Brandon Hitzel · Sep 25 · 13 min read

5 Easy Router Protection Techniques - includes Attack and Packet Analysis

Updated: Sep 27

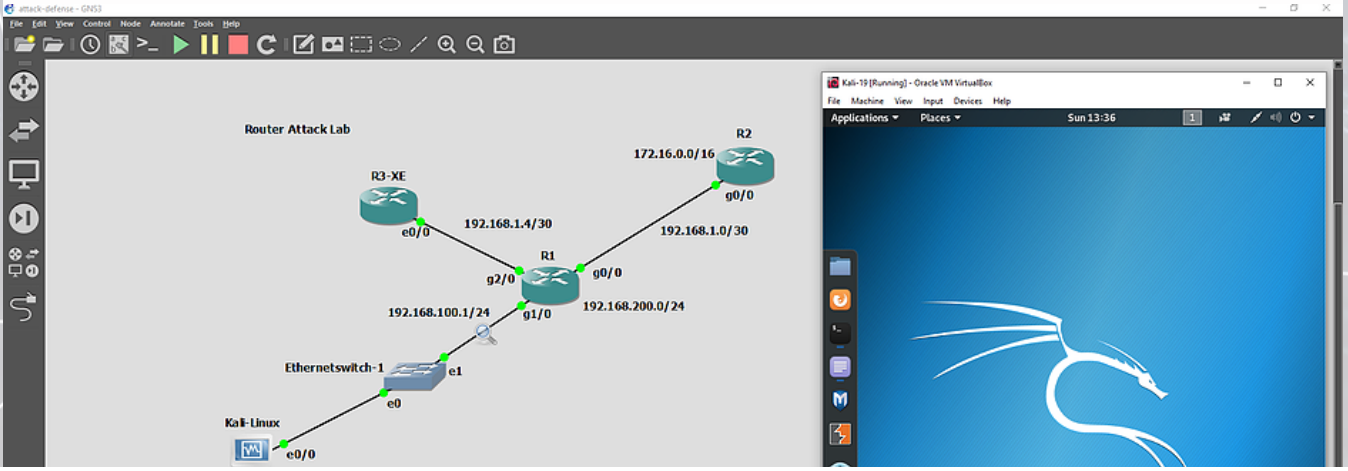
Executing attacks against a router using Kali Linux along with Wireshark and the easy steps to protect against them.

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.

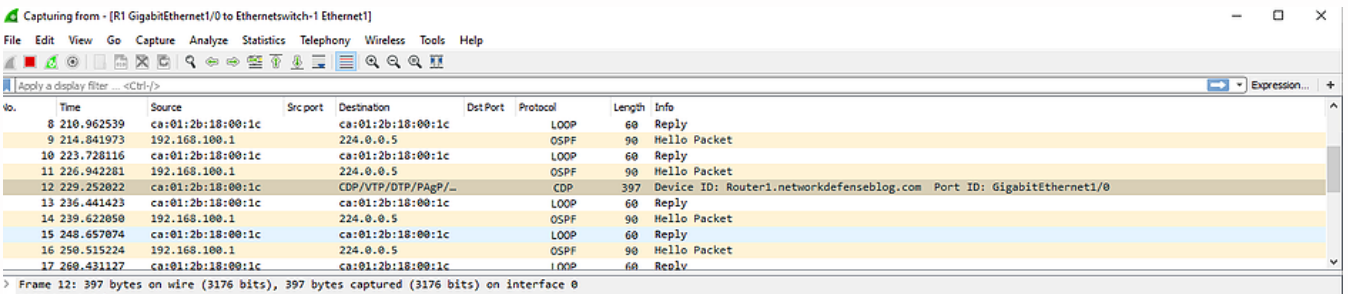
During my research and in the process I found most of the out-of-the-box tools and exploits specific to what I was looking to do lack luster and aged (i.e. pre-2010) and didn't experience stellar results, but that isn't to say there are limited options. Nevertheless, in this article I'll go through a 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 easy cause DoS conditions for your network devices.

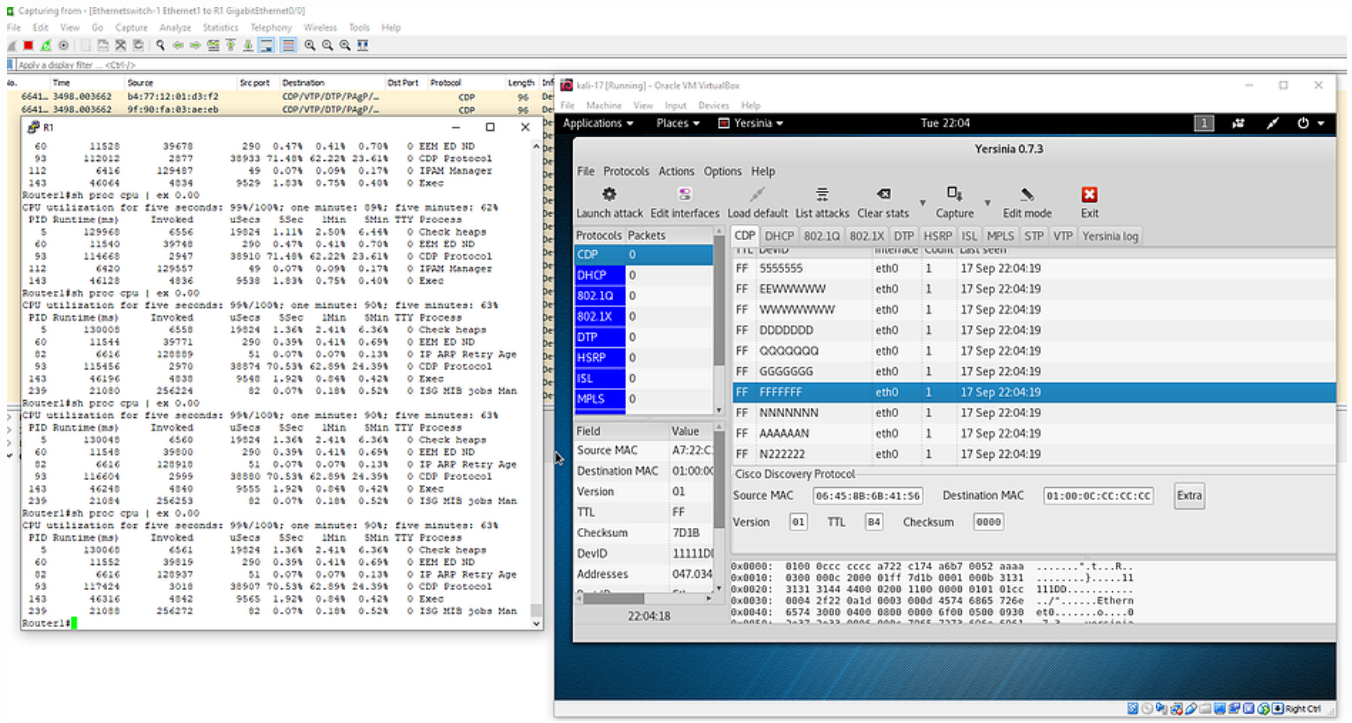
This was the easiest and most successful attack of the few I looked at. You basically have a choice of to just recon the packets and obtain information, or gain the information and then perform a flooding disruption. Since these protocols are Layer 2, by just connecting to a LAN and running Wireshark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



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.

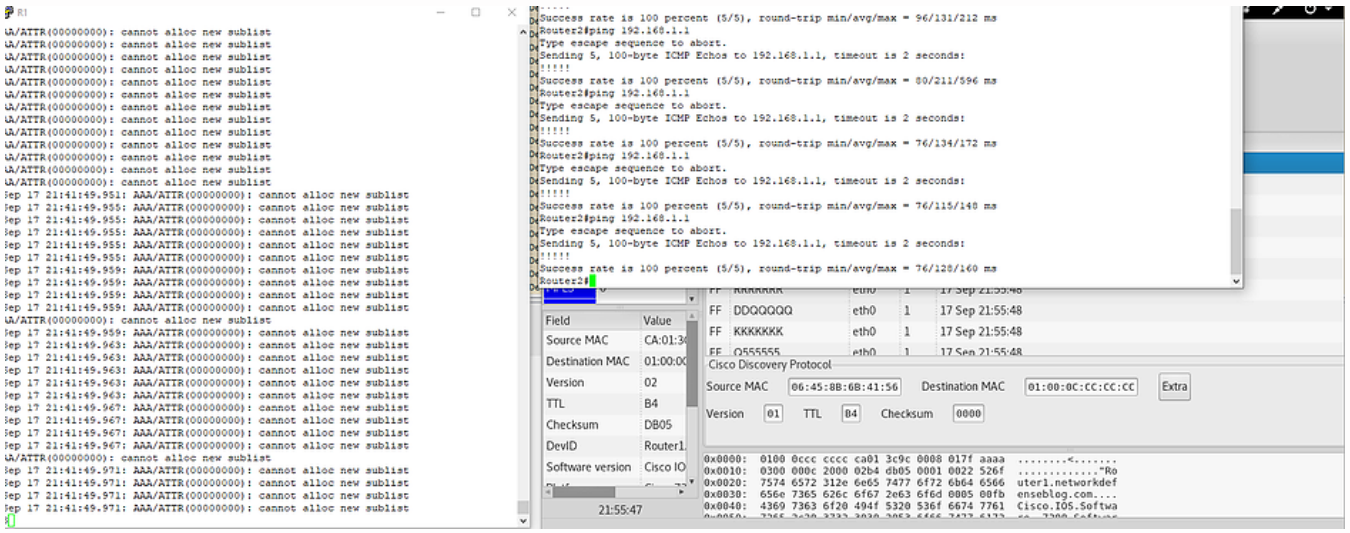
The defensive measure is to go under the interfaces and disable CDP or LLDP so its stops broadcasting and so the device stops processing the requests. Although I did see the overall CPU slightly elevated about 5-10% after disabling CDP both globally or on the interface while still running the attack, 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.



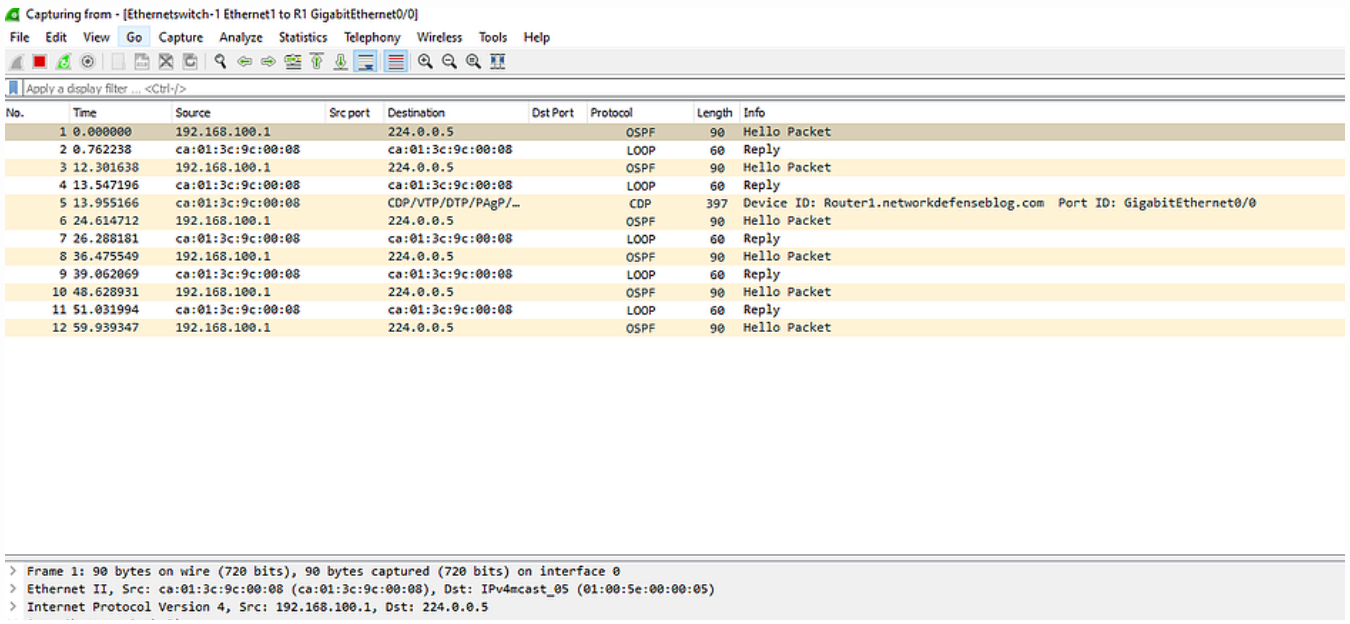
CDP Layer 2 Flood Attack with Yersinia 2

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 are connecting.

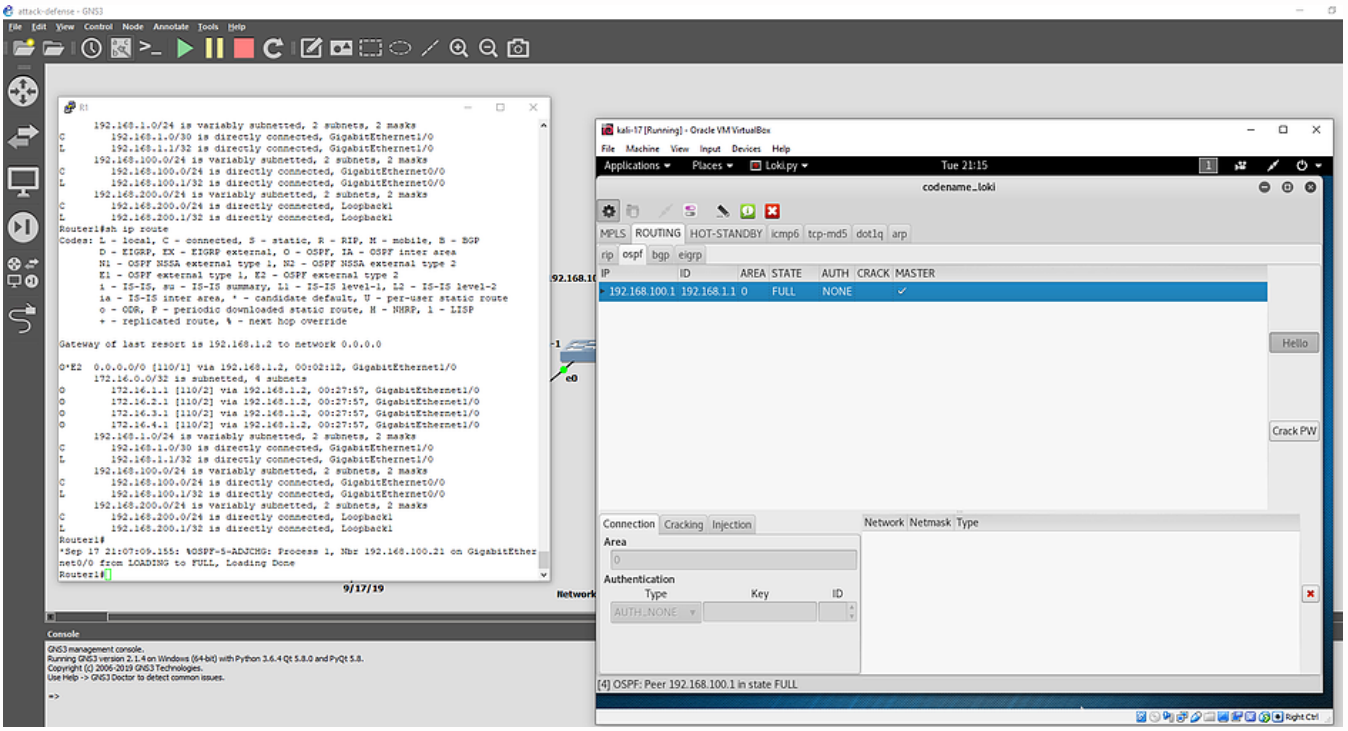
First we look at the OSPF multicast Hello packet on the local LAN to begin our reconnaissance. You'll see common attributes within the packet shown.



```
Open Shortest Path First
  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
  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
  OSPF LLS Data Block
      Checksum: 0xfff6
      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.



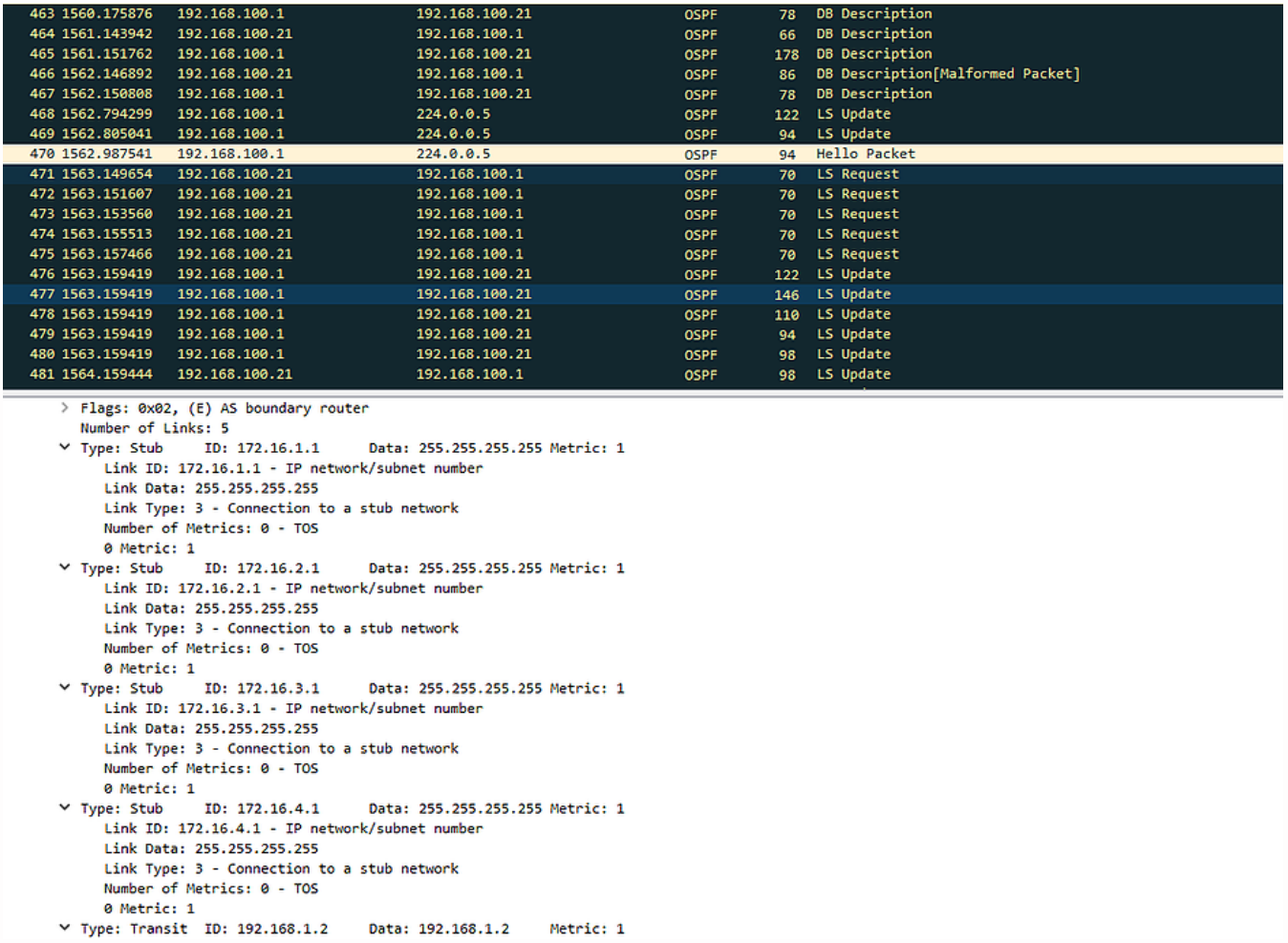
Forming an OSPF Adjaceny to gather info and perform a routing attack

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).

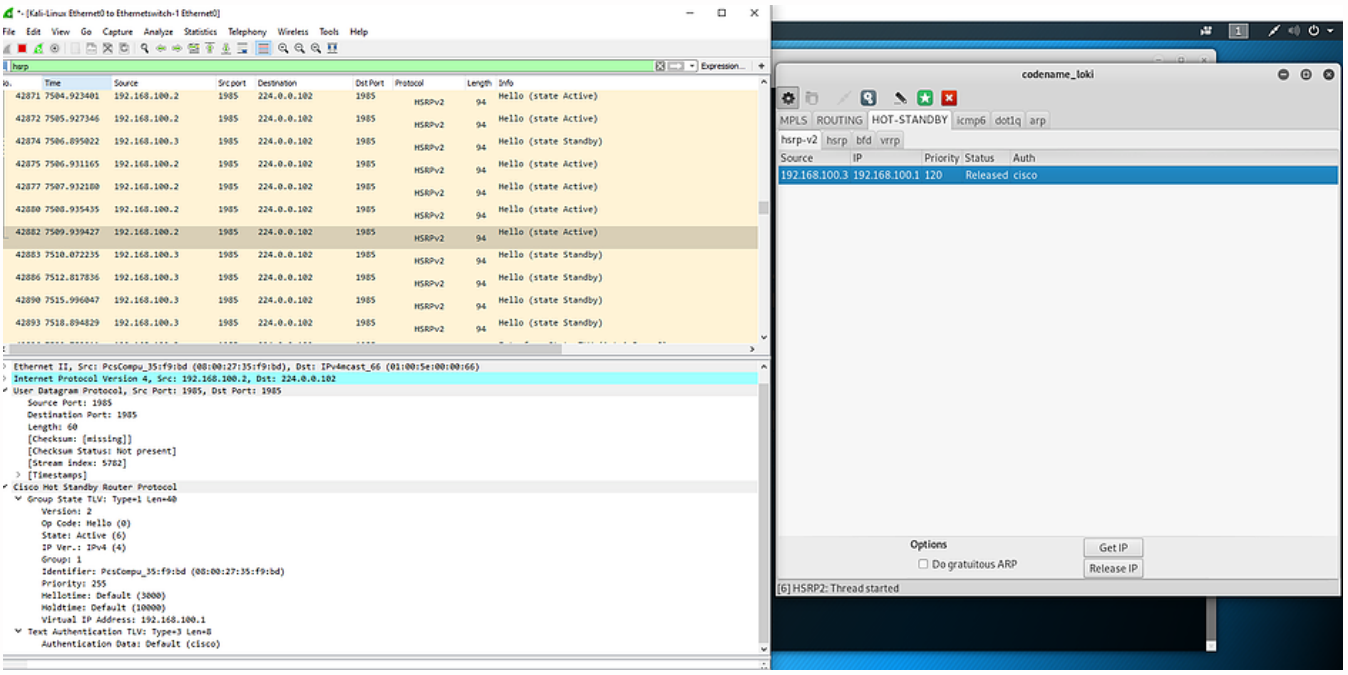


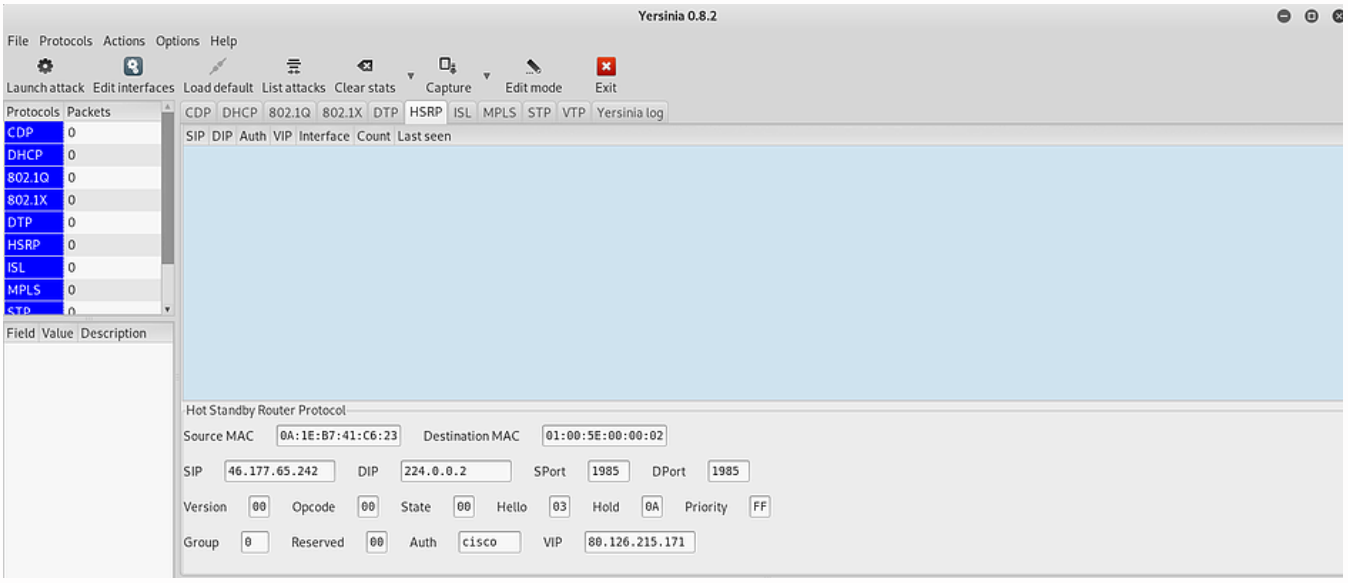
OSPF Database Exchange

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.





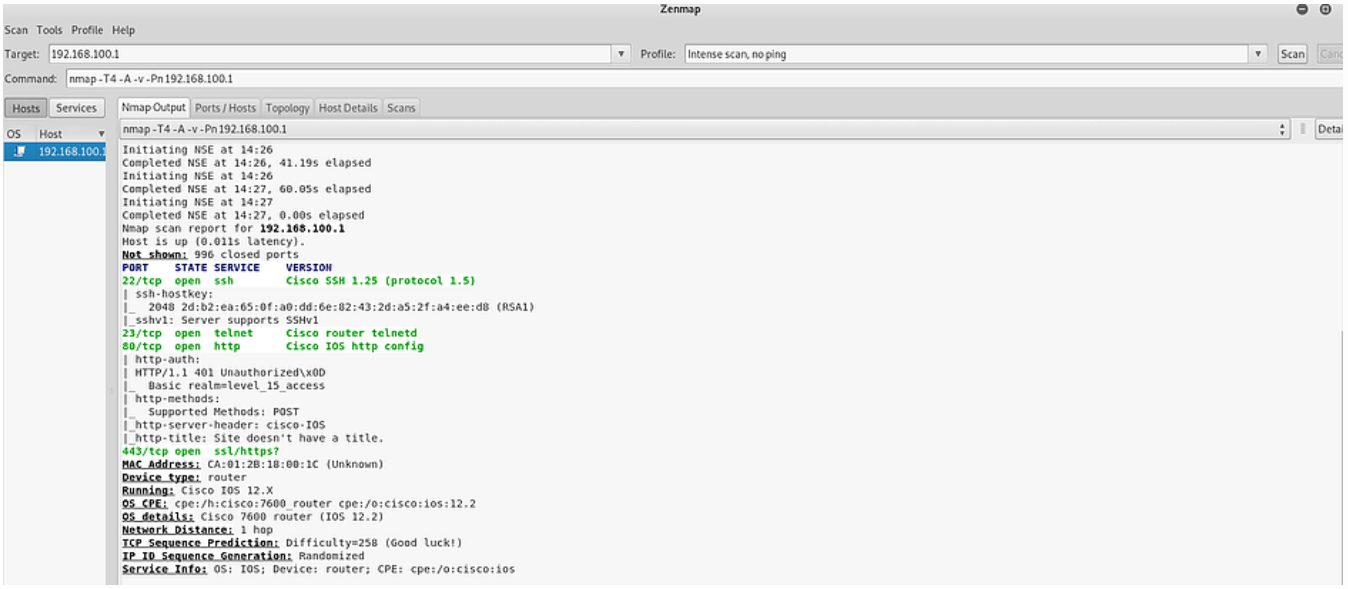
Yersinia's HSRP Hijack options

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!).

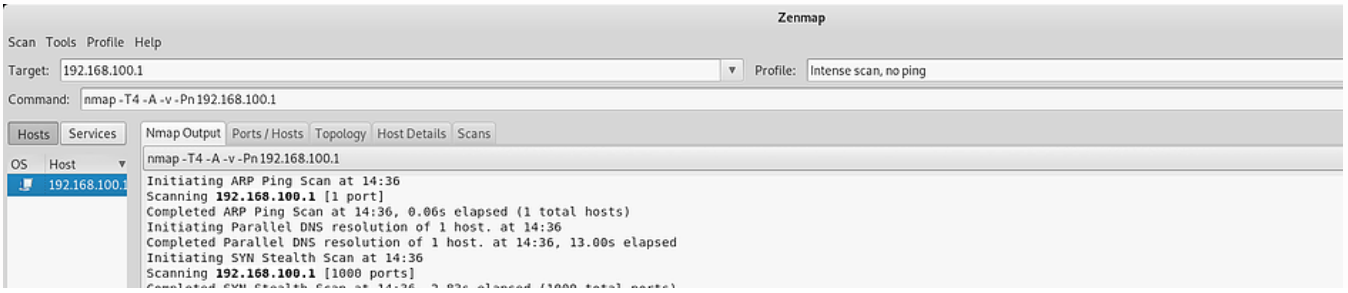
Below are 2 screenshots of a Nmap scan of a router with a before and after implementing an ACL to protect some of the services. Basically a scan was performed with the main services enabled and the second is with ACLs implemented on the VTY lines, NTP, SNMP, and HTTP services. I did perform many different 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.



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



```
Completed SYN Stealth Scan at 14:36, 2.83s 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:01:2B: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

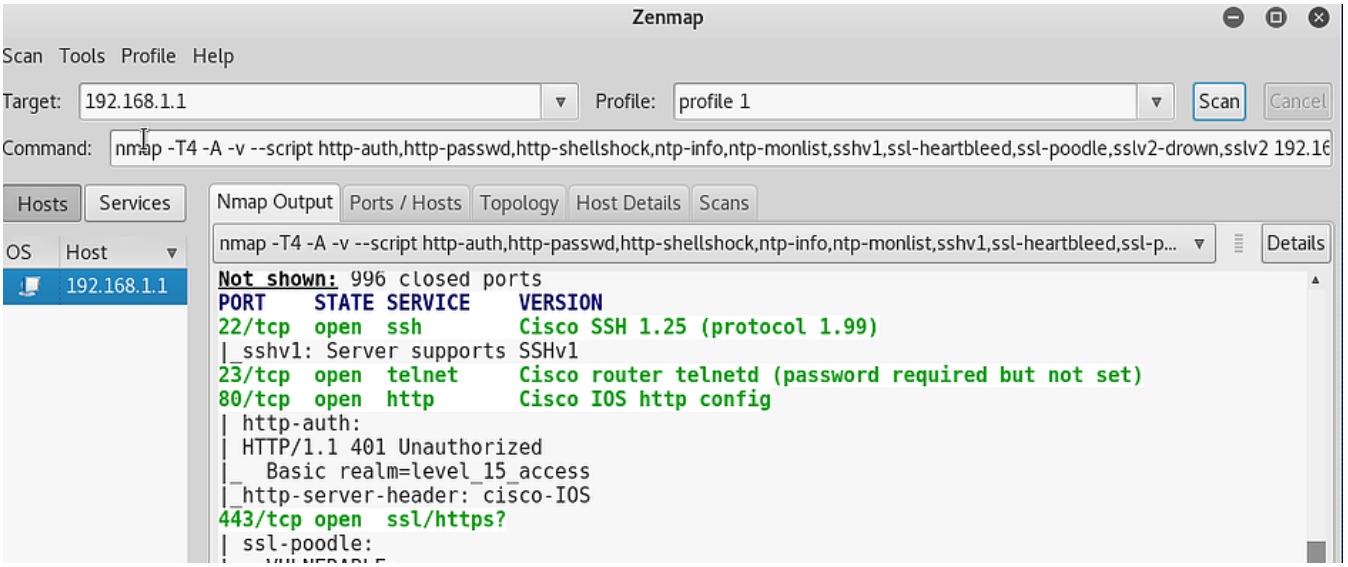
Also consider reviewing your router configurations to ensure there are no unnecessary services enabled and that the services that are enabled are protected. Do you use the web GUI at all? or is there a need to have the HTTP server enabled? if not then consult your vendor's configuration guide to disable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

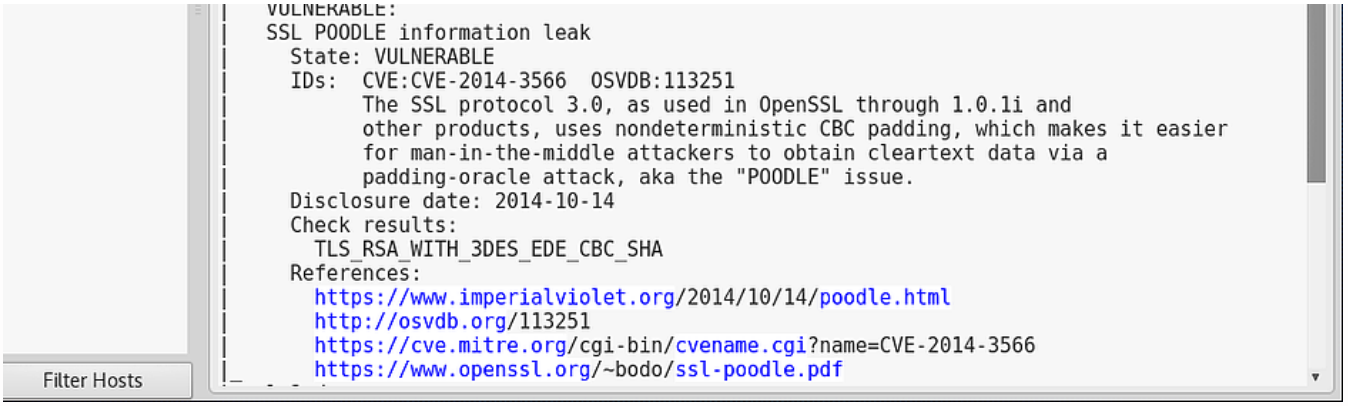
I noticed a few of the recent CVEs being released for Cisco IOS are HTTP based attacks, therefore having that service disabled mitigates the risk for those particular vulnerabilities - same thing applies for other exploits against their corresponding vulnerabilities. By having exposed services (i.e. 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.





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

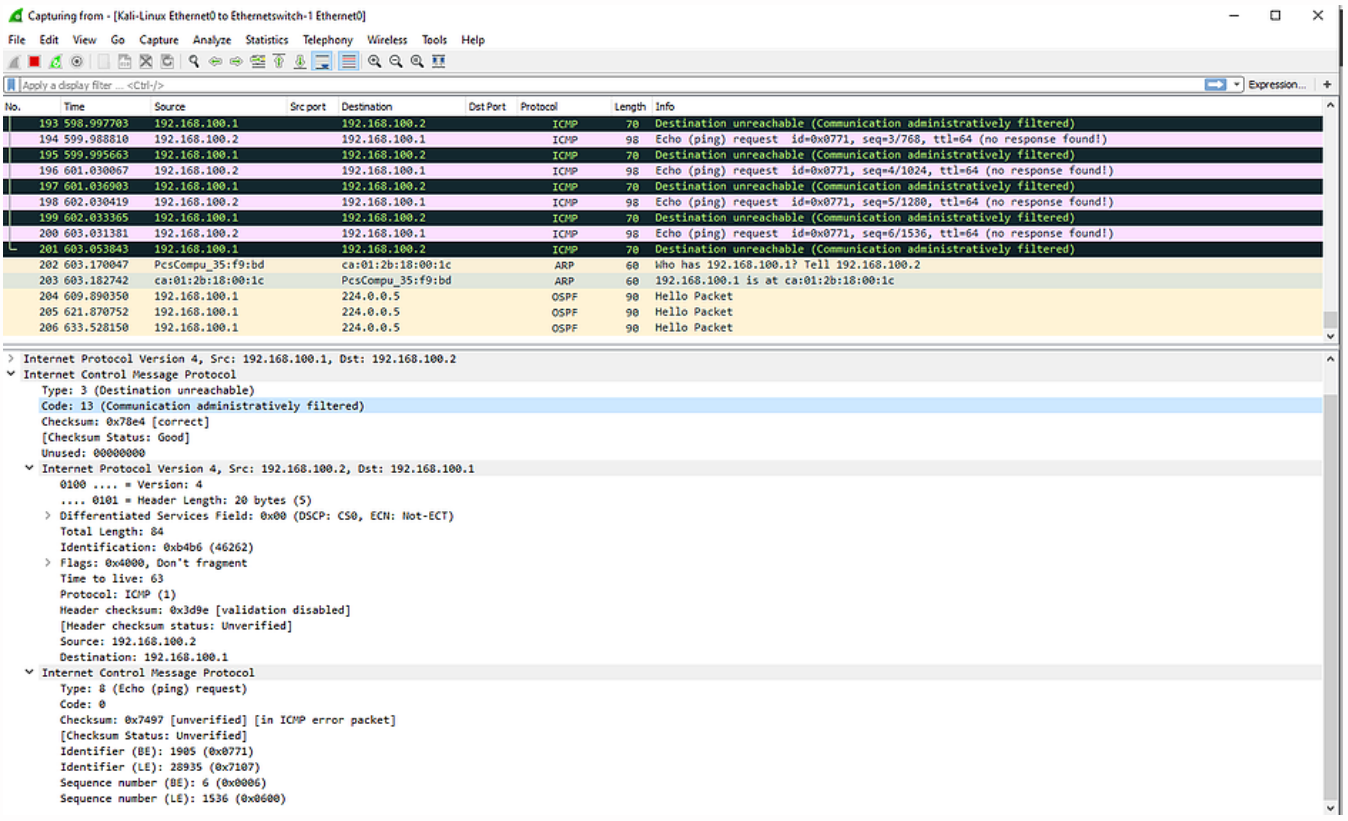
One point I want to bring to your attention is the information Internet control message protocol (ICMP) can give away. Essentially depending on the situation the router will send ICMP replies letting the sending device know certain information. On some vendor's routers you have the opportunity to disable *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,

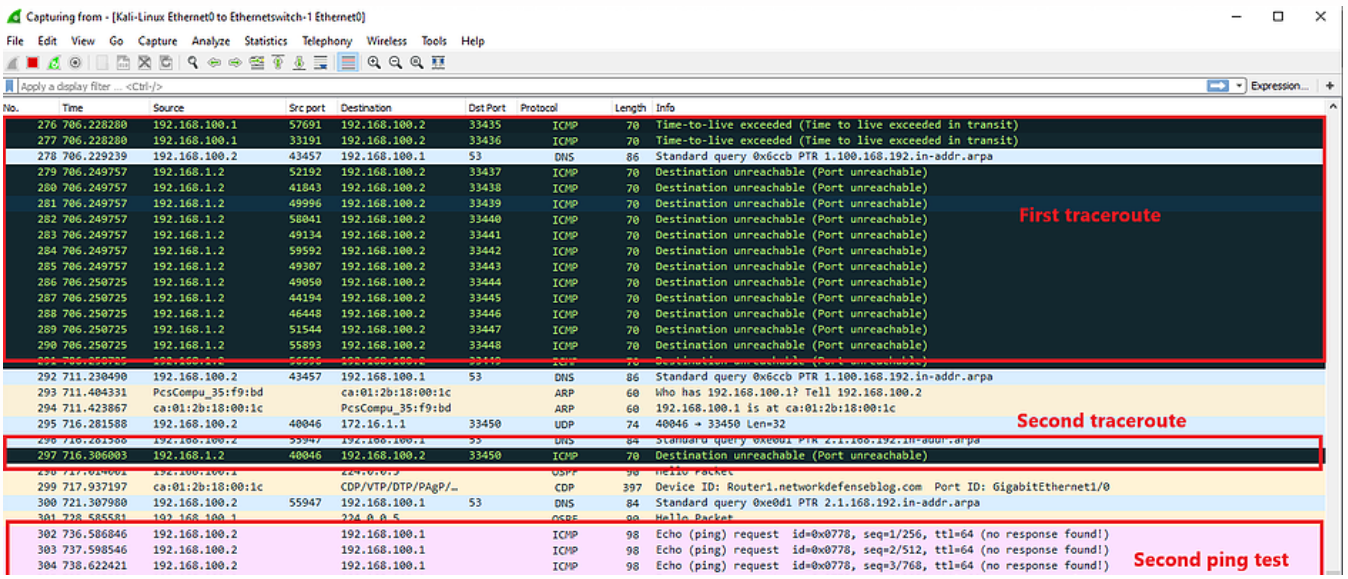
when a block entry is hit the device will respond with a type 3 code 13 " Communication administratively prohibited". This might be unwanted behavior as an attacker could utilize this to their advantage to find open ports or certain types of allowed flows. Here are the results of the first ping test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

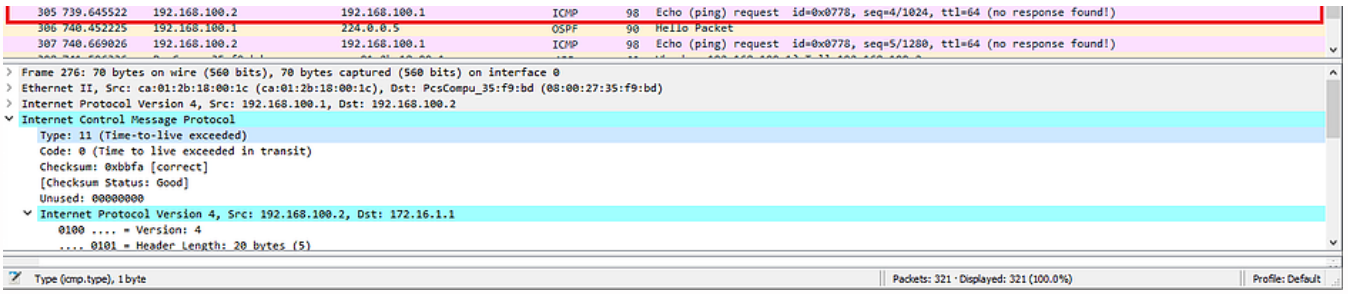


ICMP code 13 packet after pinging traffic restricted by an Access Control List

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 it's traffic was dropped by the access list because of the ICMP reply.



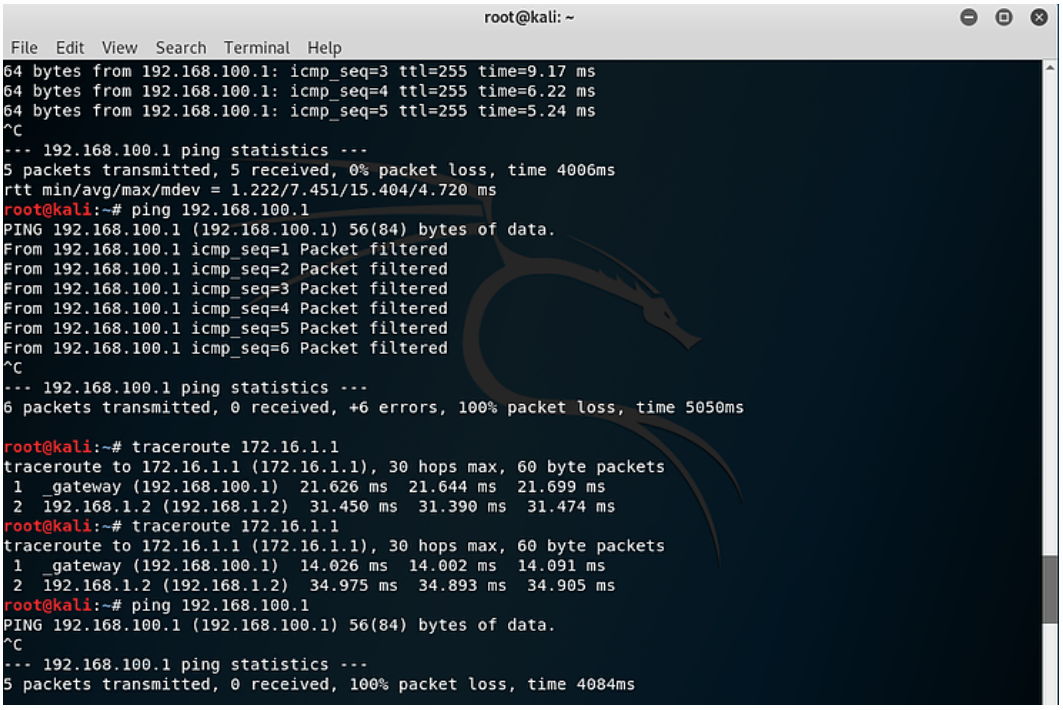


Ping and Traceroute tests as viewed in wireshark before and after IP unreachables are disabled

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

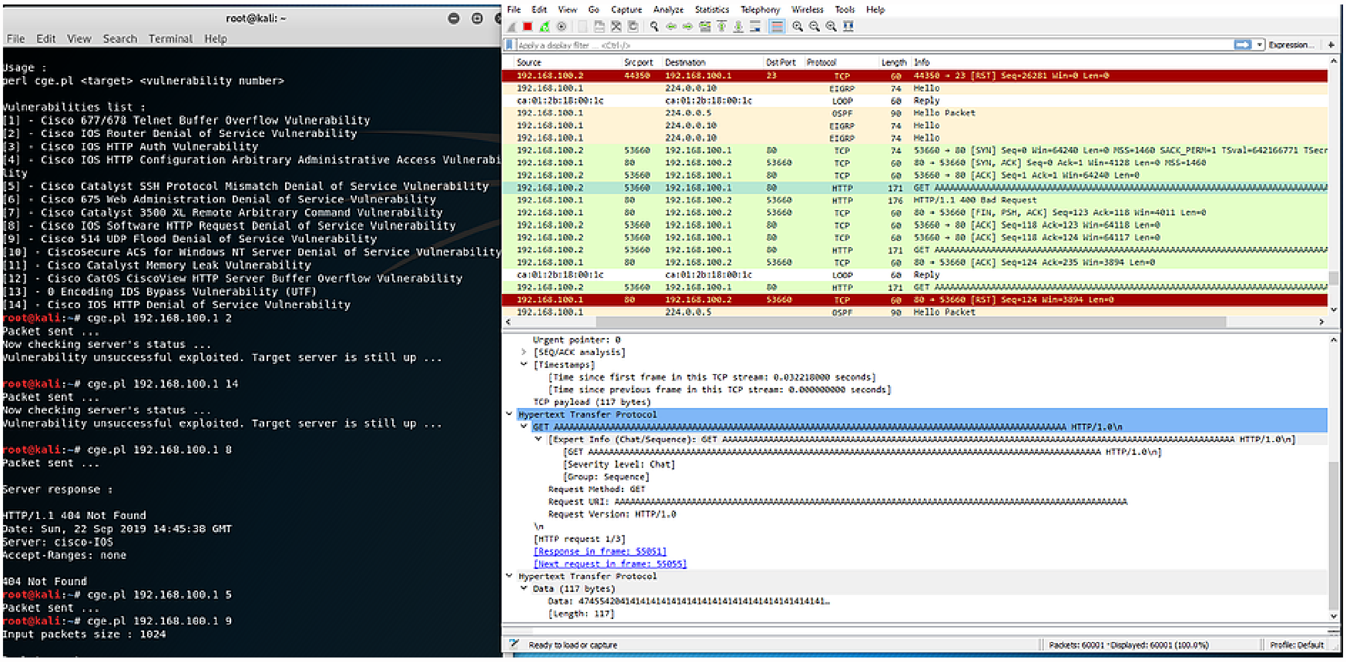


Out of the Box Review

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

patched years ago because you neglected to update.

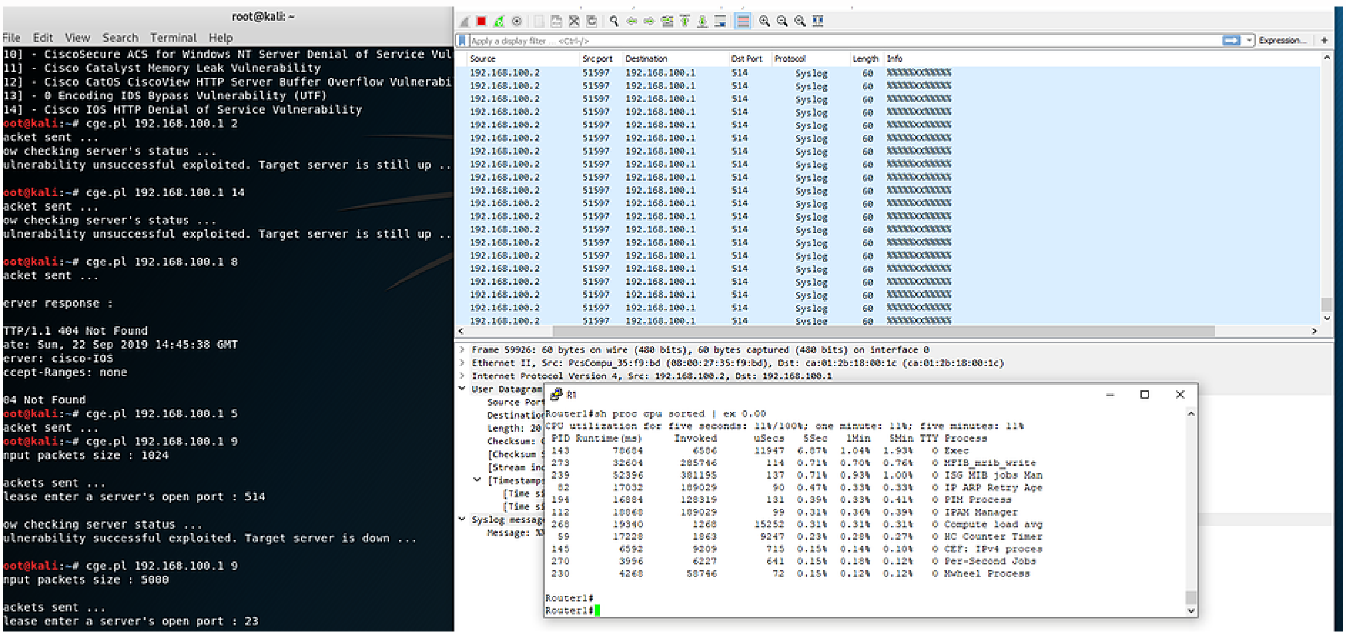
Back to the aged scripts, for example the cisco global exploit showed one option to be for catOS which is old and probably wouldn't be seen by most people and it had a few old HTTP CVEs from the early 2000s. I found one funny in that it sent a HTTP GET with a tons of AAAAAAAAAs which I assume did something 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.

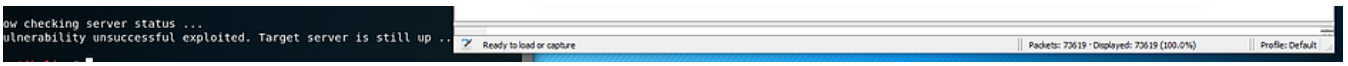


HTTP exploit ran against a Cisco Router

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.

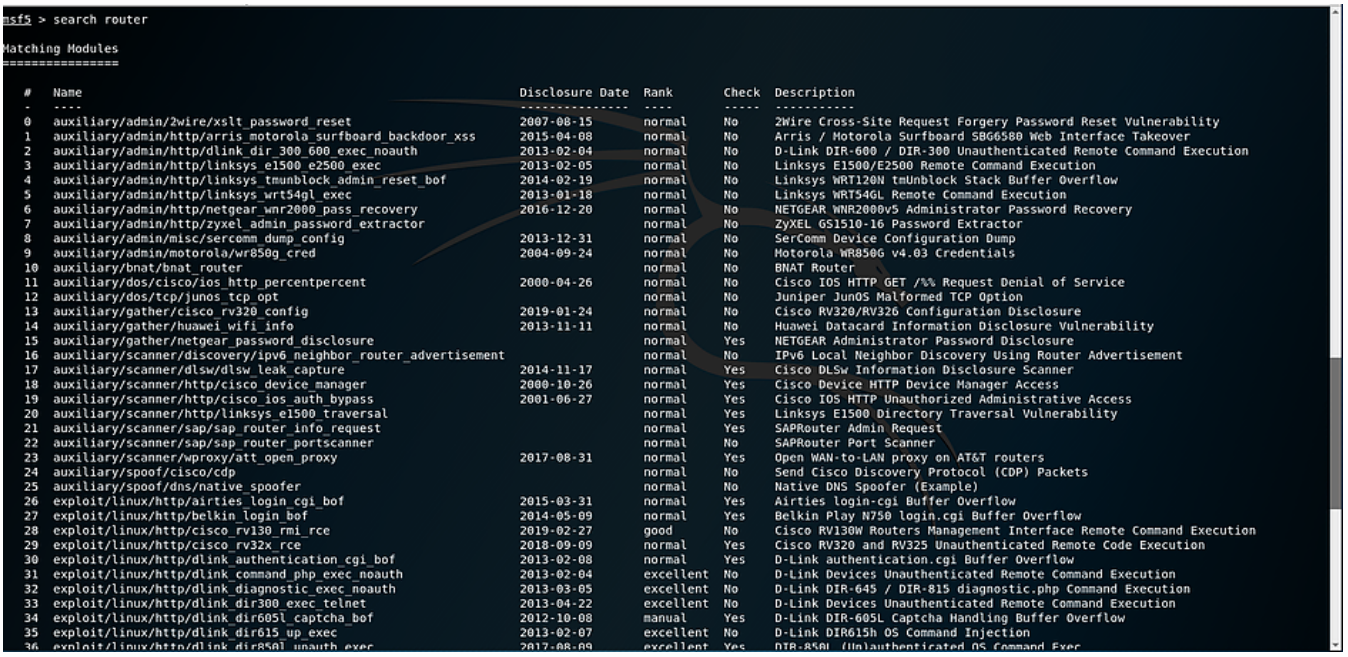
This is how the Syslog #9 vulnerability looked from GCE.





Syslog DoS attempt from CGE

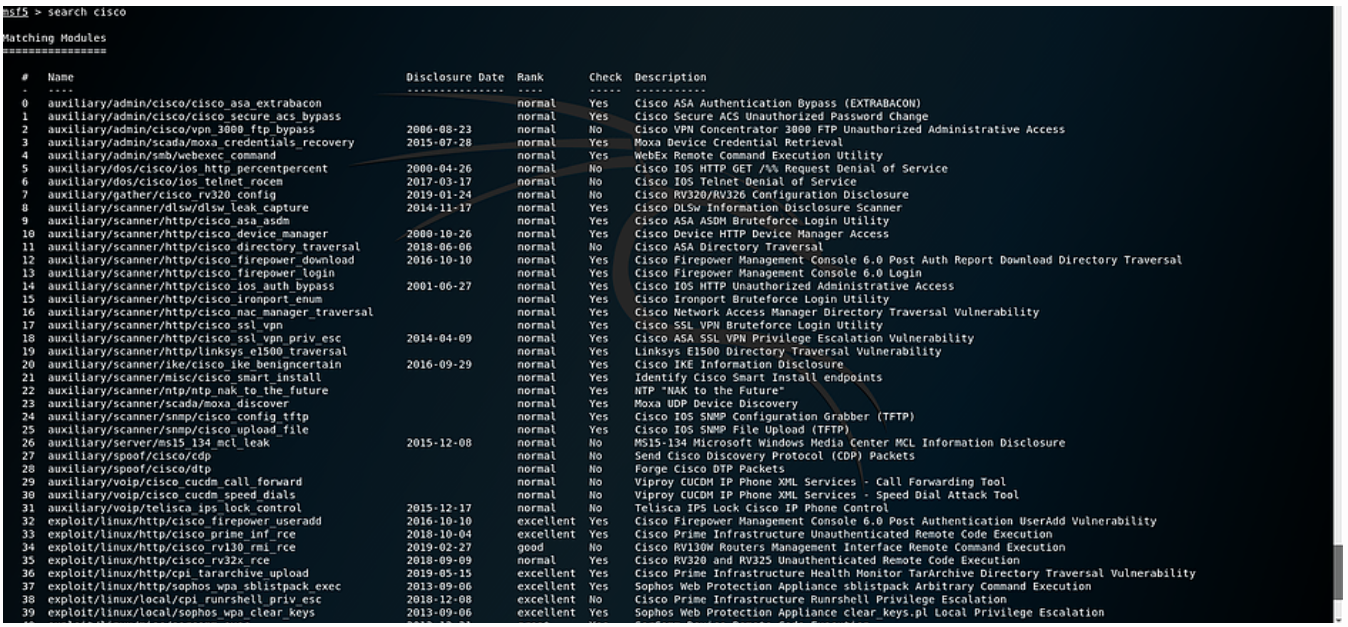
You can see on metasploit there are a lot of options to choose from, but like with the previous examples the attacks I tried didn't do much against a more up to date (although still older IOS) protected device; although I only tested on virtual instances of Cisco routers. I'm sure some of those Soho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



```
msf5 > 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/zyxel_admin_password_extractor                                normal     No     ZyXEL GS1510-16 Password Extractor
   8   auxiliary/admin/misc/sercomm_dump_config                          2013-12-31       normal     No     SerComm Device Configuration Dump
   9   auxiliary/admin/motorola/wr850g_cred                              2004-09-24       normal     No     Motorola WR850G v4.03 Credentials
   10  auxiliary/bnat/bnat_router                                                         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/junos_tcp_opt                                                    normal     No     Juniper JunOS Malformed TCP Option
   13  auxiliary/gather/cisco_rv320_config                               2019-01-24       normal     No     Cisco RV320/RV326 Configuration Disclosure
   14  auxiliary/gather/huawei_wifi_info                                 2013-11-11       normal     No     Huawei Datacard Information Disclosure Vulnerability
   15  auxiliary/gather/netgear_password_disclosure                                       normal     Yes    NETGEAR Administrator Password Disclosure
   16  auxiliary/scanner/discovery/ipv6_neighbor_router_advertisement                     normal     No     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                                     normal     Yes    Linksys E1500 Directory Traversal Vulnerability
   21  auxiliary/scanner/sap/sap_router_info_request                                      normal     Yes    SAPRouter Admin Request
   22  auxiliary/scanner/sap/sap_router_portscanner                                       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                                                          normal     No     Send Cisco Discovery Protocol (CDP) Packets
   25  auxiliary/spoof/dns/native_spoofer                                                 normal     No     Native DNS Spoofer (Example)
   26  exploit/linux/http/airties_login_cgi_bof                          2015-03-31       normal     Yes    Airties login-cgi Buffer Overflow
   27  exploit/linux/http/belkin_login_bof                               2014-05-09       normal     Yes    Belkin Play N750 login.cgi Buffer Overflow
   28  exploit/linux/http/cisco_rv130_rmi_rce                            2019-02-27       good       No     Cisco RV130W Routers Management Interface Remote Command Execution
   29  exploit/linux/http/cisco_rv32x_rce                                2018-09-09       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_dir850l_unauth_exec                      2017-08-09       excellent  Yes    DIR-850L (Unauthenticated OS Command Exec
```



```
msf5 > search cisco

Matching Modules
================

   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   auxiliary/admin/cisco/cisco_asa_extrabacon                             normal     Yes    Cisco ASA Authentication Bypass (EXTRABACON)
   1   auxiliary/admin/cisco/cisco_secure_acs_bypass                          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                                    normal     Yes    WebEx Remote Command Execution Utility
   5   auxiliary/dos/cisco/ios_http_percentpercent           2000-04-26       normal     No     Cisco IOS HTTP GET /%% Request Denial of Service
   6   auxiliary/dos/cisco/ios_telnet_rocem                  2017-03-17       normal     No     Cisco IOS Telnet Denial of Service
   7   auxiliary/gather/cisco_rv320_config                   2019-01-24       normal     No     Cisco RV320/RV326 Configuration Disclosure
   8   auxiliary/scanner/dlsw/dlsw_leak_capture              2014-11-17       normal     Yes    Cisco DLSw Information Disclosure Scanner
   9   auxiliary/scanner/http/cisco_asa_asdm                                  normal     Yes    Cisco ASA ASDM Bruteforce Login Utility
   10  auxiliary/scanner/http/cisco_device_manager           2000-10-26       normal     Yes    Cisco Device HTTP Device Manager Access
   11  auxiliary/scanner/http/cisco_directory_traversal      2018-06-06       normal     No     Cisco ASA Directory Traversal
   12  auxiliary/scanner/http/cisco_firepower_download       2016-10-10       normal     Yes    Cisco Firepower Management Console 6.0 Post Auth Report Download Directory Traversal
   13  auxiliary/scanner/http/cisco_firepower_login                           normal     Yes    Cisco Firepower Management Console 6.0 Login
   14  auxiliary/scanner/http/cisco_ios_auth_bypass          2001-06-27       normal     Yes    Cisco IOS HTTP Unauthorized Administrative Access
   15  auxiliary/scanner/http/cisco_ironport_enum                             normal     Yes    Cisco Ironport Bruteforce Login Utility
   16  auxiliary/scanner/http/cisco_nac_manager_traversal                     normal     Yes    Cisco Network Access Manager Directory Traversal Vulnerability
   17  auxiliary/scanner/http/cisco_ssl_vpn                                   normal     Yes    Cisco SSL VPN Bruteforce Login Utility
   18  auxiliary/scanner/http/cisco_ssl_vpn_priv_esc         2014-04-09       normal     Yes    Cisco ASA SSL VPN Privilege Escalation Vulnerability
   19  auxiliary/scanner/http/linksys_e1500_traversal                         normal     Yes    Linksys E1500 Directory Traversal Vulnerability
   20  auxiliary/scanner/ike/cisco_ike_benigncertain         2016-09-29       normal     Yes    Cisco IKE Information Disclosure
   21  auxiliary/scanner/misc/cisco_smart_install                             normal     Yes    Identify Cisco Smart Install endpoints
   22  auxiliary/scanner/ntp/ntp_nak_to_the_future                            normal     Yes    NTP "NAK to the Future"
   23  auxiliary/scanner/scada/moxa_discover                                  normal     Yes    Moxa UDP Device Discovery
   24  auxiliary/scanner/snmp/cisco_config_tftp                               normal     Yes    Cisco IOS SNMP Configuration Grabber (TFTP)
   25  auxiliary/scanner/snmp/cisco_upload_file                               normal     Yes    Cisco IOS SNMP File Upload (TFTP)
   26  auxiliary/server/ms15_134_mcl_leak                    2015-12-08       normal     No     MS15-134 Microsoft Windows Media Center MCL Information Disclosure
   27  auxiliary/spoof/cisco/cdp                                              normal     No     Send Cisco Discovery Protocol (CDP) Packets
   28  auxiliary/spoof/cisco/dtp                                              normal     No     Forge Cisco DTP Packets
   29  auxiliary/voip/cisco_cucdm_call_forward                                normal     No     Viproy CUCDM IP Phone XML Services - Call Forwarding Tool
   30  auxiliary/voip/cisco_cucdm_speed_dials                                 normal     No     Viproy CUCDM IP Phone XML Services - Speed Dial Attack Tool
   31  auxiliary/voip/telisca_ips_lock_control               2015-12-17       normal     No     Telisca IPS Lock Cisco IP Phone Control
   32  exploit/linux/http/cisco_firepower_useradd            2016-10-10       excellent  Yes    Cisco Firepower Management Console 6.0 Post Authentication UserAdd Vulnerability
   33  exploit/linux/http/cisco_prime_inf_rce                2018-10-04       excellent  Yes    Cisco Prime Infrastructure Unauthenticated Remote Code Execution
   34  exploit/linux/http/cisco_rv130_rmi_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         2013-09-06       excellent  Yes    Sophos Web Protection Appliance sblistpack Arbitrary Command Execution
   38  exploit/linux/local/cpi_runrshell_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

Cyber Security • Networking

Recent Posts

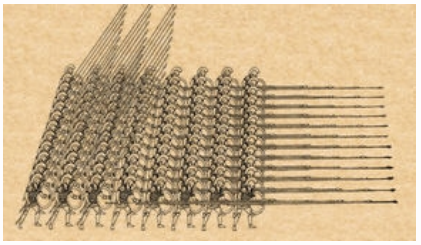
See All

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 4

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

Site Work in Progress - Best viewed on the desktop