclosure Date Rank Check Description
---- -
0 auxiliary/admin/2wire/xslt_password_reset 2007-08-15 normal No 2Wire Cross-Site Request Forgery Password Reset Vulnerability
1 auxiliary/admin/http/arris_motorola_surfboard_backdoor_xss 2015-04-08 normal No Arris / Motorola Surfboard SBG6580 Web Interface Takeover
2 auxiliary/admin/http/dlink_dir_300_600_exec_noauth 2013-02-04 normal No D-Link DIR-600 / DIR-300 Unauthenticated Remote Command Execution
3 auxiliary/admin/http/linksys_e1500_e2500_exec 2013-02-05 normal No Linksys E1500/E2500 Remote Command Execution
4 auxiliary/admin/http/linksys_tmunblock_admin_reset_bof 2014-02-19 normal No Linksys WRT120N tmunblock Stack Buffer Overflow
5 auxiliary/admin/http/linksys_wrt54gl_exec 2013-01-18 normal No Linksys WRT54GL Remote Command Execution
6 auxiliary/admin/http/netgear_wnr2000_pass_recovery 2016-12-20 normal No NETGEAR WNR2000v5 Administrator Password Recovery
7 auxiliary/admin/http/zxvel_admin_password_extractor 2013-12-31 normal No ZyxEL GS1510-16 Password Extractor
8 auxiliary/admin/misc/sercomm_dump_config 2004-09-24 normal No SerComm Device Configuration Dump
9 auxiliary/admin/motorola/wr850g_cred 2004-09-24 normal No Motorola WR850G v4.03 Credentials
10 auxiliary/onat/bnat_router 2000-04-26 normal No BNAT Router
11 auxiliary/dos/cisco/ios_http_percentpercent 2000-04-26 normal No Cisco IOS HTTP GET /%% Request Denial of Service
12 auxiliary/dos/tcp/juniper_tcp_opt 2000-04-26 normal No Juniper JUNOS Malformed TCP Option
13 auxiliary/gather/cisco_rv320_config 2019-01-24 normal No Cisco RV320/RV325 Configuration Disclosure
14 auxiliary/gather/huawei_wifi_info 2013-11-11 normal No Huawei Datacard Information Disclosure Vulnerability
15 auxiliary/gather/netgear_password_disclosure 2013-12-31 normal Yes NETGEAR Administrator Password Disclosure
16 auxiliary/scanner/discovery/ipv6_neighbr_router_advertisement 2014-11-17 normal Yes IPv6 Local Neighbor Discovery Using Router Advertisement
17 auxiliary/scanner/dlsw/dlsw_leak_capture 2014-11-17 normal Yes Cisco DLsW Information Disclosure Scanner
18 auxiliary/scanner/http/cisco_device_manager 2000-10-26 normal Yes Cisco Device HTTP Device Manager Access
19 auxiliary/scanner/http/cisco_ios_auth_bypass 2001-06-27 normal Yes Cisco IOS HTTP Unauthorized Administrative Access
20 auxiliary/scanner/http/linksys_e1500_traversal 2001-06-27 normal Yes Linksys E1500 Directory Traversal Vulnerability
21 auxiliary/scanner/sap/sap_router_info_request 2017-08-31 normal Yes SAPRouter Admin Request
22 auxiliary/scanner/sap/sap_router_portsscanner 2017-08-31 normal No SAPRouter Port Scanner
23 auxiliary/scanner/wproxy/att_open_proxy 2017-08-31 normal Yes Open-WAN-to-LAN proxy on AT&T routers
24 auxiliary/spoof/cisco/cdp 2017-08-31 normal Yes Send Cisco Discovery Protocol (CDP) Packets
25 auxiliary/spoof/dns/native_spoof 2015-03-31 normal No Native DNS Spoof (Example)
26 exploit/linux/http/airties_login.cgi_bof 2014-05-09 normal Yes Airties login.cgi Buffer Overflow
27 exploit/linux/http/belkin_login_bof 2019-02-27 good No Belkin Play N750 login.cgi Buffer Overflow
28 exploit/linux/http/cisco_rv130_rm1_rce 2018-09-09 normal Yes Cisco RV130W Routers Management Interface Remote Command Execution
29 exploit/linux/http/cisco_rv32x_rce 2013-02-08 normal Yes Cisco RV320 and RV325 Unauthenticated Remote Code Execution
30 exploit/linux/http/dlink_authentication.cgi_bof 2013-02-08 normal Yes D-Link authentication.cgi Buffer Overflow
31 exploit/linux/http/dlink_command_php_exec_noauth 2013-02-04 excellent No D-Link Devices Unauthenticated Remote Command Execution
32 exploit/linux/http/dlink_diagnostic_exec_noauth 2013-03-05 excellent No D-Link DIR-645 / DIR-815 diagnostic.php Command Execution
33 exploit/linux/http/dlink_dir300_exec_telnet 2013-04-22 excellent No D-Link Devices Unauthenticated Remote Command Execution
34 exploit/linux/http/dlink_dir605l_captcha_bof 2012-10-08 manual Yes D-Link DIR-605L Captcha Handling Buffer Overflow
35 exploit/linux/http/dlink_dir615_up_exec 2013-02-07 excellent No D-Link DIR615H OS Command Injection
36 exploit/linux/http/dlink_airmax_nac_exec 2017-08-09 excellent Yes DIR-RSMI (Unauthenticated OS Command Exec)
```

```
msfs > search cisco
Matching Modules
=====
# Name Disclosure Date Rank Check Description
---- -
0 auxiliary/admin/cisco/cisco_asa_extrabacon 2006-08-23 normal Yes Cisco ASA Authentication Bypass (EXTRABACON)
1 auxiliary/admin/cisco/cisco_secure_acs_bypass 2015-07-28 normal Yes Cisco Secure ACS Unauthorized Password Change
2 auxiliary/admin/cisco/vpn_3000_ftp_bypass 2006-08-23 normal No Cisco VPN Concentrator 3000 FTP Unauthorized Administrative Access
3 auxiliary/admin/scada/moxa_credentials_recovery 2015-07-28 normal Yes Moxa Device Credential Retrieval
4 auxiliary/admin/smb/webexec_command 2000-04-26 normal Yes WebEx Remote Command Execution Utility
5 auxiliary/dos/cisco/ios_http_percentpercent 2017-03-17 normal No Cisco IOS GET /%% Request Denial of Service
6 auxiliary/dos/cisco/ios_telnet_racm 2019-01-24 normal No Cisco IOS Telnet Denial of Service
7 auxiliary/gather/cisco_rv320_config 2014-11-17 normal No Cisco RV320/RV325 Configuration Disclosure
8 auxiliary/scanner/dlsw/dlsw_leak_capture 2000-10-26 normal Yes Cisco DLsW Information Disclosure Scanner
9 auxiliary/scanner/http/cisco_asa_asdm 2000-10-26 normal Yes Cisco ASA ASDM BruteForce Login Utility
10 auxiliary/scanner/http/cisco_device_manager 2018-06-06 normal No Cisco Device HTTP Device Manager Access
11 auxiliary/scanner/http/cisco_directory_traversal 2016-10-10 normal Yes Cisco File Management Console 6.0 Post Auth Report Download Directory Traversal
12 auxiliary/scanner/http/cisco_firmware_download 2016-10-10 normal Yes Cisco Firmware Management Console 6.0 Login
13 auxiliary/scanner/http/cisco_firepower_login 2001-06-27 normal Yes Cisco IOS HTTP Unauthorized Administrative Access
14 auxiliary/scanner/http/cisco_ios_auth_bypass 2001-06-27 normal Yes Cisco Ironport BruteForce Login Utility
15 auxiliary/scanner/http/cisco_irportntron_enum 2016-09-29 normal Yes Cisco Network Access Manager Directory Traversal Vulnerability
16 auxiliary/scanner/http/cisco_nac_manager_traversal 2014-04-09 normal Yes Cisco SSL VPN BruteForce Login Utility
17 auxiliary/scanner/http/cisco_ssl_vpn 2014-04-09 normal Yes Cisco ASA SSL VPN Privilege Escalation Vulnerability
18 auxiliary/scanner/http/cisco_ssl_vpn_priv_esc 2016-09-29 normal Yes Linksys E1500 Directory Traversal Vulnerability
19 auxiliary/scanner/http/linksys_e1500_traversal 2016-09-29 normal Yes Cisco IKE Information Disclosure
20 auxiliary/scanner/ike/cisco_ike_benigncertain 2016-09-29 normal Yes Identify Cisco Smart Install endpoints
21 auxiliary/scanner/misc/cisco_smart_install 2015-12-08 normal Yes NTP "NAX to the Future"
22 auxiliary/scanner/ntp/ntp_nak_to_the_future 2015-12-08 normal Yes Moxa UDP Device Discovery
23 auxiliary/scanner/scada/moxa_discover 2015-12-08 normal Yes Cisco IOS SNMP Configuration Grabber (TFTP)
24 auxiliary/scanner/snmp/cisco_config_tftp 2015-12-08 normal Yes Cisco IP Camera Upload (TFTP)
25 auxiliary/scanner/smb/cisco_upload_file 2015-12-08 normal Yes Microsoft Windows Media Center MCL Information Disclosure
26 auxiliary/server/cisco_mcl_leak 2015-12-08 normal No SMB Cisco Discovery Protocol (CDP) Packets
27 auxiliary/spoof/cisco/dlp 2015-12-08 normal No Forge Cisco DLP Packets
28 auxiliary/spoof/cisco/dtp 2015-12-08 normal No Viproy CUCM IP Phone XML Services : Call Forwarding Tool
29 auxiliary/voip/cisco_cudcm_call_forward 2015-12-08 normal No Viproy CUCM IP Phone XML Services : Speed Dial Attack Tool
30 auxiliary/voip/cisco_cudcm_speed_dials 2015-12-08 normal No Telitica IPS Lock Cisco IP Phone Control
31 auxiliary/voip/telitica_ip5_lock_control 2015-12-17 excellent Yes Cisco Firepower Management Console 6.0 Post Authentication UserAdd Vulnerability
32 exploit/linux/http/cisco_firepower_useradd 2016-10-10 excellent Yes Cisco Prime Infrastructure Unauthenticated Remote Code Execution
33 exploit/linux/http/cisco_prime_inf_rce 2018-10-04 excellent Yes Cisco Prime Infrastructure Health Monitor TarArchive Directory Traversal Vulnerability
34 exploit/linux/http/cisco_rv130_rm1_rce 2019-02-27 good No Cisco RV130W Routers Management Interface Remote Command Execution
35 exploit/linux/http/cisco_rv32x_rce 2018-09-09 normal Yes Cisco RV320 and RV325 Unauthenticated Remote Code Execution
36 exploit/linux/http/cpi_tararchive_upload 2019-05-15 excellent Yes Cisco Prime Infrastructure Health Monitor TarArchive Directory Traversal Vulnerability
37 exploit/linux/http/sophos_wpa_sblistpack_exec 2018-09-06 excellent Yes Sophos Web Protection Appliance sblistpack Arbitrary Command Execution
38 exploit/linux/local/cpi_rnrunshell_priv_esc 2018-12-08 excellent No Cisco Prime Infrastructure Runrshell Privilege Escalation
39 exploit/linux/local/sophos_wpa_clear_keys 2013-09-06 excellent Yes Sophos Web Protection Appliance clear.keys.pl Local Privilege Escalation
```

### Closing

I hope this post helps you in reviewing your routers or other Layer 3 network devices in order to better protect management access, along with your infrastructure's availability from Denial of Service attempts. Additionally, in the future I'm sure there will be more effective additions to the Kali distribution against enterprise routers vs. small business routers.

In this post we went through some packets and got down to the detailed level for a few of the offensive actions in order to better understand **why** we implement some of these security controls. Disabling unused services that broadcast out like CDP/LLDP and protecting your router using ACLs are simple yet effective means of network defense. If a protocol supports authentication then its wise to use it where possible. Also take into account the implications of leaving ICMP type 3 messages enabled on external facing interfaces.

The next post in this area I will probably do a write up about *Control Plane Policing* to protect the control/data planes, since that topic isn't covered a lot from what I've seen.

**Finally, this should be a given but keep your devices patched and up to date!**

Good luck out there.

Would you like to know more?

[Juniper device hardening checklist](#)

[Comprehensive Cisco configuration guide for hardening devices](#)

[Configuration guide for hardening Cisco IOS-XR devices](#)

[Mikrotik Router OS hardening guide](#)

---

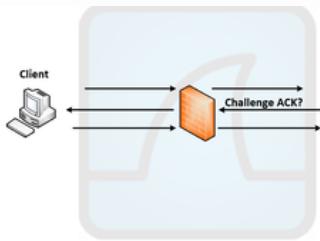
---

Thank you for reading. Please like, share, and join the mailing list!



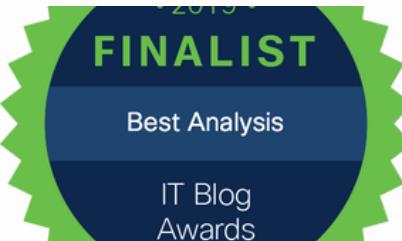
Brandon Hitzel is a network engineer hailing out of California. He holds multiple industry certifications including SSCP, CCDP, and CCNP. He has been in the industry for a number of years and enjoys everything networking, defense, and tech related.

#networkdefense #cybersecurity #kalilinux #wireshark #packetanalysis #networking  
#netsec



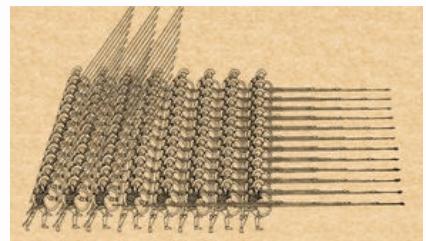
[Troubleshooting with Wireshark:  
The Case of the TCP Challenge ...](#)

[Write a comment](#)



[Cisco IT Blog Award Finalist in  
"Best Analysis" Vote Now!](#)

[Write a comment](#)



[The Art of Cyber War and Cyber  
Battle: Formations](#)

[Write a comment](#)



[Log in to leave a comment.](#)

## Contact Me

Professional | Personal | Consulting | Volunteering

Use the below form to drop me a line

**Join the mailing list**  
Never miss an update

Email Address

Click if you're a Human

[Subscribe Now](#)

Name \*

Email \*

Subject \*

Message \*

[Send](#)



Copy write © 2020 by Brandon Hitzel  
Site Work in Progress - Best viewed on the desktop



