

**530.2**

# Network Security Architecture and Engineering

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

**SANS**

THE MOST TRUSTED SOURCE FOR INFORMATION SECURITY TRAINING, CERTIFICATION, AND RESEARCH | [sans.org](https://sans.org)

PLEASE READ THE TERMS AND CONDITIONS OF THIS COURSEWARE LICENSE AGREEMENT ("CLA") CAREFULLY BEFORE USING ANY OF THE COURSEWARE ASSOCIATED WITH THE SANS COURSE. THIS IS A LEGAL AND ENFORCEABLE CONTRACT BETWEEN YOU (THE "USER") AND SANS INSTITUTE FOR THE COURSEWARE. YOU AGREE THAT THIS AGREEMENT IS ENFORCEABLE LIKE ANY WRITTEN NEGOTIATED AGREEMENT SIGNED BY YOU.

With the CLA, SANS Institute hereby grants User a personal, non-exclusive license to use the Courseware subject to the terms of this agreement. Courseware includes all printed materials, including course books and lab workbooks, as well as any digital or other media, virtual machines, and/or data sets distributed by SANS Institute to User for use in the SANS class associated with the Courseware. User agrees that the CLA is the complete and exclusive statement of agreement between SANS Institute and you and that this CLA supersedes any oral or written proposal, agreement or other communication relating to the subject matter of this CLA.

BY ACCEPTING THIS COURSEWARE, YOU AGREE TO BE BOUND BY THE TERMS OF THIS CLA. BY ACCEPTING THIS SOFTWARE, YOU AGREE THAT ANY BREACH OF THE TERMS OF THIS CLA MAY CAUSE IRREPARABLE HARM AND SIGNIFICANT INJURY TO SANS INSTITUTE, AND THAT SANS INSTITUTE MAY ENFORCE THESE PROVISIONS BY INJUNCTION (WITHOUT THE NECESSITY OF POSTING BOND) SPECIFIC PERFORMANCE, OR OTHER EQUITABLE RELIEF.

If you do not agree, you may return the Courseware to SANS Institute for a full refund, if applicable.

User may not copy, reproduce, re-publish, distribute, display, modify or create derivative works based upon all or any portion of the Courseware, in any medium whether printed, electronic or otherwise, for any purpose, without the express prior written consent of SANS Institute. Additionally, User may not sell, rent, lease, trade, or otherwise transfer the Courseware in any way, shape, or form without the express written consent of SANS Institute.

If any provision of this CLA is declared unenforceable in any jurisdiction, then such provision shall be deemed to be severable from this CLA and shall not affect the remainder thereof. An amendment or addendum to this CLA may accompany this Courseware.

SANS acknowledges that any and all software and/or tools, graphics, images, tables, charts or graphs presented in this Courseware are the sole property of their respective trademark/registered/copyright owners, including:

AirDrop, AirPort, AirPort Time Capsule, Apple, Apple Remote Desktop, Apple TV, App Nap, Back to My Mac, Boot Camp, Cocoa, FaceTime, FileVault, Finder, FireWire, FireWire logo, iCal, iChat, iLife, iMac, iMessage, iPad, iPad Air, iPad Mini, iPhone, iPhoto, iPod, iPod classic, iPod shuffle, iPod nano, iPod touch, iTunes, iTunes logo, iWork, Keychain, Keynote, Mac, Mac Logo, MacBook, MacBook Air, MacBook Pro, Macintosh, Mac OS, Mac Pro, Numbers, OS X, Pages, Passbook, Retina, Safari, Siri, Spaces, Spotlight, There's an app for that, Time Capsule, Time Machine, Touch ID, Xcode, Xserve, App Store, and iCloud are registered trademarks of Apple Inc.

PMP and PMBOK are registered marks of PMI.

SOF-ELK® is a registered trademark of Lewes Technology Consulting, LLC. Used with permission.

SIFT® is a registered trademark of Harbingers, LLC. Used with permission.

Governing Law: This Agreement shall be governed by the laws of the State of Maryland, USA.

SANS

# Network Security Architecture and Engineering

© 2019 Eric Conrad, Justin Henderson, & Ismael Valenzuela | All Rights Reserved | Version E01\_02

Welcome to SEC530.2, Network Security Architecture and Engineering!

Table of Contents	Page
Layer 3 Attacks and Mitigation.....	4
Switch and Router Benchmarks.....	15
<b>EXERCISE:</b> Auditing Router Security.....	30
Securing SNMP.....	32
Securing NTP.....	39
Bogon Filtering .....	45
Blackholes and Darknets.....	49
<b>EXERCISE:</b> Router SNMP Security.....	55
IPv6.....	57
IPv6 Misconceptions.....	86
Securing IPv6.....	91
<b>EXERCISE:</b> IPv6 .....	113

### 530.2 Table of Contents

This table of contents outlines our plan for 530.2.

Table of Contents	Page
Layer 3/4 Stateful Firewall.....	115
Web Proxy .....	131
SMTP Proxy .....	153
<b>EXERCISE:</b> Proxy Power .....	168
530.2 Summary .....	170

### 530.2 Table of Contents

This table of contents outlines our plan for 530.2.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

## Course Roadmap

- Day 1: Defensible Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- Day 5: Zero Trust Architecture
- Day 6: Capstone: Hands-On Defend the Flag Challenge

### NETWORK SECURITY ARCHITECTURE

1. **Layer 3 Attacks and Mitigation**
2. Switch and Router Benchmarks
3. **EXERCISE: Auditing Router Security**
4. Securing SNMP
5. Securing NTP
6. Bogon Filtering
7. Blackholes and Darknets
8. **EXERCISE: Router SNMP Security**
9. IPv6
10. IPv6 Misconceptions
11. Securing IPv6
12. **EXERCISE: IPv6**
13. Layer 3/4 Stateful Firewall
14. Web Proxy
15. SMTP Proxy
16. **EXERCISE: Proxy Power**
17. 530.2 Summary

### Course Roadmap

We will next discuss layer 3 attacks and mitigation.

**OSI Model Layers Discussed in Depth in 530.2**

- Layer 3: Network
  - Packets, routing, IP addressing, IPSec, ICMP
- Layer 4: Transport
  - Datagrams, UDP, TCP

7	Application
6	Presentation
5	Session
4	<b>Transport</b>
3	<b>Network</b>
2	<b>Data Link</b>
1	<b>Physical</b>

**OSI Model Layers Discussed in Depth in 530.2**

We will not be covering the OSI model in depth, but 'hitting the highlights,' focusing on the terms we will use during 530.1, such as 'layer 1,' 'layer 2,' etc.

Licensed To: Martin Brown <hermespaul56@gmail.com>



- Common basic issues related to routers too:
  - Secure administration
  - Services offered
  - Vulnerabilities
  - ACLs
  - Banners
  - Logging
  - Authentication, Authorization and Accounting

This page intentionally left blank.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020



## Red Team Scenario – VPNFilter

Knowing that Tyrell Corp. hasn't updated their routers for a while... the replicants are going to leverage a sophisticated piece of malware, VPNFilter, to do man-in-the-middle and eavesdrop on traffic passing through the router.



### Red Team Scenario – VPNFilter

[1] [https://www.schneier.com/blog/archives/2018/06/router\\_vulnerab.html](https://www.schneier.com/blog/archives/2018/06/router_vulnerab.html)

[2] <https://securingtomorrow.mcafee.com/mcafee-labs/vpnfilter-botnet-targets-networking-devices/>

[3] <https://securingtomorrow.mcafee.com/mcafee-labs/vpnfilter-malware-adds-capabilities-to-exploit-endpoints/>

## Threats on 530.2 - Routers



- Scanning
- Fingerprinting (passive)
- DOS attack
- IP spoofing
- IP source routing
- ICMP flooding
- Smurf attacks
- Unauthorized tunneling
- Routing table poisoning

This page intentionally left blank.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

## Layer 3 Attacks and Mitigation

- Layer 3 Attacks include:
  - Man-in-the-middle attacks such as IPv6 router advertisement
  - Unauthorized routing updates
  - Wormhole attack (unauthorized tunneling)
- We will discuss these attacks and respective mitigations

### Layer 3 Attacks and Mitigation

We will next discuss common layer 3 attacks, including:

- Man-in-the-middle attacks such as IPv6 router advertisement
- Unauthorized routing updates
- Wormhole attack (unauthorized tunneling)

Let's discuss each!

## Routing Protocols

- Routing protocols are designed to:
  - Automatically learn a network topology including redundant paths
  - Choose the optimal route that offers the best bandwidth and lowest latency
  - Attempt to automatically and quickly route around network outages
- There are two basic types of routing protocols: Interior Gateway Protocols (IGPs) and Exterior Gateway Protocols (EGPs)
  - IGP Examples: OSPF, EIGRP, IS-IS
  - EGP: BGP

### Routing Protocols

Simple networks (such as a home or small office networks) often use a simple static route, pointing to a single default gateway. Larger networks with multiple paths to other networks benefit from routing protocols, which learn the fastest routes, automatically detect outages, and quickly select new routes.

Interior Gateway Protocols are used internally across privately-owned networks, such as WANs. Common examples include OSPF (Open Shortest Path First), EIGRP (Extended Interior Gateway Protocol), and IS-IS (Intermediate System-to-Intermediate System). Legacy examples include RIPv1 and RIPv2 (the Routing Information Protocol). Note that RIP is a legacy protocol with a number of flaws and shortcomings, and its use should be avoided.

BGP is the primary EGP (Exterior Gateway Protocol), which may be used either internally or via the Internet.

## Unauthorized Routing Updates

- All routing protocols should use neighbor authentication to avoid unauthorized routing updates
  - Without neighbor authentication: a properly positioned black hat could inject bogus routes
  - This would make Man-in-the-Middle (MitM) attacks easy to accomplish
- Routing protocols support plaintext or key (hash) authentication (BGP and EIGRP only allow hashed)
  - Plaintext should be avoided (the key may be sniffed and reused)
  - MD5 is available across most equipment
  - Some newer/more advanced equipment support HMAC-SHA-256 or HMAC-SHA-512

### Unauthorized Routing Updates

OSPF and IS-IS support either plaintext or hashed authentication. BGP and EIGRP only support hashes. MD5 has long been the standard, but HMAC-SHA-256 is available on some equipment for EIGRP, OSPFv2 (beginning with Cisco IOS 15.4T). Note that RIP is a legacy protocol; its use should be avoided. Keys may be manually configured on each router. Key chains can automatically rotate keys. EIGRP, IS-IS and OSPFv2+ are now able to use key chains. Key chains can be given lifetimes, allowing easy rotation. Here is an example of creating two keys with different lifetimes. Note that send-lifetime controls when the key is sent for authentication, while accept-lifetime controls when it will be accepted for authentication.

```
Router(config)# key chain 530CHAIN
Router(config)# key 1
Router(config)# key-string Security530Rocks
Router(config)# send-lifetime 00:00:00 Apr 1 2018 00:00:00 Jul 1 2018
Router(config)# accept-lifetime 00:00:00 Apr 1 2018 00:00:00 Jul 1 2018
Router(config)# key chain 530CHAIN
Router(config)# key 2
Router(config)# key-string WickedSecureKey
Router(config)# send-lifetime 00:00:00 Jul 1 2018 00:00:00 Oct 1 2018
Router(config)# accept-lifetime 00:00:00 Jul 1 2018 00:00:00 Oct 1 2018
```

## Neighbor Authentication

- SHA-256 and SHA-512 should be used when supported by all routers
  - Many organizations find some (but not all) of their routers support them
- MD5 is a reasonable fallback, so we'll detail that configuration

```
RouterA(config)# interface GigabitEthernet0/0
```

```
RouterA(config-if)# ip authentication mode eigrp 530 md5
```

```
RouterA(config-if)# ip authentication key-chain eigrp 530 530CHAIN
```

- Router B configuration shown in the notes

### Neighbor Authentication

Let's configure EIGRP with md5 authentication on Router A and Router B. This assumes the key chain was set up as shown on the previous slide. Note the '530' in the commands below is for the Autonomous System (AS) number, used to identify a routing domain. The number is arbitrary in this case but must match on all configured routers.

Then configure router B:

```
RouterB(config)# interface GigabitEthernet0/0
```

```
RouterB(config-if)# ip authentication mode eigrp 530 md5
```

```
RouterB(config-if)# ip authentication key-chain eigrp 530 530CHAIN
```

Configurations for OSPF, IS-IS, etc., are available from the Cisco IOS Security Configuration Guide, Chapter: Neighbor Router Authentication: Overview and Guidelines.<sup>1</sup>

[1]

<https://cloudsso.cisco.com/sp/startSSO.ping?SpSessionAuthnAdapterId=standardnomfa&TargetResource=https://sso.cisco.com/autho/login/loginaction.html>

## Wormhole Attack

- A wormhole attack describes an unauthorized tunnel, typically configured to and from an internal router
  - The attack requires unauthorized access to a router
  - This allows a Man-in-the-Middle attack on any packet that is routed by that router
- Wormhole attacks may be used to tunnel traffic by individual ports, IP addresses, or networks
  - How many organizations would notice that a web server was suddenly 2 network hops further away than an SSH (Secure Shell) server on the same system?

### Wormhole Attack

Yih-Chun Hu describes wormhole attacks:

*In a wormhole attack, an attacker receives packets at one point in the network, “tunnels” them to another point in the network, and then replays them into the network from that point.<sup>1</sup>*

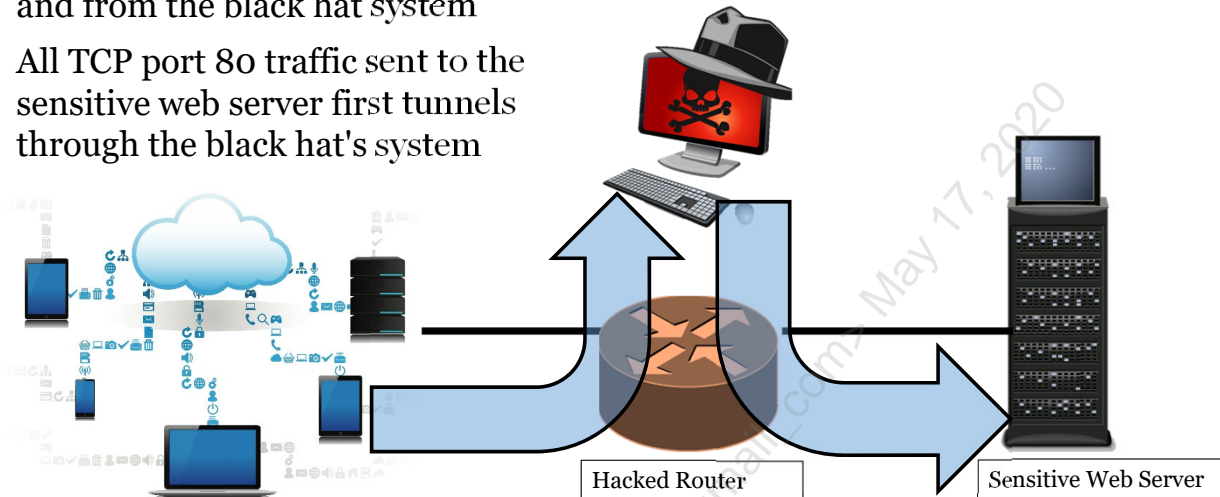
Wormhole attacks can be devastating, allowing black hats to tunnel any port, IP address, or network to and from a malicious router. As noted above: the attack requires unauthorized access to a router. The mitigation is straightforward: prevent unauthorized access to routers.

Later in 530.2, we will show an SNMP attack that allows downloading the Cisco IOS configuration if the attacker can guess the SMMP write string (we will also discuss SNMP later in 530.2). If successful: the attacker could read type IOS 0 passwords, decode type 7 (Vigenère Cipher), or attempt to crack the IOS type 5, 8, or 9 passwords.

[1] <https://www.semanticscholar.org/paper/Wormhole-attacks-in-wireless-networks-Hu-Perrig/730473b193c56d4996dd11569837700e5e5bc9b4>

## Wormhole Attack Illustrated

- The arrows represent GRE tunnels to and from the black hat system
- All TCP port 80 traffic sent to the sensitive web server first tunnels through the black hat's system



### Wormhole Attack Illustrated

The sensitive web server on the right is running SSH on port 22, and HTTP on port 80. It will also respond to ICMP echo requests.

The black hat shown above has gained configuration access to an internal router. The attacker has chosen to tunnel port 80 (only) to and from a compromised system on the internet and has configured a bidirectional GRE (Generic Routing Encapsulation, discussed later in 530.2) tunnel from the compromised router. The attacker routes traffic destined for port 80 on the web server to and from the compromised system.

Assume network on the left is one hop away from the web server, and ICMP or SSH traffic sent to hit will arrive with the time to live decremented by one (when they route via the compromised router. The HTTP server on TCP port 80 is now three hops away: it will route via the compromised router, to the compromised internet system via the GRE tunnel, back to the compromised router via the same GRE tunnel, to the server.

The tunnel can be easily spotted via the compromised router's Cisco IOS configuration. However, diagnosing this issue with typical network testing tools such as ping or traceroute will not be effective since their packets will not traverse the tunnel. The tool hping could be used to diagnose the issue, this will show the initial TTL -3, while a regular ping would show the initial TTL -1:

```
$ hping -S -p80 sensitive.sec530.com
```



## Course Roadmap

- Day 1: Defensible Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- Day 5: Zero Trust Architecture
- Day 6: Capstone: Hands-On Defend the Flag Challenge

### NETWORK SECURITY ARCHITECTURE

1. Layer 3 Attacks and Mitigation
2. **Switch and Router Benchmarks**
3. EXERCISE: Auditing Router Security
4. Securing SNMP
5. Securing NTP
6. Bogon Filtering
7. Blackholes and Darknets
8. EXERCISE: Router SNMP Security
9. IPv6
10. IPv6 Misconceptions
11. Securing IPv6
12. EXERCISE: IPv6
13. Layer 3/4 Stateful Firewall
14. Web Proxy
15. SMTP Proxy
16. EXERCISE: Proxy Power
17. 530.2 Summary

### Course Roadmap

We will next discuss switch and router benchmarks.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

## Layer 2 and 3 Benchmarks and Auditing Tools

We will next discuss layer 2 and layer 3 benchmarks and auditing tools, including:

- Cisco's Best Practices
- Cisco AutoSecure
- DISA (Defense Information Systems Agency) STIGs (Security Technical Implementation Guide)
- CISecurity
- Nipper-ng

### Layer 2 and 3 Benchmarks and Auditing Tools

We will next discuss layer 2 and layer 3 benchmarks and auditing tools, including:

- Cisco's Best Practices
- Cisco AutoSecure
- DISA (Defense Information Systems Agency) STIGs (Security Technical Implementation Guide)
- CISecurity
- Nipper-ng

Let's begin!

## Cisco Best Practices

- Cisco produces lots of high-quality documentation for securing their devices
  - Also loads of great third-party sites, blogs, etc.
  - Google away: you will discover a **lot** of information
- The best 'one-stop shopping' guide for securing Cisco switches and routers is the 'Cisco Guide to Harden Cisco IOS Devices'<sup>1</sup>
  - It is concise and actionable
  - It also covers switches and routers in equal depth
  - Many hardening guides (such as CISecurity) cover routers in detail, but not switches

### Cisco Best Practices

There are many high-quality guides and best practices for securing routers, but many of them do not cover switches in detail. The 'Cisco Guide to Harden Cisco IOS Devices' covers both quite well (the same is true for the DISA STIGs, discussed next).

Here is the high-level summary of 'Cisco Guide to Harden Cisco IOS Devices' secure operations section:

- *Monitor Cisco Security Advisories and Responses*
- *Leverage Authentication, Authorization, and Accounting*
- *Centralize Log Collection and Monitoring*
- *Use Secure Protocols When Possible*
- *Gain Traffic Visibility with NetFlow*
- *Configuration Management<sup>1</sup>*

[1] <https://www.cisco.com/c/en/us/support/docs/ip/access-lists/13608-21.html>

## Cisco AutoSecure

- Cisco's AutoSecure automatically configures a switch or router for a variety of best practices
  - "The AutoSecure feature secures a router by using a single CLI command to disable common IP services that can be exploited for network attacks, enable IP services and features that can aid in defense of a network when under attack, and simplify and harden the security configuration of the router."<sup>1</sup>

```
ericconrad — ssh student@192.168.198.133 — 63x17
--- AutoSecure Configuration ---

*** AutoSecure configuration enhances the security of
the router, but it will not make it absolutely resistant
to all security attacks ***

AutoSecure will modify the configuration of your device.
All configuration changes will be shown. For a detailed
explanation of how the configuration changes enhance security
and any possible side effects, please refer to Cisco.com for
Autosecure documentation.
At any prompt you may enter '?' for help.
Use ctrl-c to abort this session at any prompt.

Gathering information about the router for AutoSecure

Is this router connected to internet? [no]: █
```

**Cisco AutoSecure** provides a very fast 'win' for automatically configuring and enabling best practices for both switch and routers. It secures a variety of features, including the Firewall, SSH, NTP, the management plane, and more:

```
Router# auto secure ?
firewall      AutoSecure Firewall
forwarding    Secure Forwarding Plane
full          Interactive full session of AutoSecure
login         AutoSecure Login
management    Secure Management Plane
no-interact   Non-interactive session of AutoSecure
ntp           AutoSecure NTP
ssh           AutoSecure SSH
tcp-intercept AutoSecure TCP Intercept
```

We chose 'full' and will show some of the guidance beginning next.

[1] [https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec\\_usr\\_cfg/configuration/xc-3s/sec-usr-cfg-xc-3s-book/sec-autosecure.html](https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_usr_cfg/configuration/xc-3s/sec-usr-cfg-xc-3s-book/sec-autosecure.html)

## AutoSecure Mitigations

- The screenshot on the right shows AutoSecure's automatic hardening of the management plane
  - Includes configuring and enabling many of the best practices discussed previously, such as disabling CDP, bootp, fingerd, httpd, etc.

```

ericconrad — ssh student@192.168.198.13...
Securing Management plane services...

Disabling service finger
Disabling service pad
Disabling udp & tcp small servers
Enabling service password encryption
Enabling service tcp-keepalives-in
Enabling service tcp-keepalives-out
Disabling the cdp protocol

Disabling the bootp server
Disabling the http server
Disabling the finger service
Disabling source routing
Disabling gratuitous arp

Is SNMP used to manage the router? [yes/no]: █
  
```

### AutoSecure Mitigations

In addition to the actions shown above, AutoSecure also performed numerous other steps. Here is the resulting (partial) Cisco IOS configuration:

```

no service pad
service tcp-keepalives-in
service tcp-keepalives-out
service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone
service password-encryption
service sequence-numbers
security authentication failure rate 10 log
security passwords min-length 6
logging console critical
aaa new-model
aaa authentication login local_auth local
aaa session-id common
no ip source-route
no ip gratuitous-arps
no ip icmp rate-limit unreachable
login block-for 5 attempts 10 within 5
  
```

## DISA STIGs

- DISA is the United States DoD Defense Information Systems Agency
- They produce the Secure Technical Implementation Guides (STIGs)
  - They are freely available and exhaustive in their coverage
- U.S. government and military personnel are usually quite familiar with the STIGs, but many private sector companies are unaware of them (and therefore don't use them)
- Their guidance on switches and routers (and other devices) is concise, direct and actionable

The **DISA STIGs** are available at: <https://iase.disa.mil/stigs/Pages/index.aspx>

DISA describes the STIGs:

*The Security Technical Implementation Guides (STIGs) are the configuration standards for DOD IA and IA-enabled devices/systems. Since 1998, DISA has played a critical role in enhancing the security posture of DoD's security systems by providing the Security Technical Implementation Guides (STIGs). The STIGs contain technical guidance to "lock down" information systems/software that might otherwise be vulnerable to a malicious computer attack.<sup>1</sup>*

"IA" stands for Information Assurance.

As of course publication: there are over 300 STIGs, ranging from switches to routers to operating systems, applications, and much more.

STIG Viewer website offers a simple search function for quickly locating specific guidance:  
<https://www.stigviewer.com>

<https://iase.disa.mil/stigs/Pages/index.aspx>

## Cisco Layer 2 Benchmarks: DISA STIG High Severity I

- *The network device must require authentication for console access*
- *The switch must be configured to use 802.1x authentication on host facing access switch ports*
- *Group accounts must not be configured for use on the network device.*
- *The emergency administration account must be set to an appropriate authorization level to perform necessary administrative functions when the authentication server is not online*
- *Network devices must be password protected<sup>1</sup>*

### Cisco Layer 2 Benchmarks: DISA STIG High Severity 1

Note that the list on this slide and next cover the CAT 1 (High) findings only. There are 10 as of course publication. There are also currently 33 CAT II (medium) and 15 CAT III (low) findings.

The site <https://vaulted.io> has an excellent guide for configuring devices to meet the STIGs. This includes Cisco devices, as well as many others.

In addition to specific Cisco IOS syntax examples (example shown on the next slide), the site also offers workarounds for lowering the severity of the findings. For example, regarding "The switch must be configured to use 802.1x authentication on host facing access switch ports" finding shown above:

#### **Check Content**

*Verify if the switch configuration has 802.1x authentication implemented for all access switch ports connecting to LAN outlets (i.e. RJ-45 wall plates) or devices not located in the telecom room, wiring closets, or equipment rooms. If 802.1x authentication is not configured on these host-facing access switch ports, this is a CAT 1 finding. If MAC address filtering is implemented in lieu of 802.1x authentication, this finding will be downgraded to a CAT 3.<sup>1</sup>*

[1] [https://www.stigviewer.com/stig/layer\\_2\\_switch\\_-\\_cisco/](https://www.stigviewer.com/stig/layer_2_switch_-_cisco/)

## Cisco Layer 2 Benchmarks: DISA STIG High Severity II

- *Network devices must not have any default manufacturer passwords*
- *The network element must be configured to ensure passwords are not viewable when displaying configuration information*
- *The network device must not use the default or well-known SNMP community strings public and private*
- *The network devices must require authentication prior to establishing a management connection for administrative access*
- *The network device must use SNMP Version 3 Security Model with FIPS 140-2 validated cryptography for any SNMP agent configured on the device.<sup>1</sup>*

### Cisco Layer 2 Benchmarks: DISA STIG High Severity II

Vaulted also provides guidance for configuring the requirements shown above, including specific Cisco IOS example syntax. For example, this example configures type 5 passwords (so that plaintext type 0 passwords are not shown in the configuration). While this meets the STIG requirements, type 8 or 9 passwords would be even better (as discussed previously).

#### *Fix Text*

*Configure the network element to ensure passwords are not viewable when displaying configuration information.*

```
Device (config) # service password  
Device (config) # username name secret S3cr3T!  
Device (config) # enable secret $MyS3cr3TPW$  
Device (config) # end1
```

[1] [https://www.stigviewer.com/stig/layer\\_2\\_switch\\_-\\_cisco/](https://www.stigviewer.com/stig/layer_2_switch_-_cisco/)



## Layer 3 Benchmarks

- The Center for Internet Security (CISecurity) offers a broad range of security benchmarks
  - Switches, routers, and much more
  - See notes for details
- Router benchmarks include:
  - Cisco IOS 12
  - Cisco IOS 15
  - Juniper JUNOS 8.X.9.X.10.X (currently unsupported, the last update was 2010)
- Additional benchmarks are offered by vendors, DISA (Defense Information Systems Agency) STIGs (Security Technical Implementation Guide), and others
- Most switch benchmarks cover both routing and switching, so we'll discuss both in detail in the upcoming layer 3 (routing) section

### Layer 3 Benchmarks

**CISecurity does fine work, notably with their popular list of benchmarks. They have recently moved many behind a 'registration wall,' requiring free registration before they'll email a link to their current PDFs.**

You may register for free downloads by registering here: <https://www.cisecurity.org/unsupported-cis-benchmarks/>

The active list of benchmarks (as of late 2017) includes Distribution Independent Linux, Windows Desktops, Debian Linux, Ubuntu Linux, Amazon Linux, CentOS Linux, Oracle Linux, SUS Linux, Apple OS, IBM AIX, Windows Server, IIS, VMware, MongoDB, IBM DB2, Bind, Apache Tomcat, Apache HTTP, Docker, Oracle Database, Kubernetes, MIT Kerberos, Oracle MySQL, Amazon Web (cloud), Apple iOS, Google Android, Cisco devices, Palo Alto Firewalls, Microsoft Office, Google Chrome, Internet Explorer, and Mozilla Firefox.

CISecurity has also reduced their formerly exhaustive list of benchmarks, moving many older ones to the unsupported list. Unfortunately, that includes the Juniper JUNOS 8-10 benchmarks.

The unsupported benchmarks may be freely downloaded here: <https://www.cisecurity.org/cis-benchmarks/unsupported-cis-benchmarks/>

## CIS Cisco IOS Benchmark

- The Center for Internet Security provides the following router benchmarks:
  - CIS Cisco IOS 12 Benchmark
  - CIS Cisco IOS 15 Benchmark
  - Juniper JUNOS 8.x / 9.x / 10.x
- The benchmarks are broken up into sections:
  - Management plane
  - Control Plane
  - Data Plane
- The benchmarks are categorized as level 1 or level 2

### CIS Cisco IOS Benchmark

The CISecurity benchmarks are **high** quality and provide specific syntax for performing each step, and free. For example, here is a subset of their syntax for securing SSH on Cisco routers:

**Remediation:**

Generate an RSA key pair for the router.

```
hostname (config) #crypto key generate rsa general-keys modulus 2048
```

**Impact:**

Organizations should plan and implement enterprise network cryptography and generate an appropriate RSA key pair, such as 'modulus', greater than or equal to 2048.

**Remediation:**

Configure the SSH timeout:

```
hostname (config) #ip ssh authentication-retries [3]
```

**Impact:**

Organizations should implement a security policy limiting the number of authentication attempts for network administrators and enforce the policy through the 'ip ssh authentication-retries' command. <sup>1</sup>

[1] <https://www.cisecurity.org/benchmark/cisco/>

## CISSecurity Level 1 and Level 2 Benchmarks

- *Level 1 - Items in this profile intend to:*
  - *be practical and prudent*
  - *provide a clear security benefit; and*
  - *do not inhibit the utility of the technology beyond acceptable means*
- *Level 2 - This profile extends the "Level 1" profile. Items in this profile exhibit one or more of the following characteristics:*
  - *are intended for environments or use cases where security is paramount*
  - *acts as a defense in depth measure.*
  - *may negatively inhibit the utility or performance of the technology.<sup>1</sup>*
- **Organizations should strive to complete all level 1 benchmarks, and complete as many level 2 as beneficial**

### CISSecurity Level 1 and Level 2 Benchmarks

Level 1 benchmarks are universal best practices that should be followed in nearly all cases. Level 2 benchmarks provide meaningful security but could be problematic for some organizations or topologies.

For example: the SNMPv3 recommendations provided by CISSecurity are level 2. Why? Many organizations have legacy equipment, as well as simpler equipment such as SOHO (Small Office/Home Office) devices that do not support SNMPv3. Requiring SNMPv3 would mean these devices could not be monitored via SNMPv3, meaning requiring SNMPv3 could cause more security problems than it solves.

The same is true for NTP authentication: older and simpler equipment may not support that functionality.

They also provide specific level 2 guidance for routing protocols such as EIGRP (Extended Interior Gateway Routing Protocol) and OSPF (Open Shortest Path First). They will not apply for organizations that don't use one or the other.

[1] <https://www.cisecurity.org/benchmark/cisco/>

### Layer 3 Auditing Tools

- CISecurity's (free) Router Audit Tool is now end-of-life
  - Their newer tool is CIS-CAT Pro, which audits against 80+ benchmarks
  - This tool requires a paid CIS SecureSuite Membership. Prices begin at \$1,320/year for organizations with less than 50 employees
- Nipper (formerly CiscoParse) has split into two versions: Nipper Studio (commercial) and nipper-ng (open source)
  - Nipper Studio (commercial) cost begins at \$1,045.00/year for 25 devices, available at: <https://www.titania.com/products/nipper-studio>
  - nipper-ng source code is available at: <https://github.com/arpitn30/nipper-ng>

### Layer 3 Auditing Tools

As noted above, the free Router Audit Tool is end-of-life, and links to the source code from the project page (<http://ncat.sourceforge.net/>) are currently broken. Information about CIS-CAT Pro is available at: <https://learn.cisecurity.org/cis-cat-landing-page>

Fortunately, the source for the open source version of Nipper (released in 2013) is available via the project page at: <https://sourceforge.net/projects/nipper/>. Described from the Nipper README:

- *Cisco Switches (IOS)*
- *Cisco Routers (IOS)*
- *Cisco Firewalls (PIX, ASA, FWSM)*
- *Cisco Catalysts (NMP, CatOS, IOS)*
- *Cisco Content Service Switches (CSS)*
- *Juniper NetScreen Firewalls (ScreenOS)*
- *CheckPoint Firewall-1 (FW1)*
- *Nortel Passport Devices*
- *SonicWALL SonicOS Firewalls*

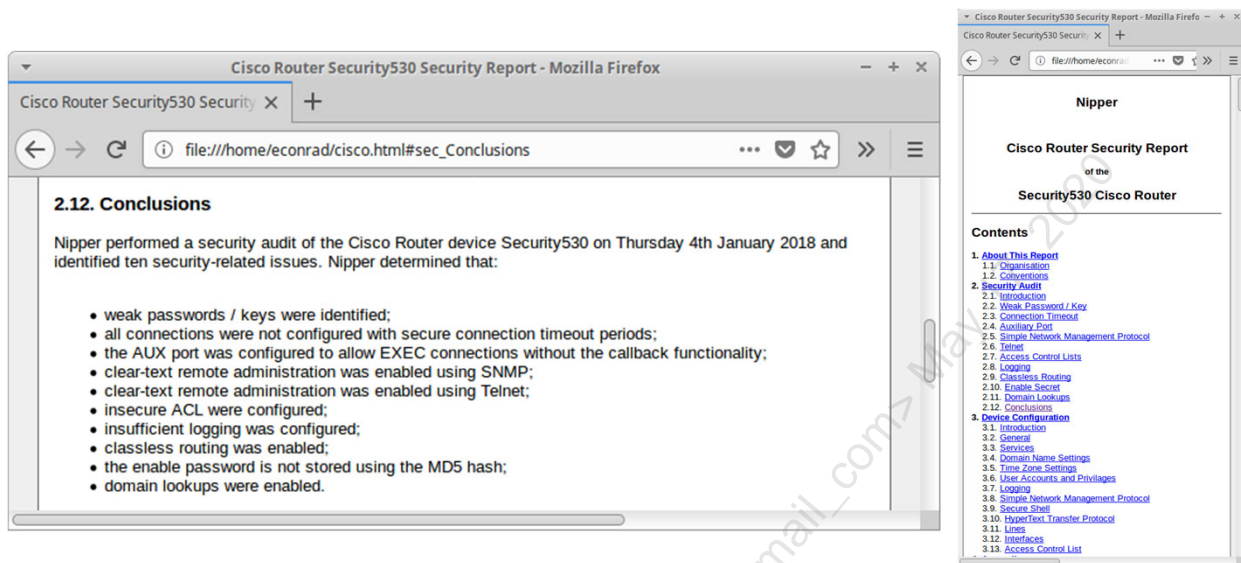
The output from nipper is in HTML, Latex, XML and Text.<sup>1</sup>

Information about Nipper Studio is available from: <https://www.titania.com/products/nipper-studio>

[1] <https://sourceforge.net/projects/nipper/>



## Nipper-ng Report



### Nipper-ng Report

A subset of the Nipper-NG HTML report is shown above.

In addition to the high-level summary guidance, Nipper-NG also provides specific syntax advice. For example, here is the SNMP guidance:<sup>1</sup>

#### 2.5. Simple Network Management Protocol

**Observation:** Simple Network Management Protocol (SNMP) is used to assist network administrators in monitoring and managing a wide variety of network devices. There are three main versions of SNMP in use. Versions 1 and 2 of SNMP are both secured with a community string and authenticate and transmit network packets without any form of encryption. SNMP version 3 provides several levels of authentication and encryption. The most basic level provides a similar protection to that of the earlier protocol versions. However, SNMP version 3 can be configured to provide encrypted authentication (auth) and secured further with support for encrypted data communications (priv).

Nipper determined that SNMP protocol version 1 was configured on Security530.

**Impact:** Due to the unencrypted nature of SNMP protocol versions 1 and 2c, an attacker who was able to monitor network traffic could capture device configuration settings, including authentication details.

**Ease:** Network packet monitoring and capture tools are widely available on the Internet and SNMP tools are included as standard with some operating systems.

**Recommendation:** Nipper recommends that, if possible, SNMP version 1 be disabled. Furthermore, Nipper recommends that, if SNMP is required, protocol version 3 be configured with Auth and Priv authentication. SNMP protocol version 1 can be disabled with the following command for each community string:

```
no snmp-server community {Community String} {[RO] | [RW]}
```

SNMP version 3 Auth and Priv access can be configured with the following commands:

```
snmp-server group {Group Name} v3 priv
```

```
snmp-server user {Username} {Group Name} v3 auth md5 {Auth Keyword} priv {[3des] | [aes 128] | [aes 192]} {Priv Keyword}
```

[1] <https://github.com/arpitn30/nipper-ng>

## Auditing Tool Pro Tip

- Audit your switches and routers before beginning remediation
- A course author began applying the Cisco CIS Benchmarks to a large enterprise with 400+ routers
  - Goal: apply all CIS level 1 recommendations, and most level 2
  - He generated CIS' Router Audit Tool (RAT) two-thirds of the way through the project (average score was 73%)
  - The goal was achieved with an average final score of 96%
- Lesson learned: run the tool before any changes are made, report the score to management, and then report weekly progress
  - Otherwise: management may not appreciate (or fund) the work required to get to 96%

### Auditing Tool Pro Tip

A course author made a mistake described on the slide above. Had he generated an average CIS RAT score on day 1, it would have probably been in the 30-40% range.

He should have done that and reported the following to management; the bad news is, we have a failing score on switch and router security. The good news: I have a plan to address the problem, and I will report back weekly.

The 30-40% score would have increased weekly, showing management the effort required to improve security. Absent that reporting: management may feel that security is simple, easy, and/or inexpensive. Information security is none of those things.

The author's CIO was once overheard saying (referring to the information security team): "Why do we have that team? We haven't had any serious security problems in years!"

That's a risk of silently solving difficult problems.

## Course Roadmap

- Day 1: Defensible Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- Day 5: Zero Trust Architecture
- Day 6: Capstone: Hands-On Defend the Flag Challenge

### NETWORK SECURITY ARCHITECTURE

1. Layer 3 Attacks and Mitigation
2. Switch and Router Benchmarks
3. **EXERCISE: Auditing Router Security**
4. Securing SNMP
5. Securing NTP
6. Bogon Filtering
7. Blackholes and Darknets
8. **EXERCISE: Router SNMP Security**
9. IPv6
10. IPv6 Misconceptions
11. Securing IPv6
12. **EXERCISE: IPv6**
13. Layer 3/4 Stateful Firewall
14. Web Proxy
15. SMTP Proxy
16. **EXERCISE: Proxy Power**
17. 530.2 Summary

### Course Roadmap

We will next conduct an exercise on Router Security.





## Exercise 2.1: Auditing Router Security

- Exercise 2.1 is in the digital wiki found in your student VM (recommended)
- Alternatively, you may use your Workbook

### SEC530 Exercise: Auditing Router Security

Please go to the SEC530 lab workbook, section 2.1.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

## Course Roadmap

- Day 1: Defensible Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- Day 5: Zero Trust Architecture
- Day 6: Capstone: Hands-On Defend the Flag Challenge

### NETWORK SECURITY ARCHITECTURE

1. Layer 3 Attacks and Mitigation
2. Switch and Router Benchmarks
3. EXERCISE: Auditing Router Security
4. **Securing SNMP**
5. Securing NTP
6. Bogon Filtering
7. Blackholes and Darknets
8. EXERCISE: Router SNMP Security
9. IPv6
10. IPv6 Misconceptions
11. Securing IPv6
12. EXERCISE: IPv6
13. Layer 3/4 Stateful Firewall
14. Web Proxy
15. SMTP Proxy
16. EXERCISE: Proxy Power
17. 530.2 Summary

### Course Roadmap

We will next discuss securing SNMP.

## Securing SNMP

- SNMP (Simple Network Management Protocol) can represent a significant security vulnerability
- All version prior to SNMPv3 expose the plaintext community string on the network
  - SNMPv2c is the most commonly-deployed version of SNMP
- **SNMP Read community string:** allows read access to the SNMP-enabled device
- **SNMP Write community string:** allows write access (meaning the ability to change) SNMP-enabled device
  - Also allows downloading the complete IOS configuration on Cisco routers and switches

### Securing SNMP

SNMP is an often overlooked, and highly dangerous protocol. SNMP version 2c is the most commonly-deployed version, which uses plaintext community strings. These strings often zip around a network thousands of times per day, sent from network monitoring systems, often every five minutes to hundreds of devices, 24/7/365.

SNMP community strings are effectively passwords. Default strings (such as public and private for read and write access, respectively) are quite common, as are well-known vendor default strings.

New devices are sometimes automatically polled on a network (meaning the SNMP read string is sent to new devices). An attacker in control of a device that is polled may be able to sniff the community strings, and then use them to leverage deeper access into a network.

## SNMP Attack: Guess the Community Strings

- The Metasploit penetration testing framework includes a number of SNMP modules
- We used the `snmp_logon` auxiliary module to guess the SNMP RW community string
  - This module includes a wordlist of 120 common SNMP community strings
- The SNMP RW community for this router is "xyzzzy"

```

Terminal - econrad@Security530: ~
File Edit View Terminal Tabs Help

Metasploit

=[ metasploit v4.13.8-dev ]
+ -- --[ 1607 exploits - 914 auxiliary - 278 post ]
+ -- --[ 471 payloads - 39 encoders - 9 nops ]
+ -- --[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > use auxiliary/scanner/snmp/snmp_logon
msf auxiliary(snmp_logon) > set RHOSTS 10.99.99.250
RHOSTS => 10.99.99.250
msf auxiliary(snmp_logon) > run

[*] No active DB -- Credential data will not be saved!
[*] 10.99.99.250:161 - LOGIN SUCCESSFUL: xyzzzy (Access level:
read-write); Proof (sysDescr.0): Cisco IOS Software, C870 Soft
ware (C870-ADVIPSERVICESK9-M), Version 12.4(4)T6, RELEASE 50FT
WARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Sat 11-Nov-06 00:28 by kellythw
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

msf auxiliary(snmp_logon) >

```

### SNMP Attack: Guess the Community Strings

The screenshot above shows Metasploit (<https://metasploit.com>), using the `snmp_logon` auxiliary script. It uses a 120-entry SNMP community string wordlist by default. The same wordlist is available via Daniel Miessler's excellent SecLists GitHub site at:

<https://github.com/danielmiessler/SecLists/blob/master/Discovery/SNMP/common-snmp-community-strings.txt>

Here we launched the same attack via nmap, using the SecList SNMP community string wordlist:

```
$ sudo nmap -sU -p 161 --script snmp-brute --script-args snmp-brute.communitiesdb=wordlist-common-snmp-community-strings.txt 10.99.99.250
```

```

Terminal - econrad@Security530: ~
File Edit View Terminal Tabs Help

[~]$ sudo nmap -sU -p 161 --script snmp-brute --script-args snmp-brute.
communitiesdb=wordlist-common-snmp-community-strings.txt 10.99.99.250

Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-04 11:55 EST
Nmap scan report for 10.99.99.250
Host is up (0.00038s latency).

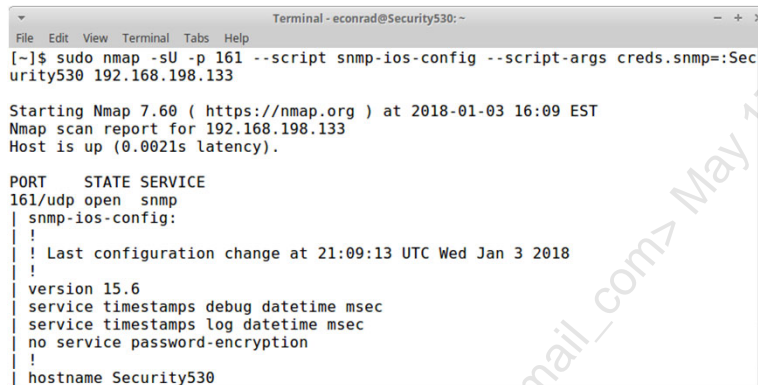
PORT      STATE SERVICE
161/udp   open  snmp
| snmp-brute:
|_ xyzzzy - Valid credentials

Nmap done: 1 IP address (1 host up) scanned in 6.64 seconds
[~]$

```

## Cisco SNMP Attack: Download the Cisco IOS Configuration

- Access to the SNMP read/write string on a Cisco switch or router allows downloading the Cisco IOS configuration, including any passwords and/or password hashes



```

Terminal - econrad@Security530: ~
File Edit View Terminal Tabs Help
[~]$ sudo nmap -sU -p 161 --script snmp-ios-config --script-args creds.snmp=Security530 192.168.198.133

Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-03 16:09 EST
Nmap scan report for 192.168.198.133
Host is up (0.0021s latency).

PORT      STATE SERVICE
161/udp   open  snmp
| snmp-ios-config:
| !
| ! Last configuration change at 21:09:13 UTC Wed Jan 3 2018
| !
| version 15.6
| service timestamps debug datetime msec
| service timestamps log datetime msec
| no service password-encryption
| !
| hostname Security530

```

### Cisco SNMP Attack: Download the Cisco IOS Configuration

The screenshot shown above shows the nmap running the NSE (Nmap Scripting Engine) script 'snmp-ios-config'.

Here's the full command, run against a router at 192.168.198.133, with a write SNMP community string of "Security530":

```
$ sudo nmap -sU -p 161 --script snmp-ios-config --script-args creds.snmp=Security530 192.168.198.133
```

This downloads the entire Cisco IOS configuration, including the usernames, as well as any passwords or hashes, as we will show on the next slide.

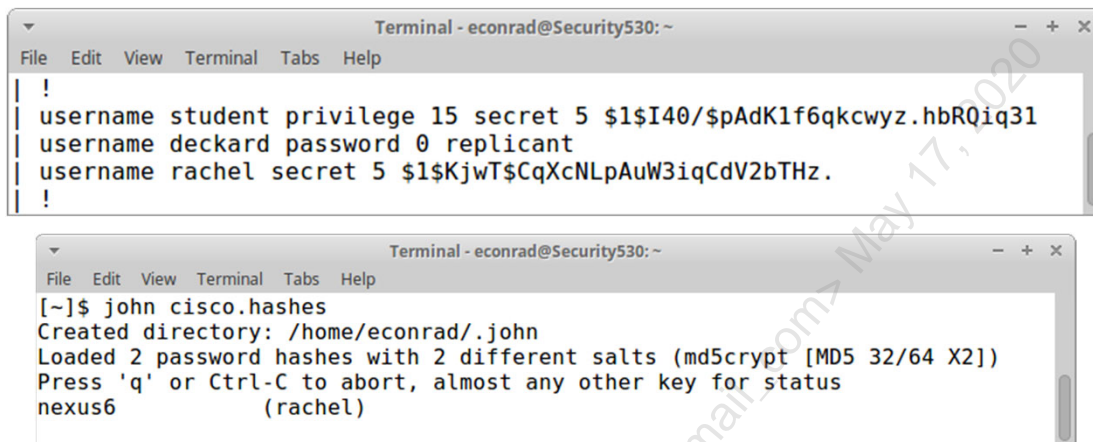
Penetration testers often begin this attack by guessing or brute forcing the SNMP community string. Nmap also has the "snmp-brute" NSE script, which:

*Attempts to find an SNMP community string by brute force guessing. This script opens a sending socket and a sniffing pcap socket in parallel threads. The sending socket sends the SNMP probes with the community strings, while the pcap socket sniffs the network for an answer to the probes. If valid community strings are found, they are added to the creds database and reported in the output.<sup>1</sup>*

[1] <https://nmap.org/nsedoc/scripts/snmp-brute.html>

## Passwords Exposed via SNMP Attack

- The Cisco IOS configuration downloaded via SNMP contained type 0 (plaintext) and type 5 (salted MD5 hashes)

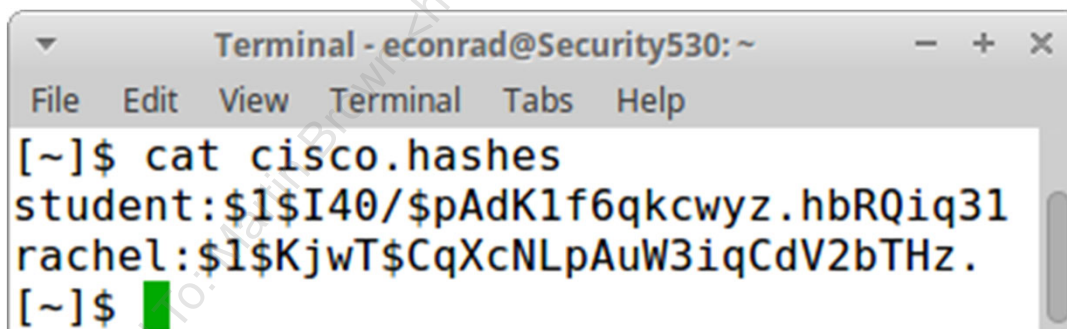


```
Terminal - econrad@Security530: ~
File Edit View Terminal Tabs Help
!
username student privilege 15 secret 5 $1$I40/$pAdK1f6qkcwyz.hbRQiq31
username deckard password 0 replicant
username rachel secret 5 $1$KjwT$CqXcNLpAuW3iqCdV2bTHz.
!

Terminal - econrad@Security530: ~
File Edit View Terminal Tabs Help
[~]$ john cisco.hashes
Created directory: /home/econrad/.john
Loaded 2 password hashes with 2 different salts (md5crypt [MD5 32/64 X2])
Press 'q' or Ctrl-C to abort, almost any other key for status
nexus6          (rachel)
```

### Passwords Exposed via SNMP Attack

In the screenshot above: we took the two type5 (MD5) hashes, and converted them to a shadow file-style format of username:hash, and saved the results to 'cisco.hashes'.



```
Terminal - econrad@Security530: ~
File Edit View Terminal Tabs Help
[~]$ cat cisco.hashes
student:$1$I40/$pAdK1f6qkcwyz.hbRQiq31
rachel:$1$KjwT$CqXcNLpAuW3iqCdV2bTHz.
[~]$
```

Note the hash is comprised of three fields delimited by dollar signs: hash algorithm (1, which is salted BSD), followed by the salt (which is a random number: Rachel's salt is 'KjwT'), followed by the hash itself (Rachel's hash is 'CqXcNLpAuW3iqCdV2bTHz.').

As shown above: we used the password cracking tool John the Ripper to crack the hashes. It found Rachel's password (nexus6). It had not (yet) found the student password (which happens to be 'Security530', a more complex password than rachel's).

We can then use the cracked passwords to attempt to log into the switch or router. Users often manually synchronize passwords between different systems, so we could also try to log into other devices such as Windows systems.

## Hardening SNMP

- Disable SNMP if not required
- If SNMP is required:
  - Disable SNMP write access if possible
  - Use complex (or randomly-generated) community strings
  - Use SNMP version 3 on all supported equipment
  - For non SNMPv3-capable devices that require SNMP: use SNMP version 2c with access lists that restrict polling to required servers only (such as network management and/or monitoring systems)

### Hardening SNMP

SNMP is disabled by default on modern Cisco switches and routers. This command enables read access only (version 2c):

```
Router(config)# snmp-server community <community string> RO
```

This command enables write access:

```
Router(config)# snmp-server community <community string> RW
```

These commands will restrict SNMP access from 10.5.30.0/24 via an access control list (ACL) number 30:

```
Router(config)# access-list 30 permit 10.5.30.0 0.0.0.255
Router(config)# snmp-server community READONLY RO 30
Router(config)# snmp-server community READWRITE RW 30
```

## SNMPv3

- SNMP version three offers three level of access:
  - no auth: unauthenticated access
  - auth: authenticated access via plaintext
  - priv: authenticated and encrypted access (most secure mode)
- Priv supports the following encryption algorithms:
  - Single DES, Triple DES, AES-128, AES-192, and AES-256
- The syntax for configuring SNMPv3 priv mode with AES-128 encryption is shown in the notes

### SNMPv3

Cisco devices must support the 'snmp engineID' command to use SNMPv3, here is sample syntax and output:

```
Router# show snmp engineID
Local SNMP engineID: 8000000903000045B8F7BE00
Remote Engine ID      IP-addr      Port
```

The create the SNMPv3 priv group (called "PRIVGROUP"):

```
Router(config)# snmp-server group PRIVGROUP v3 priv
```

Next: create SNMPv3 user 'student', use MD5 authentication with a password of <auth password> , and use AES-128 encryption with a password of '<priv password>'.

```
Router(config)# snmp-server user student PRIVGROUP v3 auth md5 <auth
password> priv aes 128 <priv password>
```



## Course Roadmap

- Day 1: Defensible Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- Day 5: Zero Trust Architecture
- Day 6: Capstone: Hands-On Defend the Flag Challenge

### NETWORK SECURITY ARCHITECTURE

1. Layer 3 Attacks and Mitigation
2. Switch and Router Benchmarks
3. EXERCISE: Auditing Router Security
4. Securing SNMP
5. **Securing NTP**
6. Bogon Filtering
7. Blackholes and Darknets
8. EXERCISE: Router SNMP Security
9. IPv6
10. IPv6 Misconceptions
11. Securing IPv6
12. EXERCISE: IPv6
13. Layer 3/4 Stateful Firewall
14. Web Proxy
15. SMTP Proxy
16. EXERCISE: Proxy Power
17. 530.2 Summary

### Course Roadmap

We will next discuss securing NTP.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

## Securing NTP

- Clock skew can create considerable issues
  - Especially during incident handling and forensic investigations
  - Authentication frameworks, including Kerberos, can be affected by serious clock skew
- NTP (Network Time Protocol) is used to set clocks on networked computers
  - It uses UDP port 123
- SNTP (Simple Network Time Protocol) is sometimes used
  - The same basic protocol as NTP, but with much of the complexity removed
  - SNTP is also less accurate than NTP
  - Windows 2000 and XP's Windows Time service (W32Time) used SNTP
  - Newer versions of Windows use NTP

### Securing NTP

Clock skew can create serious operational issues, ranging from authentication to investigations. Building a forensic timeline among systems with different local times can be quite challenging.

Kerberos can also be affected: Microsoft recommends a maximum skew of 5 minutes:

*To prevent replay attacks, the Kerberos protocol uses time stamps as part of its definition. For time stamps to work properly, the clocks of the client computer and the domain controller need to be synchronized as closely as possible. Because the clocks of two computers are often out of sync, administrators can use this policy setting to establish the maximum acceptable difference to the Kerberos protocol between a client computer clock and domain controller clock. If the difference between a client computer clock and the domain controller clock is less than Maximum tolerance for computer clock synchronization, any time stamp that is used in a session between the two computers is considered authentic.*

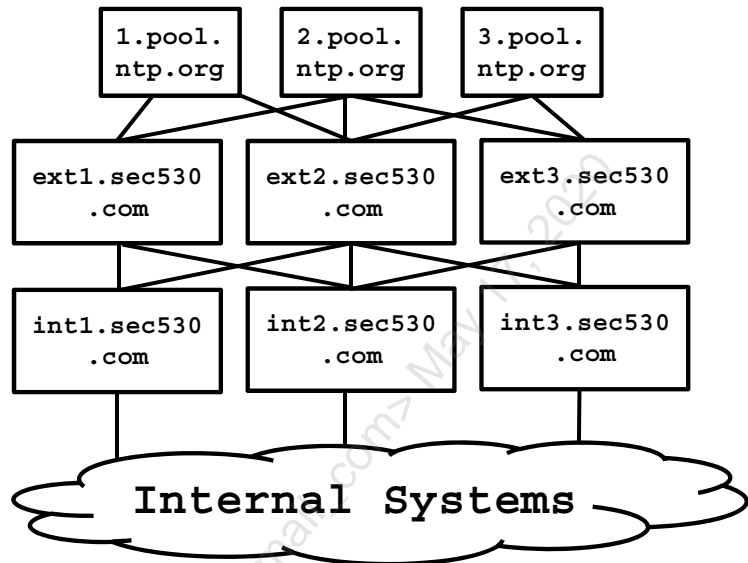
#### **Best practices**

- *It is advisable to set Maximum tolerance for computer clock synchronization to a value of 5 minutes.*

[1] [https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/jj852172\(v=ws.11\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/jj852172(v=ws.11))

## NTP Design

- Point a number of internet-facing external devices to pool.ntp.org
- Then point internal servers (such as AD controllers) to the internet-facing devices
- Then point all other internal systems to the internal servers



### NTP Design

The diagram above shows a suggested NTP design. Internet-facing external systems point to pool.ntp.org. Then a number of internal servers point to the internet-facing servers. Finally, the remaining internal systems point to the internal NTP servers. ntp.org describes how pool.ntp.org works:

*pool.ntp.org uses DNS round robin to make a random selection from a pool of time servers who have volunteered to be in the pool. This is usually good enough for end-users.*

*The minimal ntpd configuration file (e.g. /etc/ntp.conf) for using pool.ntp.org is:*

```

driftfile /var/lib/ntp/ntp.driftserver
0.pool.ntp.orgserver
1.pool.ntp.orgserver
2.pool.ntp.orgserver
3.pool.ntp.org
  
```

*The NTP Pool DNS system automatically picks time servers which are geographically close for you, but if you want to choose explicitly, there are sub-zones of pool.ntp.org.<sup>1</sup>*

[1] <http://support.ntp.org/bin/view/Servers/NTPPoolServers>

## NTP Authentication

- NTP is sent over UDP, which may be spoofed
  - It can also be vulnerable to man-in-the-middle attacks
- NTP supports authentication
  - The public pool.ntp.org web servers do not use authentication
- NIST provides free public authenticated NTP service
  - Requires registration by snail mail or fax(!)
  - See notes for details
- Another option: purchasing stratum one time servers, and syncing to them locally
- The device shown below is a stratum one NTP server, available for under \$300 US



### NTP Authentication

Note that a stratum 1 NTP server contains an onboard clock, such as **an atomic clock, or GPS-based clock**. An NTP server that syncs to a stratum 1 server becomes a stratum 2 server. A system that syncs to a stratum 2 server becomes a stratum 3 server, etc.

NIST describes their authenticated NTP service:

*The service will be provided at no charge, and user keys may be used to connect to any of the servers whose addresses are listed below. Additional hardware will be added in the future if the demand for the service is sufficiently great to warrant it.*

*Users who wish to use this service should send a letter to NIST using the US mail or FAX machine (e-mail is not acceptable).<sup>1</sup>*

For more details (including the mailing address and fax number), please see the NIST link below.

The device shown in the slide above is a "TimeMachines, NTP Network Time Server with GPS, TM1000A, A GPS Antenna maintains current time broadcast by U.S. Satellites"<sup>2</sup>. It is a stratum 1 GPS NTP server.

[1] <https://www.nist.gov/pml/time-and-frequency-division/time-services/nist-authenticated-ntp-service>

[2] <https://www.amazon.com/TimeMachines-TM1000A-maintains-broadcast-Satellites/dp/B002RC3Q4Q>

## NTP Amplification Attacks

- UDP-based services can sometimes be used for spoofed Denial of Server (DoS) attacks
- NTP supports a 'monlist' command, which will return the client IP addresses that have synced most recently
  - Up to 600 addresses can be sent
- The attacker can then send a spoofed NTP monlist command to a vulnerable server
  - In a recent test by Cloudflare<sup>1</sup>, one spoofed 234-byte UDP packet resulted in 100 response packets, totaling 48,000 bytes
  - Resulting in an amplification factor of 206 times

### NTP Amplification Attacks

Cloudflare describes the attack described above:

*At the command line, I typed*

```
ntpdc -c monlist 1xx.xxx.xxx.xx9
```

*to send the MON\_GETLIST command to the server at 1xx.xxx.xxx.xx9. The request packet is 234 bytes long. The response is split across 10 packets totaling 4,460 bytes. That's an amplification factor of 19x and because the response is sent in many packets an attack using this would consume a large amount of bandwidth and have a high packet rate.*

*This particular NTP server only had 55 addresses to tell me about. Each response packet contains 6 addresses (with one short packet at the end), so a busy server that responded with the maximum 600 addresses would send 100 packets for a total of over 48k in response to just 234 bytes. That's an amplification factor of 206x!*

*An attacker, armed with a list of open NTP servers on the Internet, can easily pull off a DDoS attack using NTP. And NTP servers aren't hard to find. Common tools like Metasploit and NMAP have had modules capable of identifying NTP servers that support monlist for a long time.<sup>2</sup>*

[1] <https://blog.cloudflare.com/understanding-and-mitigating-ntp-based-ddos-attacks/>

[2] *ibid.*

## Monlist DoS

- This PCAP shows an NTP monlist DoS victim receiving 482-byte responses
- The attacker sent a 234-byte spoofed UDP/NTP request
- This ntp.conf setting blocks monlist:  
`disable monitor`

No	Time	Source	Destination	Protoc	Leng	Info
1	0.000000	216.86.138.141	24.86.162.1...	NTP	482	NTP Version 2, private
2	0.000064	125.152.1.118	24.86.162.1...	NTP	482	NTP Version 2, private
3	0.000104	176.197.82.140	24.86.162.1...	NTP	482	NTP Version 2, private
4	0.000141	118.141.208.9	24.86.162.1...	NTP	482	NTP Version 2, private
5	0.000180	83.142.184.237	24.86.162.1...	ICMP	70	Destination unreachable
6	0.000210	47.22.1.216	24.86.162.1...	NTP	482	NTP Version 2, private
7	0.000248	47.22.1.216	24.86.162.1...	NTP	122	NTP Version 2, private
8	0.000285	113.160.224.80	24.86.162.1...	NTP	482	NTP Version 2, private

▶ User Datagram Protocol, Src Port: 123, Dst Port: 6667  
 ▼ Network Time Protocol (NTP Version 2, private)  
 ▶ Flags: 0xd7, Response bit: Response, Version number: NTP Version 2, Mode: r  
 ▶ Auth, sequence: 215  
 Implementation: XNTPD (3)  
 Request code: MON\_GETLIST\_1 (42)  
 0000 .... = Err: No error (0x00)  
 .... 0000 0000 0110 = Number of data items: 6  
 0000 .... = Reserved: 0x00  
 ... 0000 0100 1000 = Size of data item: 0x0048  
 ▶ Monlist item: address: 69.28.57.108:10031  
 ▶ Monlist item: address: 66.215.226.12:80  
 ▶ Monlist item: address: 182.188.159.162:80  
 ▶ Monlist item: address: 110.33.122.171:3074  
 ▶ Monlist item: address: 64.40.6.14:50557  
 ▶ Monlist item: address: 62.235.151.102:3074

### Monlist DoS

This command will send the NTP monlist command:

```
$ ntpdc -c monlist 10.5.30.119
```

There is also a Nmap NSE (Nmap Scripting Engine) script<sup>1</sup>:

```
$ nmap -sU -pU:123 -Pn -n --script=ntp-monlist 10.5.30.119
```

Note that there is no direct way for the victim identify the system sending the spoofed requests.

Monlist may also be used by black hats who have gained internal access to a network, providing a list of local peers that have synced their clocks:

*In this case, the attackers are taking advantage of the monlist command. Monlist is a remote command in an older version of NTP that sends the requester a list of the last 600 hosts who have connected to that server. For attackers, the monlist query is a great reconnaissance tool. For a localized NTP server, it can help to build a network profile. However, as a DDoS tool, it is even better because a small query can redirect megabytes worth of traffic.<sup>2</sup>*

[1] <https://nmap.org/nsedoc/scripts/ntp-monlist.html>

[2] <https://www.symantec.com/connect/blogs/hackers-spend-christmas-break-launching-large-scale-ntp-reflection-attacks>

## Course Roadmap

- Day 1: Defensible Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- Day 5: Zero Trust Architecture
- Day 6: Capstone: Hands-On Defend the Flag Challenge

### NETWORK SECURITY ARCHITECTURE

1. Layer 3 Attacks and Mitigation
2. Switch and Router Benchmarks
3. EXERCISE: Auditing Router Security
4. Securing SNMP
5. Securing NTP
6. **Bogon Filtering**
7. Blackholes and Darknets
8. EXERCISE: Router SNMP Security
9. IPv6
10. IPv6 Misconceptions
11. Securing IPv6
12. EXERCISE: IPv6
13. Layer 3/4 Stateful Firewall
14. Web Proxy
15. SMTP Proxy
16. EXERCISE: Proxy Power
17. 530.2 Summary

### Course Roadmap

We will next discuss Bogon filtering.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

## Bogon Filtering

- Bogons are network blocks that are not routed on the internet
    - Includes RFC1918 address, unallocated addresses, and others
    - These addresses are often used in DDoS attacks
    - The current list is on the right
  - Team Cymru publishes the updated list in various formats
    - Dotted Decimal, Bit (CIDR) Notation, and others
      - <http://www.team-cymru.org/bogon-reference.html>
- 0.0.0.0/8
  - 10.0.0.0/8
  - 100.64.0.0/10
  - 127.0.0.0/8
  - 169.254.0.0/16
  - 172.16.0.0/12
  - 192.0.0.0/24
  - 192.0.2.0/24
  - 192.168.0.0/16
  - 198.18.0.0/15
  - 198.51.100.0/24
  - 203.0.113.0/24
  - 224.0.0.0/4
  - 240.0.0.0/4

### Bogon Filtering

Team Cymru defines a bogon, "A bogon prefix is a route that should never appear in the Internet routing table. A packet routed over the public Internet (not including over VPNs or other tunnels) should never have a source address in a bogon range. These are commonly found as the source addresses of DDoS attacks." [1]

**What types of traffic uses bogon source addresses? DDoS attacks, as well as other forms of malware frequently do.** Another common example is misconfigured traffic, sometimes sent by malfunctioning NAT gateways that aren't properly translating addresses from RFC1918 to public.

There is no business value in accepting this traffic, since it is bogus (hence the name 'bogon'), and there is no way to send a response.

[1] <https://www.team-cymru.com/bogon-reference.html>



## Where to Configure a Bogon Filter

- Most organizations configure the bogon filter in their outermost external router, dropping inbound traffic sent from the internet with a bogon source addresses
  - The external firewall may also be used
  - It's a simpler routing decision since it uses the IP address only, but either routers or firewalls may be used
- Also consider dropping spoofed traffic sent 'from' the company's internal addresses, from the internet
- Team Cymru makes templates available for a variety of devices, see notes for details
- The bogon list is updated periodically, so add calendar reminders to check for list updates

### Where to Configure a Bogon Filter

The bogon filter is configured on the external (internet-facing) interface of an edge router or firewall, dropping traffic sent from the internet with a matching source address. For companies that use non-private IP addresses on their internal networks: add those addresses to the bogon filter as well. This will drop spoofed external traffic sent from the internet, with a forged (internal) source address.

The bogon list used to change frequently in the past, as unallocated network blocks were issued. The rate of change has slowed dramatically since February 2011. That was when the remaining five legacy class A networks were issued:

*The Number Resource Organization (NRO) announced today that the free pool of available IPv4 addresses is now fully depleted. On Monday, January 31, the Internet Assigned Numbers Authority (IANA) allocated two blocks of IPv4 address space to APNIC, the Regional Internet Registry (RIR) for the Asia Pacific region, which triggered a global policy to allocate the remaining IANA pool equally between the five RIRs. Today IANA allocated those blocks. This means that there are no longer any IPv4 addresses available for allocation from the IANA to the five RIRs.[1]*

Team Cymru maintains a list of secure templates for a variety of devices and services, including Cisco IOS, Juniper, BIND, NTP, and others. They include bogon blocking templates (where applicable, such as on routers or firewalls). The templates are available at: <http://www.team-cymru.org/templates.html>

[1] <https://www.nro.net/ipv4-free-pool-depleted/>

## Cisco IOS Bogon Filter Configuration

- Create the access list:

```
Router(config)#ip access-list extended bogons
Router(config-ext-nacl)# deny ip 10.0.0.0 0.255.255.255 any log
Router(config-ext-nacl)# deny ip 192.168.0.0 0.0.255.255 any log
Router(config-ext-nacl)# deny ip 172.16.0.0 0.15.255.255 any log
Router(config-ext-nacl)# deny ip 169.254.0.0 0.0.255.255 any log
Router(config-ext-nacl)# deny ip 192.0.2.0 0.0.0.255 any log
(etc..)
Router(config-ext-nacl)# permit ip any any
```

- Then apply it to an interface (see notes)

### Cisco IOS Bogon Filter Configuration

The complete list of bogons is shown on the previous "Bogon Filtering slide." and is also available from Team Cymru at: <http://www.team-cymru.org/bogon-reference.html>

**Note that the access list shown above ends with "permit ip any any" (which allows all traffic that was not explicitly denied).** That this may not be appropriate in all cases (where other traffic needs to be dropped). **It is quite common for an external (border) router to use this ACL, as the firewall behind it is typically used as the primary filtering device.**

Assuming the internet interface is gigabitEthernet0/0, these commands will apply the bogon filter:

```
Router(config)# interface gigabitEthernet0/0
Router(config-if)# ip access-group bogons in
```

## Course Roadmap

- Day 1: Defensible Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- Day 5: Zero Trust Architecture
- Day 6: Capstone: Hands-On Defend the Flag Challenge

### NETWORK SECURITY ARCHITECTURE

1. Layer 3 Attacks and Mitigation
2. Switch and Router Benchmarks
3. EXERCISE: Auditing Router Security
4. Securing SNMP
5. Securing NTP
6. Bogon Filtering
7. **Blackholes and Darknets**
8. EXERCISE: Router SNMP Security
9. IPv6
10. IPv6 Misconceptions
11. Securing IPv6
12. EXERCISE: IPv6
13. Layer 3/4 Stateful Firewall
14. Web Proxy
15. SMTP Proxy
16. EXERCISE: Proxy Power
17. 530.2 Summary

### Course Roadmap

We will next discuss blackholes and darknets.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

## Blackholes and Darknets

- A "darknet" originally referred to unused/non-routed IP addresses owned by an organization
  - The term "darknet" has been co-opted lately by "dark web" concepts, such as TOR hidden nodes
  - We will use the term "IP darknet" to distinguish the two
- A blackhole route is used to drop traffic sent to specific IP addresses
  - Usually routed to the 'null0' interface (like /dev/null for a router)
- Blackholes and darknets are related; the difference is darknets are routed to a darknet router, where the traffic is analyzed

### Blackholes and Darknets

Googling "darknet" usually returns sites relating to the newer definition: dark websites accessed via TOR hidden nodes (a platform for providing anonymous internet services). Silk Road is the most famous "dark web" site, which openly sold illegal drugs (and more). Wired Magazine has a great summary of law enforcement's takedown of the original Silk Road.

*He took an interest in Tor, the encryption software that allowed users to visit sites such as Silk Road. Tor's protocol is a kind of digital invisibility cloak, hiding users and the sites they visit. Tor stands for "the Onion Router" and was launched by the Navy in 2002. It has since become a tool for all manner of clandestine communications, licit and illicit, from circumventing censorship in countries like China to powering contraband sites like Silk Road. Tor's encryption is so layered, agents thought it was unbreakable. When cybercrime investigations hit a Tor IP, they would give up. The supposed impossibility only attracted Tarbell. I'm gonna take on Tor, he thought.<sup>1</sup>*

We will use the term IP darknet to avoid confusion, referring to unused IP addresses/network blocks. Most organizations simply ignore network blocks that they own but do not use. That amounts to a missed opportunity: it's better to route traffic to unused networks to a device that will count and/or analyze it, and then drop it.

Our industry is full of words with repurposed and/or dual meanings, for example: hacker. Depending on who you talk to: it means explorer, someone who breaks into systems, or someone who breaks into systems maliciously.

[1] <https://www.wired.com/2015/04/silk-road-1/>

## Why Monitor IP Darknets?

- Simple: malware likes to scan
- Assume an organization is using the following IP addresses internally
  - Servers: 192.168.1.0/24
  - Clients: 192.168.2.0/24, 192.168.3.0/24, and 192.168.4.0/24
- The following networks are IP darknets in the same /16:
  - 192.168.0.0/24
  - 192.168.5.0/24 -> 192.168.255.0/24
- We recommend setting up a darknet route to those addresses and monitoring the resulting traffic
  - Watch for explosions in traffic (this can be your fastest IDS)
- Note the use of RFC1918 addresses in the example above
  - Either public or private addresses may be darknets internally and are equally useful internally

### Why Monitor IP Darknets?

The slide above discusses internal IP darknet routes: what about external IP darknets?

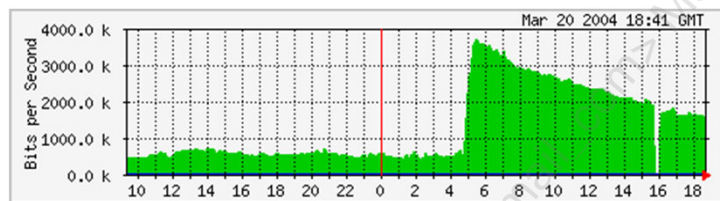
In this case, only public IP addresses are useful (since bogons are not routed publicly, including the private RFC1918 addresses).

Public IP darknets are used to collect internet attack data (and sometimes to sinkhole aggressive malware such as worms). External IP darknets are usually less actionable than internal IP darknets because worms and other forms of malware hit your external firewall all day long, while the same is (hopefully) not true for your internal networks.

The Internet Storm Center's project DShield collects data from public IP darknets, including unused dropped traffic sent to public netblocks, submitted by volunteers to the DShield project. You can learn more about DShield at: <https://isc.sans.edu/howto.html>

## What Kind of Traffic Is Sent to an IP Darknet?

- All traffic sent to a darknet is bogus, by definition
  - There are two types of darknet traffic sources: misconfigured and/or malicious traffic
  - IP darknet monitoring can offer critical insights into misconfigured and/or malicious traffic on a network
- Team Cymru's IP Darknet monitor discovered the Witty worm<sup>1</sup>:



### What Kind of Traffic Is Sent to an IP Darknet?

Team Cymru has a fantastic paper called "The Darknet Project," it is well worth checking out. The screenshot above is from that paper, where they note, "The Witty worm is an example where our Darknets alerted us within minutes of the release of the worm."<sup>2</sup>

Team Cymru describes IP darknets:

*A Darknet is a portion of routed, allocated IP space in which no active services or servers reside. These are "dark" because there is, seemingly, nothing within these networks.*

*A Darknet, does in fact, include at least one server, designed as a packet vacuum. This server gathers the packets and flows that enter the Darknet, useful for real-time analysis or post-event network forensics.*

*Any packet that enters a Darknet is by its presence aberrant. No legitimate packets should be sent to a Darknet. Such packets may have arrived by mistake or misconfiguration, but the majority of such packets are sent by malware. This malware, actively scanning for vulnerable devices, will send packets into the Darknet, and this is exactly what we want.*

*Darknets have multiple uses. These can be used to host flow collectors, backscatter detectors, packet sniffers, and IDS boxes. The elegance of the Darknet is that it cuts down considerably on the false positives for any device or technology.<sup>1</sup>*

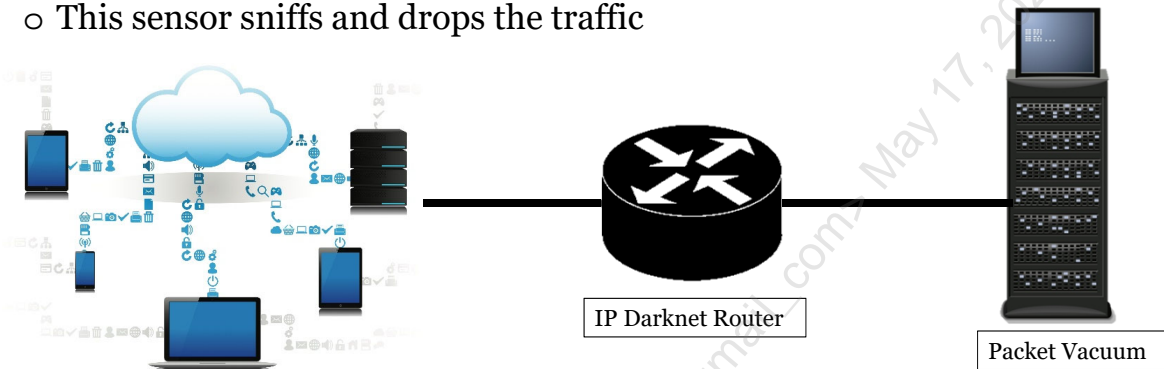
[1] <https://www.team-cymru.com/darknet.html>

[2] Ibid.

[3] Ibid.

## IP Darknet Architecture

- Route all IP darknet traffic to a dedicated darknet router
  - Monitor this traffic via SNMP
- That router forwards traffic to a 'packet vacuum' sensor
  - This sensor sniffs and drops the traffic



### IP Darknet Architecture

The first piece of the IP darknet architecture is the darknet router. The router can be lower-end: it simply needs to forward packets to a 'packet vacuum' interface, and (ideally) run SNMP, so that darknet traffic can be monitored via network monitoring solutions such as MRTG (the Multi Router Traffic Grapher, available at: <https://oss.oetiker.ch/mrtg/>). An older 10/100 router would work fine. These are available online at sites such as ebay.com for under USD \$40. A small Linux/UNIX system may also be used as the IP darknet router.

The packet vacuum contains two physical interfaces: the sniffing interface, and the management interface. All traffic sent to the sniffing interface is dropped, while Network Security Monitoring (NSM) tools are used to monitor the traffic. Options include Security Onion, or any of the tools used by it, including Snort, Suricata, Bro, SiLK, Argus, etc., etc. The management interface runs SSHD, for remote access. Traffic may also be captured with netsniff-ng, tcpdump, etc., for further analysis.

More details are available in Team Cymru's excellent *The Darknet Project*, available at: <https://www.team-cymru.com/darknet.html>

## Re-Design & Implement on 530.1 – Routers



- Harden routers according to Cisco's best practices, CISecurity benchmarks and DISA STIGs
- Disable SNMP if not required, harden it otherwise
  - Use SNMP v3
- Secure NTP
- Configure bogon filters
- Route IP darknet traffic to darknet router
- ACLs to filter inbound / outbound traffic
- Strong access controls
- Secure the control plane
- Configure privilege levels and role-based CLI access
- Provide routing update authentication
- Enable centralized logging

This page intentionally left blank.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020



## Course Roadmap

- Day 1: Defensible Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- Day 5: Zero Trust Architecture
- Day 6: Capstone: Hands-On Defend the Flag Challenge

### NETWORK SECURITY ARCHITECTURE

1. Layer 3 Attacks and Mitigation
2. Switch and Router Benchmarks
3. **EXERCISE: Auditing Router Security**
4. Securing SNMP
5. Securing NTP
6. Bogon Filtering
7. Blackholes and Darknets
8. **EXERCISE: Router SNMP Security**
9. IPv6
10. IPv6 Misconceptions
11. Securing IPv6
12. **EXERCISE: IPv6**
13. Layer 3/4 Stateful Firewall
14. Web Proxy
15. SMTP Proxy
16. **EXERCISE: Proxy Power**
17. 530.2 Summary

### Course Roadmap

We will next conduct an exercise on Router Security.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020



## Exercise 2.2: Router SNMP Security

- Exercise 2.2 is in the digital wiki found in your student VM (recommended)
- Alternatively, you may use your Workbook

### SEC530 Exercise: Router SNMP Security

Please go to the SEC530 lab workbook, section 2.2.

## Course Roadmap

- Day 1: Defensible Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- Day 5: Zero Trust Architecture
- Day 6: Capstone: Hands-On Defend the Flag Challenge

### NETWORK SECURITY ARCHITECTURE

1. Layer 3 Attacks and Mitigation
2. Switch and Router Benchmarks
3. EXERCISE: Auditing Router Security
4. Securing SNMP
5. Securing NTP
6. Bogon Filtering
7. Blackholes and Darknets
8. EXERCISE: Router SNMP Security
9. **IPv6**
10. IPv6 Misconceptions
11. Securing IPv6
12. EXERCISE: IPv6
13. Layer 3/4 Stateful Firewall
14. Web Proxy
15. SMTP Proxy
16. EXERCISE: Proxy Power
17. 530.2 Summary

### Course Roadmap

We will next conduct an exercise on IPv6.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020



- You think you have no IPv6 traffic on your network? Think twice...
- Modern operating systems support it and generate IPv6 traffic unless explicitly disabled
- Needs to be examined and assessed because attackers definitely will try to find it and use its weaknesses to their advantage, including exfiltration

This page intentionally left blank.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

## Red Team Scenario – IPv6 and Evil Foca

Using a tool called Evil Foca, the replicants, once on the internal network, will try to MITM on IPv6 networks using Neighbor Advertisement Spoofing, SLAAC Attack and fake DHCPv6.



### Red Team Scenario – IPv6 and Evil Foca

[1] <https://www.slideshare.net/elevenpaths/evil-focawp>

[2] <https://www.elevenpaths.com/labstools/evil-foca/index.html>

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

## Threats on 530.2 – IPv6



- Neighbor Advertisement Spoofing
- SLAAC attack
- Fake DHCPv6
- Other vulnerabilities in IPv6
- Blindness of IPv6 traffic of IDS/IPS
- Tunneling

This page intentionally left blank.

## IPv4: Nearly Exhausted

- IPv4 allows 4.2+ billion addresses, which probably seemed like a lot in 1982 when TCP was initially released
- All of the major publicly-routable IPv4 netblocks have been issued
  - If you want IPv4 address blocks in most areas of the world: you must pay for them, or go on a waiting list
  - Recent IPv4 prices (per IP address in \$USD)<sup>1</sup>:

Block Size*	/24	/23	/22	/21	/20	/19	/18	/17	/16
Price/IP (USD)	24.00	21.00	18.00	16.00	13.00	13.00	13.00	13.00	14.50

- IPv6 is growing very quickly (as we will show shortly) as a result

### IPv4: Nearly Exhausted

Per the price chart shown above, a class B (/16) is worth nearly 1 million U.S. dollars (\$950,272).

All of the major regional registrars except for AFRINIC began running out of IPv4 allocations beginning in 2011:

*On 31st January 2011, the IANA allocated their two remaining top-level address blocks to APNIC. APNIC run out of IPv4 public addresses some months later. RIPE followed the next year, as well as LACNIC in 2014 and ARIN in 2015. Today, the only RIR with IPv4 public addresses available is AFRINIC, but it won't last long, only until 2018.<sup>2</sup>*

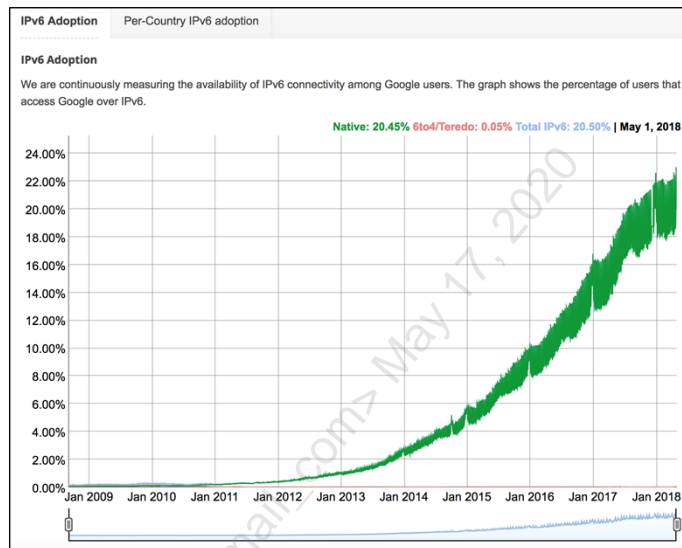
This means IPv6 is coming: now. There are workarounds to slow the need for IPv6 adoption, including CIDR addresses, NAT, and carrier-grade NAT (discussed next). In reality: CIDR and NAT have historically helped stem the tide but can no longer keep up with the growing demand for Internet-connected devices.

[1] [https://ipv4marketgroup.com/broker-services/buy/#id\\_table\\_prices](https://ipv4marketgroup.com/broker-services/buy/#id_table_prices)

[2] <https://blogs.igalia.com/dpino/2017/05/25/ipv4-exhaustion/>

## IPv6: Growing Fast

- IPv6 traffic continues to grow quickly, now making up over 20% of internet backbone traffic
  - The graph on the right is from Google's IPv6 statistics page.
- Many companies are ignoring IPv6 (while currently using it)
  - This is a mistake



### IPv6: Growing Fast

**You are probably using IPv6 now and may not realize it. Turn your Wi-Fi off on your cell phone, and Google "what is my IP address?" You are likely to be using an IPv6 address (though this is very carrier and location dependent).**

Microsoft makes heavy use of IPv6, including when organizations have not configured any IPv6 infrastructure:

*From Microsoft's perspective, IPv6 is a mandatory part of the Windows operating system, and it is enabled and included in standard Windows service and application testing during the operating system development process. Because Windows was designed specifically with IPv6 present, Microsoft does not perform any testing to determine the effects of disabling IPv6. If IPv6 is disabled on Windows Vista, Windows Server 2008, or later versions, some components will not function. Moreover, applications that you might not think are using IPv6—such as Remote Assistance, HomeGroup, DirectAccess, and Windows Mail—could be.*

*Therefore, Microsoft recommends that you leave IPv6 enabled, even if you do not have an IPv6-enabled network, either native or tunneled. By leaving IPv6 enabled, you do not disable IPv6-only applications and services (for example, HomeGroup in Windows 7 and DirectAccess in Windows 7 and Windows Server 2008 R2 are IPv6-only) and your hosts can take advantage of IPv6-enhanced connectivity.<sup>2</sup>*

[1] <https://www.google.com/intl/en/ipv6/statistics.html>

[2] <https://blogs.technet.microsoft.com/rmilne/2014/10/29/disabling-ipv6-and-exchange-going-all-the-way/>



## Dual-Stack Systems and Happy Eyeballs

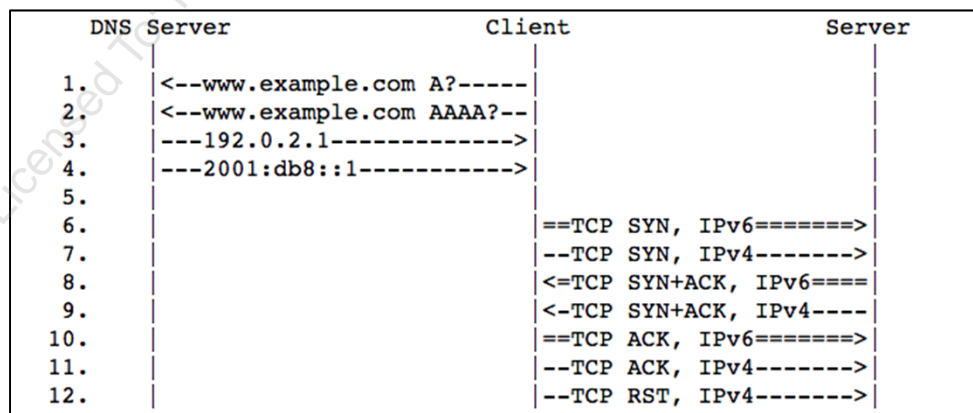
- IPv6 is usually deployed "dual-stack," meaning systems use both IPv4 and IPv6 addresses
- RFC 6555 describes the process of deciding which address to use via the Happy Eyeballs (HE) algorithm (aka fast fallback):
  - "The proposed approach is simple – if the client system is dual-stack capable, then fire off connection attempts in both IPv4 and IPv6 in parallel, and use (and remember) whichever protocol completes the connection sequence first. The user benefits because there is no wait time and the decision favours speed – whichever protocol performs the connection fastest for that particular end site is the protocol that is used to carry the payload."<sup>1</sup>
- In practice: many dual-stack systems will try to resolve both the A (IPv4) and AAAA (IPv6) DNS records of a name
  - And then immediately attempt to use the IPv6 address if the AAAA record resolves

### Dual-Stack Systems and Happy Eyeballs

Why Happy Eyeballs? When IPv6 was first deployed on dual-stack systems, the system would attempt to use the IPv6 address first and fall back to the IPv4 address if IPv6 failed. The problem with that scheme: there were noticeable delays, leading to user frustration (and unhappy eyeballs):

*When a server's IPv4 path and protocol are working, but the server's IPv6 path and protocol are not working, a dual-stack client application experiences significant connection delay compared to an IPv4-only client. This is undesirable because it causes the dual-stack client to have a worse user experience.<sup>2</sup>*

This diagram from RFC 6555<sup>2</sup> shows the process:



[1] <https://blog.apnic.net/2016/07/25/happy-eyeballs-promoting-healthy-ipv4-ipv6-coexistence/>

[2] <https://tools.ietf.org/html/rfc6555>

## IPv4 and IPv6

### IPv4 addresses are 32 bits long

- 4.2+ billion possible addresses

### IPv6 addresses are 128 bits long

- 340 undecillion addresses
- 340, followed by 36 zeroes

### IPv6 offers massively larger address space

- Also, cleaner routing
- Flexible embedded protocol support
- Stateless autoconfiguration

#### IPv4 and IPv6

IPv6 was born out of a need for more IP addresses. 4.2+ billion IPv4 addresses probably sounded like a lot in the late 1970s when TCP/IP was initially designed, but the number of devices on the Internet now exceeds 4.2 billion.

IPv6 addresses are 128 bits long, as opposed to 32-bit IPv4 addresses. This translates into a massively larger address space: 340 undecillion addresses.

In addition to more addresses, the IPv6 protocol is cleaner, with a simpler (though larger) header compared to IPv4. Routing is simpler and additional features like stateless autoconfiguration are supported, which we will discuss shortly.

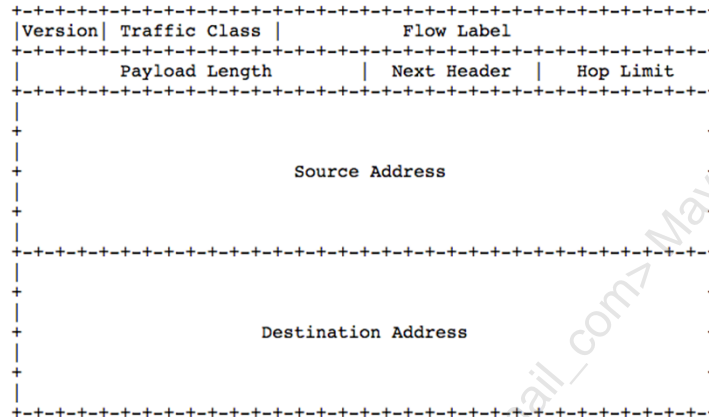
Wondering what happened to IPv5? It was used for, "Experimental Internet Stream Protocol: Version 2 (RFC 1190) and Internet Stream Protocol Version 2+ (RFC 1819) ST2 and ST2+ respectively. 2.0 ST2 and ST2+ Both ST2 and ST2+ have been given the Internet Protocol Version 5 (IPv5) designation."<sup>1</sup>

RFCs (Request for Comments) 1190 and 1819 do not mention IPv5 in relation to ST2 or ST2+, but RFC 1946 does. Those protocols never saw widespread adoption,

[1] <https://tools.ietf.org/html/rfc1946>

## IPv6 Header

- The IPv6 is larger (and simpler) than the IPv4 header (shown in the notes)

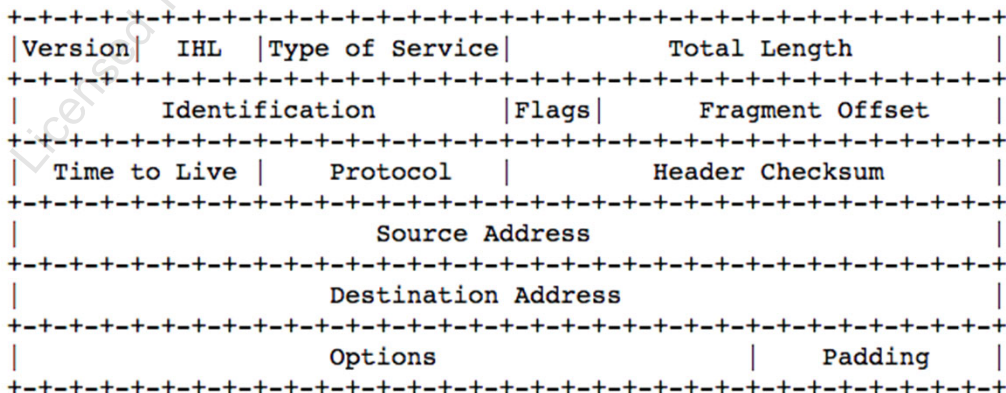


### IPv6 Header

The IPv4 header shown below is 20 bytes long (each row represents 8 bytes), not counting options (which are optional). The IPv6 header shown above is 40 bytes long and has fewer fields. The larger header length is due to the length of IPv6 addresses, which are 128 bits long (as opposed to IPv4's 32-bit addresses).

IPv6 omits the checksum, assuming that damaged or altered packets will be detected at another layer (such as layer 4 for UDP or TCP, which calculate checksums). This offloads the checksum verification from the router to the receiving system, which simplifies (and lowers) processing performed by a router.

IPv4 header:



## IPv6 Header Fields

- Version (4 bits): 6
  - IPv4 uses 4
- Traffic Class (8 bits): sets the priority of the packet
- Flow Label (20 bits): used to associate multiple packets in the same stream, such as a video transfer
- Payload Length (16 bits): length (in bytes) of the payload.
- Next Header (8 bits): describes the IPv6 extension header (if used), or the layer 4 header (TCP, UDP, etc.)
- Hop Limit (8 bits): renamed from IPv4's "Time to Live" field
- Source and Destination IP addresses (128 bits each)

### IPv6 Header Fields

The Flow Label allows routers to recognize associated packets (such as a video stream), and allow them to route subsequent packets with less resources: "The idea is that packets belonging to the same stream, session, or flow share a common flow label value, making the session easily recognizable without having to look "deep" into the packet. Recognizing a stream or session is often useful in Quality of Service (QoS) mechanisms.<sup>1</sup>

IPv6 renamed "Time to Live" to "Hop Limit," to more accurately describe its use. RFC 791 originally describes Time to Live (TTL) as using seconds:

*The time is measured in units of seconds, but since every module that processes a datagram must decrease the TTL by at least one even if it process the datagram in less than a second, the TTL must be thought of only as an upper bound on the time a datagram may exist. The intention is to cause undeliverable datagrams to be discarded and to bound the maximum datagram lifetime.<sup>2</sup>*

Implementers simply used Time to Live as a hop count, with each router decrementing the TTL by one, and dropping the packet if the TTL reached zero. IPv6 renamed the field "Hop Limit" for this reason.

[1] <https://www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-13/ipv6-internals.html>

[2] <https://tools.ietf.org/html/rfc791>

## IPv6 Extension Headers I

- The first IPv6 header is always 40 bytes long
- IPv6 extension headers may be used for options such as routing, fragmentation, authentication, encapsulation, and others
- Multiple extension headers may be used in one packet, chained together
  - The maximum chain size is unlimited
  - This can lead to interesting attacks, including denial of service by creating very long extension header chains

### IPv6 Extension Headers I

Cisco has a great guide for IPv6 extension headers in 'IPv6 Extension Headers Review and Considerations'. They also discuss the security implications of chained extension headers:

*Security Note: There is always the possibility that IPv6 traffic with a significant number of extension headers or very large extension headers is sent into the network with the malicious intent of overrunning the HW resources of network devices. Regardless of the platform HW design, this is a possible DDoS type of attack vector. Security features protecting against it must be implemented. To protect the CPU from being overwhelmed by high rates of this type of traffic, Cisco routers implement rate limiting of packets that are diverted from the hardware to software path.*

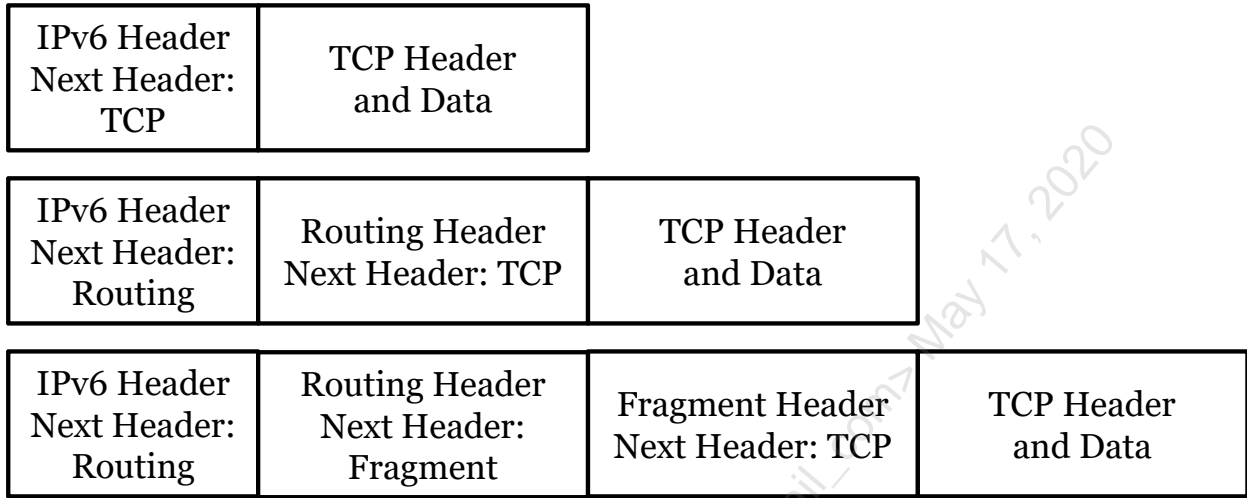
Chaining can also be used to attempt to bypass IPS (Intrusion Prevention Systems) and other controls:

*Someone could create an IPv6 packet that meets the protocol specification and has an unlimited number of extension headers linked together in a big list. A packet like this might cause a DoS of intermediary systems along the transmission path or the destination systems. The crafted packet might also pass through the network without causing any problems. Chaining lots of extension headers together is a way for attackers to avoid firewalls and Intrusion Prevention Systems (IPS). Packets that have a large chain of extension headers could be dangerous. Numerous extension headers in a single packet could spread the payload into a second fragmented packet that would not be checked by a firewall that is only looking at the initial fragment.<sup>2</sup>*

[1] [https://www.cisco.com/en/US/technologies/tk648/tk872/technologies\\_white\\_paper0900aecd8054d37d.html](https://www.cisco.com/en/US/technologies/tk648/tk872/technologies_white_paper0900aecd8054d37d.html)

[2] <http://dergipark.gov.tr/download/article-file/147978>

**IPv6 Extension Headers II (See Notes for Details)**



**IPv6 Extension Headers II (See Notes for Details)**

The images above are based on RFC 2460 (<https://tools.ietf.org/html/rfc2460>).

The first image shows a TCP packet carried via IPv6, with no extension headers.

The second shows the same packet, with one extension header (routing).

The third image shows the same packet, with both Routing and Fragment extension headers chained together.

## IPv6 Addresses

### IPv6 uses colon-separated hexadecimal values

- Repeated zeroes may be summarized as “::”

### Address below is:

- fe80:0000:0000:0000:020c:29ff:fec0:f094
- Summarized to: fe80::20c:29ff:fec0:f094

```
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0c:29:c0:f0:94
          inet addr:10.20.30.100  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::20c:29ff:fec0:f094/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:57766  errors:1  dropped:0  overruns:0  frame:0
          TX packets:33036  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0  txqueuelen:1000
          RX bytes:51302371 (51.3 MB)  TX bytes:2412928 (2.4 MB)
          Interrupt:19  Base address:0x2000
```

### IPv6 Addresses

Both IPv4 and IPv6 addresses are a series of bits (32 and 128, respectively). Both use a common format for displaying to humans. Both IPv4 and IPv6 addresses have a network portion and a host portion.

IPv4 uses a “dotted quad” format, for example, 192.168.1.7. Instead of dots, IPv6 uses colons to break up a series of numbers, such as fe80::20c:29ff:fec0:f094. Note that IPv6 uses hexadecimal values when displayed, while IPv4 uses decimals only.

IPv6 may also summarize repeating strings of zeroes as “::”. This may be done only once, to avoid ambiguity. This includes the leading zero in any series of hexadecimal numbers. For example: “0000:020c” may be summarized as “::20c”.

## IPv6 Subnet Size

- The default IPv6 subnet size is a /64
  - The entire IPv4 address space is 32 bits (~4.3 billion addresses)
- How big is a /64?
  - Take 4.3 billion, and double it... 32 times in a row
  - 8.6 billion, 17.2 billion, 34.4 billion (and double 29 times more)
  - Totaling 18,446,744,073,709,551,616 addresses (18+ quintillion addresses)

```

ericconrad ~ --bash -- 90x12
en4: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
options=10b<RXCSUM, TXCSUM, VLAN_HWTAGGING, AV>
ether 38:c9:86:1f:9a:d7
inet6 fe80::18e6:6f21:253a:a069%en4 prefixlen 64 secured scopeid 0x12
inet 10.99.99.100 netmask 0xfffff00 broadcast 10.99.99.255
inet6 2001:470:1f11:78e:4c:a13:3711:1579 prefixlen 64 autoconf secured
inet6 2001:470:1f11:78e:b174:1d8:4003:6548 prefixlen 64 autoconf temporary
inet6 fdfe:9e87:9d56:1000:4b4:7f27:9a75:2d0d prefixlen 64 autoconf secured
inet6 fdfe:9e87:9d56:1000:601a:abd:bd7e:fb7d prefixlen 64 autoconf temporary
nd6 options=201<PERFORMNUD,DAD>
media: 1000baseT <full-duplex>
status: active
  
```

### IPv6 Subnet Size

/64 is the default IPv6 subnet size, and (typically) the smallest subnet used by IPv6. Note that some applications may use smaller subnets: for example, IPv6 tunnel brokers may assign portions of their own IPv6 subnet to remote networks connected via tunnels. In this case: it is common to assign the upper half a /64 (a /65) to the remote network.

The number 18,446,744,073,709,551,616 is pronounced (in US English) as: eighteen quintillion, four hundred forty six quadrillion, seven hundred forty four trillion, seventy three billion, seven hundred nine million, five hundred fifty one thousand, six hundred sixteen.<sup>1</sup>

The screenshot above is from the author's office network, which covers the grand space of... 3 rooms. Yes, those 3 rooms contain a network that allows 18+ quintillion addresses.

It is common to have multiple address types, plus a mix of temporary and permanent ('secured') addresses. Secured addresses do not change and are not based on the adapter's MAC address. We will discuss the IPv6 address types and methods for creating IPv6 addresses shortly.

[1] <http://www.webmath.com/saynum.html>



## Types of IPv6 Addresses

IPv6 systems may use three separate address types:

- Link-local addresses
  - Used on the local subnet only, network prefix begins with "fe80"
  - All IPv6-enabled systems have this address
- Unique Local Addresses (ULA)
  - May be used on privately owned networks, network prefix begins with "fd00"
  - They are not routed publicly
  - Some organizations do not use these addresses (see notes)
- Global Unicast Addresses
  - Routed publicly
- Systems may have multiple Unique Local and Global Unicast Addresses, which we will discuss next

### Types of IPv6 Addresses

IPv6 supports three separate address types, as described above. Systems may also have multiple Unique Local Addresses and Global Unicast Addresses (including the use of privacy extension addresses), which we will discuss next.

Simply put: link-local addresses work on one subnet/LAN, unique local addresses work on a private WAN (not routed via the internet), and global unicast addresses may route via the internet.

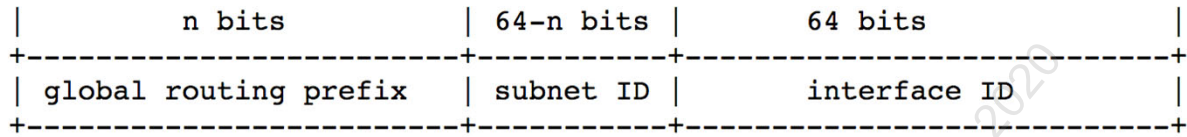
All IPv6-enabled systems have a link-local address. This happens automatically: no additional steps are required. This means every organization that has deployed Microsoft Vista (the first Microsoft client OS to support IPv6) or newer (including any recent Linux, macOS, etc.) are using IPv6 today.

Unique Local Addresses are often skipped by organizations that use IPv6: they simply use Link Local and Global Unicast Addresses. Why use Unique Local Addresses? These addresses cannot (directly) reach the internet, which can add a layer of defense in depth protection (in addition to firewalls, etc.)

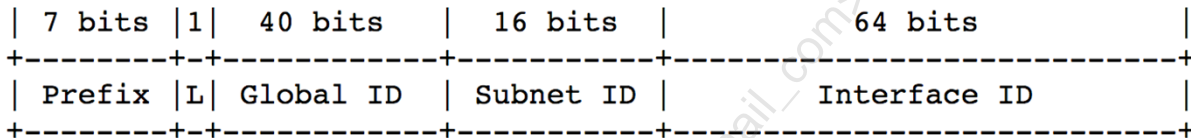
Let's assume your organization uses SMB (Server Message Block) internally only: no SMB is routed to the internet. If you have the SMB service listen to the Unique Local Address and \*not\* listen on the Global Unicast Address: SMB cannot reach the internet directly.

## IPv6 Address Format

- IPv6 Global Unicast Addresses allow 65536 subnets (16 bits), each with 18+ quintillion host addresses (64 bits):



- Unique Local Addresses include a 40-bit Global ID, which is designed to be generated randomly (as we will discuss next). They also allow 65536 subnets (16 bits), each with 18+ quintillion host addresses (64 bits):



### IPv6 Address Format

IPv6 address formats are shown above. The primary format difference between Global Unique Addresses and Unique Local Addresses (ULA) is the use of Global ID in ULAs.

RFC 4193 describes the Unique Local Address format:

- *Prefix:* FC00::/7 prefix to identify Local IPv6 unicast addresses.
- *L:* Set to 1 if the prefix is locally assigned. Set to 0 may be defined in the future.
- *Global ID:* 40-bit global identifier used to create a globally unique prefix.
- *Subnet ID:* 16-bit Subnet ID is an identifier of a subnet within the site.
- *Interface ID:* 64-bit Interface ID<sup>3</sup>

[1] <https://tools.ietf.org/html/rfc3587>

[2] <https://tools.ietf.org/html/rfc4193>

[3] Ibid.

## Determining IPv6 Network Allocations

- Global Unicast Address allocations are issued to organizations by Regional Internet Registries, such as ARIN, RIPE, AFRINIC, APNIC, and LACNIC
- Unique Local Addresses are used locally but are designed to be globally unique
  - This avoids requiring renumbering subnets if two organizations connect Unique Local Address subnets via an extranet connection
- Unique local address Global IDs are generated randomly
  - 40 bits of the address are set randomly
  - There are 1.1 trillion possible subnets
  - The odds of a collision between two organizations is quite small

### Determining IPv6 Network Allocations

Note that the Regional Internet Registries are:

- African Network Information Center (AFRINIC)
- American Registry for Internet Numbers (ARIN)
- Asia-Pacific Network Information Centre (APNIC)
- Latin America and Caribbean Network Information Centre (LACNIC)
- Réseaux IP Européens Network Coordination Centre (RIPE NCC)

RFC 4193 describes the process of generating a Unique Local Address:

*Locally assigned Global IDs MUST be generated with a pseudo-random algorithm... It is important that all sites generating Global IDs use a functionally similar algorithm to ensure there is a high probability of uniqueness.*

*The use of a pseudo-random algorithm to generate Global IDs in the locally assigned prefix gives an assurance that any network numbered using such a prefix is highly unlikely to have that address space clash with any other network that has another locally assigned prefix allocated to it. This is a particularly useful property when considering a number of scenarios including networks that merge, overlapping VPN address space, or hosts mobile between such networks.<sup>1</sup>*

[1] <https://tools.ietf.org/html/rfc4193>

## IPv6 Stateless Address Auto Configuration (SLAAC)

- IPv6 Stateless Address Auto Configuration (SLAAC) originally used the MAC address to form an IPv6 address
- The system listens for IPv6 global prefix router advertisement
  - Link-local prefix “0xfe80” is used for the link-local address
- 48-bit MAC address is split in half
  - Constant “0xfffe” inserted in the middle
- The 7<sup>th</sup> bit of MAC address is flipped for universal addresses
- Note that static or DHCPv6-assigned addresses may also be used

### IPv6 Stateless Address Auto Configuration (SLAAC)

An additional feature of IPv6 is autoconfiguration. DHCP is no longer needed (though exists as an option, called DHCPv6).

Stateless Address Auto Configuration (SLAAC) means a system can independently determine its own IPv6 address, with no additional infrastructure or servers needed. The term “stateless” means there is no requirement to maintain a server or table of addresses, as with DHCP (which is stateful).

The system listens for an IPv6 global prefix router advertisement and uses the advertised prefix as the network portion of its global address. The host portion is based on the system’s MAC address, the constant “0xfffe” inserted into the middle (for 48-bit MAC addresses). If the MAC address is globally unique (across the Internet), the 7<sup>th</sup> bit is flipped. If the MAC is not unique (for locally-set MAC addresses, such as those used for some virtual systems), the 7<sup>th</sup> bit is not flipped.

The link-local address (begins with fe80) uses the same process but does not require a router advertisement. If no routers advertise IPv6 global prefixes, the system will assign itself a link-local address only.

## Link Local IPv6 Stateless Address Auto Configuration (SLAAC) Example

- SLAAC uses the MAC address to determine the IPv6 address
  - This can result in privacy issues, as described in the notes

MAC address	00 0c 29 c0 f0 94
Add "fffe" constant	00 0c 29 ff fe c0 f0 94
Set universal bit	02 0c 29 ff fe c0 f0 94
Add prefix, use ":"	fe80:0000:0000:0000:020c:29ff:fec0:f094
Summarize 0's	fe80::20c:29ff:fec0:f094

### Link Local IPv6 Stateless Address Auto Configuration (SLAAC) Example

This diagram shows the process of assigning a link-local address. The constant "fffe" (16 bits) is embedded in the middle of the 48-bit MAC address, making it "EUI-64" format (Extended Unique Identifier). The universal bit (also called the global bit) is set if the address is globally unique on the internet (meaning it could be routed publicly using the Global Unicast Address, which we will discuss shortly). The 64-bit network prefix is added, resulting in a 128-bit value.

The system's MAC address is `00:0c:29:c0:f0:94`, which results in a link local IPv6 address of `fe80::20c:29ff:fec0:f094`.

The (arguable) benefit of this scheme: no additional infrastructure is needed to assign IPv6 address (beyond requiring the network prefix, discussed shortly). This includes not requiring DHCP. The system itself is able to assign itself a unique address. The reason this is a debatable benefit: many organizations seek more control of their IP address assignments, not less.


This method was originally used to create Global Unicast Addresses ('public' IPv6 addresses), embedding the system's MAC address, while using a public network prefix. The issue with the scheme: it creates privacy issues. Any site on the internet could track the same system via IPv6, regardless of the network. MAC addresses are designed to be globally unique, so sites could build databases of the MAC address portion of addresses created with this scheme and track them.

As a result: other schemes for creating Global Unicast Addresses now exist (and are quite common), including 'privacy extension' addresses, which we will discuss next.

## One System, Six IP Addresses

This macOS High Sierra system has six IP addresses:

- `fe80::18e6:6f21:253a:a069%en4`: IPv6 Link Local (`%en4` identifies the Ethernet adapter)
- `2001:470:1f11:78e:4c:a13:3711:1579`: IPv6 Global Unicast secured (not temporary)
- `2001:470:1f11:78e:b174:1d8:4003:6548`: IPv6 Global Unicast temporary
- `fdfe:9e87:9d56:1000:4b4:7f27:9a75:2d0d`: IPv6 Unique Local secured (not temporary)
- `fdfe:9e87:9d56:1000601a:abd:bd7e:fb7d`: IPv6 Unique Local temporary
- `10.99.99.100`: IPv4



```

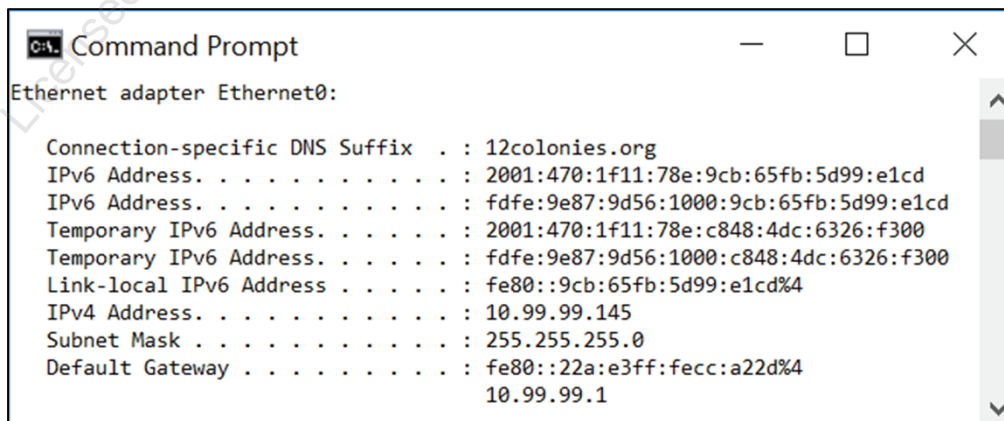
ericconrad ~ --bash -- 90x12
en4: flags=8963<UP, BROADCAST, SMART, RUNNING, PROMISC, SIMPLEX, MULTICAST> mtu 1500
options=10b<RXCSUM, TXCSUM, VLAN_HWTAGGING, AV>
ether 38:c9:86:1f:9a:d7
inet6 fe80::18e6:6f21:253a:a069%en4 prefixlen 64 secured scopeid 0x12
inet 10.99.99.100 netmask 0xfffff00 broadcast 10.99.99.255
inet6 2001:470:1f11:78e:4c:a13:3711:1579 prefixlen 64 autoconf secured
inet6 2001:470:1f11:78e:b174:1d8:4003:6548 prefixlen 64 autoconf temporary
inet6 fdfe:9e87:9d56:1000:4b4:7f27:9a75:2d0d prefixlen 64 autoconf secured
inet6 fdfe:9e87:9d56:1000:601a:abd:bd7e:fb7d prefixlen 64 autoconf temporary
nd6 options=201<PERFORMNUD, DAD>
media: 1000baseT <full-duplex>
status: active
  
```

### One System, Six IP Addresses

"Secured" is the macOS term for non-temporary IPv6 addresses. The reason we're describing the "secured" addresses as "not temporary" (and not using the term "permanent") is because these addresses can be configured via DHCPv6 (among other methods), meaning they could change (as leases expire, etc.). The temporary IPv6 addresses *will* change.

The temporary addresses are used to provide privacy and were initially designed when SLAAC was used to create the Global Unicast Address (which exposed the system's MAC address to the internet). We will discuss privacy-enhanced and temporary IPv6 addresses in more detail shortly.

Note that the number of addresses is not only a macOS feature; it is common among recent operating systems. Here is the "ipconfig" output from a Windows 10 system on the same network:



```

C:\> ipconfig

Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . : 12colonies.org
   IPv6 Address. . . . . : 2001:470:1f11:78e:9cb:65fb:5d99:e1cd
   IPv6 Address. . . . . : fdfe:9e87:9d56:1000:9cb:65fb:5d99:e1cd
   Temporary IPv6 Address. . . . . : 2001:470:1f11:78e:c848:4dc:6326:f300
   Temporary IPv6 Address. . . . . : fdfe:9e87:9d56:1000:c848:4dc:6326:f300
   Link-local IPv6 Address . . . . . : fe80::9cb:65fb:5d99:e1cd%4
   IPv4 Address. . . . . : 10.99.99.145
   Subnet Mask . . . . . : 255.255.255.0
   Default Gateway . . . . . : fe80::22a:e3ff:fecc:a22d%4
                               10.99.99.1
  
```

## IPv6 Privacy Extension Addresses and Temporary Addresses

- As noted previously: IPv6 addresses created via SLAAC expose the MAC address, which may result in privacy issues
  - As a result: IPv6 privacy extension addresses are used by most current operating systems
  - The privacy extension address is not based on the MAC address (discussed next)
  - Most systems use privacy extension addresses for the unique local and global unicast addresses, and continue to embed the MAC address in the link-local address (used on the local subnet only)
- Most systems also create two addresses for each unique local and global unicast address
  - The temporary address is normally preferred for all communication
- This combination adds an additional layer of privacy: these addresses are not tied to the MAC (privacy extensions), \*and\* they change routinely (temporary addresses)
  - This is now the default behavior on current Windows and macOS operating systems
  - Older Linux Ubuntu distros (such as Ubuntu 14.04) do not use IPv6 privacy extension addresses or temporary addresses by default, see notes for details

### IPv6 Privacy Extension Addresses and Temporary Addresses

Some Linux distributions do not support IPv6 privacy extension addresses by default, including older versions of Ubuntu Linux (note that Ubuntu 16.04 and newer support IPv6 privacy extension addresses by default).

To enable privacy extensions and temporary IPv6 addresses on older versions of Ubuntu, perform the following steps<sup>1</sup>:

```
$ sudo sysctl net.ipv6.conf.eth0.use_tempaddr=2
```

Then restart networking:

```
$ sudo /etc/init.d/networking restart
```

Or (depending on the version of Ubuntu):

```
$ sudo ifdown eth0 && sudo ifup eth0
```

Note that your interface name may be different.

[1] <https://docs.menandmice.com/display/MM/enable+IPv6+privacy+extension+on+Ubuntu+Linux>

## Ubuntu 14.04 System: Before and After Privacy Extensions

```

Terminal - student@5ec-511-Linux:~$ ip a
2: eth0: <BROADCAST,MULTICAST,PROMISC,UP,LOWER UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:52:4b:a6 brd ff:ff:ff:ff:ff:ff
    inet 10.99.99.109/24 brd 10.99.99.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fdfe:9e87:9d56:1000:20c:29ff:fe52:4ba6/64 scope global dynamic
        valid_lft 86399sec preferred_lft 14399sec
    inet6 2001:470:1f11:78e:20c:29ff:fe52:4ba6/64 scope global tentative dynamic
        valid_lft 86399sec preferred_lft 14399sec
    inet6 fe80::20c:29ff:fe52:4ba6/64 scope link
        valid_lft forever preferred_lft forever

Terminal - student@5ec-511-Linux:~$ ip a
2: eth0: <BROADCAST,MULTICAST,PROMISC,UP,LOWER UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:52:4b:a6 brd ff:ff:ff:ff:ff:ff
    inet 10.99.99.109/24 brd 10.99.99.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fdfe:9e87:9d56:1000:895:3e41:2e73:988c/64 scope global temporary dynamic
        valid_lft 86344sec preferred_lft 14344sec
    inet6 fdfe:9e87:9d56:1000:20c:29ff:fe52:4ba6/64 scope global dynamic
        valid_lft 86344sec preferred_lft 14344sec
    inet6 2001:470:1f11:78e:895:3e41:2e73:988c/64 scope global temporary dynamic
        valid_lft 86344sec preferred_lft 14344sec
    inet6 2001:470:1f11:78e:20c:29ff:fe52:4ba6/64 scope global dynamic
        valid_lft 86344sec preferred_lft 14344sec
    inet6 fe80::20c:29ff:fe52:4ba6/64 scope link
        valid_lft forever preferred_lft forever
  
```

### Ubuntu 14.04 System: Before and After Privacy Extensions

The screenshots show the result of the "ip a" command on an Ubuntu 14.04 system. Note the IPv6 addresses in the "before" (upper) screenshot: they are based on the MAC address, and there are no temporary addresses.

The system's MAC address is: **00:0c:29:52:4b:a6**.

The Global Unicast Address is: **2001:470:1f11:78e:20c:29ff:fe52:4ba6**, which is based directly off the MAC address. As described previously: the constant "fffe" is inserted in the middle, and the Globally Unique bit is flipped, making the host portion "20c:29ff:fe52:4ba6". The network prefix is then added, creating the full address. This exposes the MAC address publicly, leading to privacy concerns.

The same process is followed for the Unique Local Address (using a different network prefix, resulting the address: **fdfe:9e87:9d56:1000:20c:29ff:fe52:4ba6**).

We then made a single change ("sudo sysctl net.ipv6.conf.eth0.use\_tempaddr=2", described on the previous page), which enables both privacy extensions and temporary IPv6 addresses on Ubuntu 14.04 systems. After restarting networking, the system had the IPv6 addresses shown in the bottom screenshot.



## How Are Privacy-Enhanced IPv6 Addresses Generated?

- The host portion of IPv6 privacy-enhanced addresses are generated randomly by each host system
- This raises a (very small) risk of duplicate addresses
  - Remember: each subnet allows 18+ quintillion addresses
  - The odds of a collision are extremely small
- IPv6 hosts using privacy extension addresses also perform Duplicate Address Detection (DAD), per RFC 4941:
  - *The node MUST perform duplicate address detection (DAD) on the generated temporary address. If DAD indicates the address is already in use, the node MUST generate a new randomized interface identifier<sup>1</sup>*
- Note that privacy-enhanced IPv6 addresses are used when systems use SLAAC to generate an IP address
  - They are not typically used when the IPv6 address is assigned via DHCPv6 or other methods

### How Are Privacy-Enhanced IPv6 Addresses Generated?

Most non-mathematicians struggle with the sheer size of total IPv6 address space, as well as the size of a /64 subnet, which is (typically) the smallest subnet used by IPv6. As noted previously: a /64 subnet allows 18,446,744,073,709,551,616 (over 18 quintillion<sup>2</sup> see below) hosts.

Assuming the hosts are chosen truly randomly (and true randomness is difficult on a computer): the sun will likely supernova before any two hosts on any network in the world assign themselves the same random number.

Despite this math, RFC 4941 (Privacy Extensions for Stateless Address Autoconfiguration in IPv6) requires the use of Duplicate Address Detection (DAD) and will generate a new random host address if the current one is in use on the network.

Note that 'quintillion' describes the United States, Canadian, and Modern British English word for the number. Fun fact: traditional British and European usage would call that number '18 trillion.'<sup>2</sup>

As noted above: privacy-enhanced IPv6 addresses are used when systems use SLAAC to generate an IP address. We will connect to an IPv6 tunnel broker (via IPv4) in the upcoming IPv6 lab, and the Security530 Linux VM will not use a privacy-enhanced IPv6 address: it will use the address (which is not based on the local MAC address) assigned by the tunnel broker.

[1] <https://tools.ietf.org/html/rfc4941.html>

[2] <http://eyeful-tower.com/muse/billion.htm>

## IPv6 Temporary Address Lifetime

- Temporary IPv6 addresses have a preferred lifetime and a valid lifetime
  - Preferred lifetime: once expired, the system will use a new temporary address for new connections
  - Valid lifetime: the system will continue to accept connections from existing connections on this address for this period of time
- You may view the lifetimes with these commands:
  - Windows: `netsh interface ipv6 show address`
  - Linux: `ip a`
  - macOS: `ifconfig -L`

```

Command Prompt
C:\Users\student>netsh interface ipv6 show address

Interface 1: Loopback Pseudo-Interface 1

Addr Type DAD State Valid Life Pref. Life Address
-----
Other Preferred Infinite Infinite ::1

Interface 4: Ethernet0

Addr Type DAD State Valid Life Pref. Life Address
-----
Public Preferred 23h56m45s 3h56m45s 2001:470:1f11:78e:9cb:65fb:5d99:e1cd
Temporary Preferred 23h56m45s 3h56m45s 2001:470:1f11:78e:c848:4dc:6326:f300
Public Preferred 23h56m45s 3h56m45s fdfe:9e87:9d56:1000:9cb:65fb:5d99:e1cd
Temporary Preferred 23h56m45s 3h56m45s fdfe:9e87:9d56:1000:c848:4dc:6326:f300
Other Preferred Infinite Infinite fe80::9cb:65fb:5d99:e1cd%4
  
```

## IPv6 Temporary Address Lifetime

The Linux command is "ip a", where "a" is short for "addr" or "address" (both options also work).

```

[~]$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,PROMISC,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:52:4b:a6 brd ff:ff:ff:ff:ff:ff
    inet 10.99.99.109/24 brd 10.99.99.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fdfe:9e87:9d56:1000:20c:29ff:fe52:4ba6/64 scope global dynamic
        valid_lft 85898sec preferred_lft 13898sec
    inet6 2001:470:1f11:78e:20c:29ff:fe52:4ba6/64 scope global dynamic
        valid_lft 85898sec preferred_lft 13898sec
    inet6 fe80::20c:29ff:fe52:4ba6/64 scope link
        valid_lft forever preferred_lft forever
  
```

macOS added the "-L" flag to the "ifconfig" command, which means "If -L flag is supplied, address lifetime is displayed for IPv6 addresses, as a time offset string."<sup>[1]</sup>

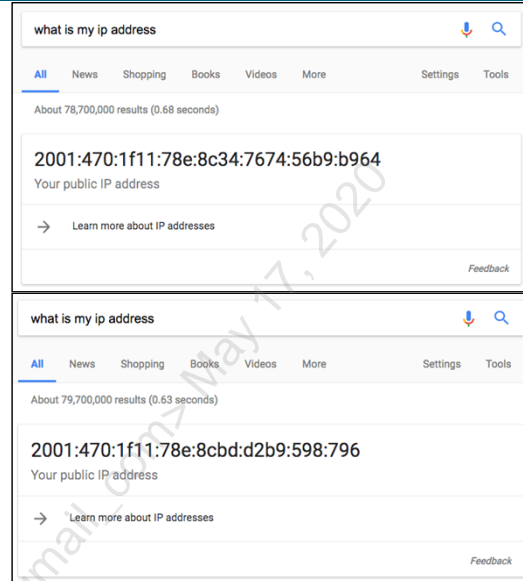
```

Orion:~ ericconrad$ ifconfig -L en4
en4: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
    options=10b<RXCSUM, TXCSUM, VLAN_HWTAGGING, AV>
    ether 38:c9:86:1f:9a:d7
    inet6 fe80::18e:6f21:253a:a069%en4 prefixlen 64 secured scopeid 0x12
    inet 10.99.99.100 netmask 0xfffff00 broadcast 10.99.99.255
    inet6 2001:470:1f11:78e:4c:a13:3711:1579 prefixlen 64 autoconf secured pltime 14223 vltime 86223
    inet6 2001:470:1f11:78e:b174:1d8:4003:6548 prefixlen 64 autoconf temporary pltime 14223 vltime 86223
    inet6 fdfe:9e87:9d56:1000:4b4:7f27:9a75:2d0d prefixlen 64 autoconf secured pltime 14223 vltime 86223
    inet6 fdfe:9e87:9d56:1000:601a:abd:bd7e:fb7d prefixlen 64 autoconf temporary pltime 14223 vltime 86223
    nd6 options=201<PERFORMNUD,DAD>
    media: 1000baseT <full-duplex>
    status: active
Orion:~ ericconrad$
  
```

[1] <https://ss64.com/osx/ifconfig.html>

## Effect of Temporary IPv6 Addresses

- In Windows, macOS, and most Linux distributions, the default IPv6 preferred lifetime is 1 day, and the valid default lifetime is 7 days
  - This value can be overridden by IPv6 route advertisements (see notes)
- The screenshots on the right show the same system, 24 hours apart
- IPv4-style thinking often fails with IPv6
  - Manually filtering or blacklisting temporary IPv6 addresses is not effective



### Effect of Temporary IPv6 Addresses

You may check the local default lifetimes with the following commands:

- Windows: `netsh interface ipv6 show privacy`
- Linux: `sysctl -a | grep net.ipv6.conf.all.temp`
- macOS: `sysctl -a | grep net.inet6.ip6.temp`

Note that IPv6 routing daemons such as the Router Advertisement Daemon (radvd) can override the local systems default preferred and valid lifetimes (radvd defaults to 1 day valid lifetime, 4 hours preferred lifetime, which is considerably shorter than the default OS settings). Linux/Unix IPv6 clients with the 'radvdump' command installed can check those settings with this command: **radvdump**

```
ericconrad — root@peaks: /etc — ssh econrad@10.99.99.1 — 77x29
root@peaks:/etc# radvdump
#
# radvd configuration generated by radvdump 2.11
# based on Router Advertisement from fe80::22a:e3ff:fecc:a22d
# received by interface enp2s0
#
interface enp2s0
{
    AdvSendAdvert on;
    # Note: {Min,Max}RtrAdvInterval cannot be obtained with radvdump
    AdvManagedFlag off;
    AdvOtherConfigFlag off;
    AdvReachableTime 0;
    AdvRetransTimer 0;
    AdvCurHopLimit 64;
    AdvDefaultLifetime 1800;
    AdvHomeAgentFlag off;
    AdvDefaultPreference medium;
    AdvSourceLLAddress on;

    prefix 2001:470:1f11:78e::/64
    {
        AdvValidLifetime 86400;
        AdvPreferredLifetime 14400;
        AdvOnLink on;
        AdvAutonomous on;
        AdvRouterAddr off;
    }; # End of prefix definition
}
```

## There's No Place Like ::1

- **::1** is the equivalent of the IPv4 address 127.0.0.1
- **fc00::/7** is reserved for unique local addresses
  - Equivalent to IPv4 RFC1918 addresses (such as 192.168.0.0/16, 10.0.0.0/8, etc.)
  - Includes **fc00::/8** and **fd00::/8**
  - While reserved, usage of **fc00::/7** is not yet defined
  - Sites use **fd00::/7** to assign unique local addresses

### There's No Place Like ::1

The IPv6 address **::1** is the equivalent of the IPv4 address 127.0.0.1. **::1** in binary is 127 zeroes followed by a one. As discussed previously, repeating strings of zeroes may be summarized as "**::**".

The address including the netmask is **::1/128**. Much like IPv4's /32 netmask (indicating one IP address), IPv6 uses /128 as the netmask for one IP address.

## IPv6 Multicast Addresses

- IPv6 does not support broadcast addresses and uses multicast addresses to perform a similar function to IPv4's broadcast addresses
  - Broadcast addresses are used for one -> all communication
  - Multicast addresses are used for one -> many communication
- IPv6 uses the **ff00::/8** network prefix for multicast addresses
- Two important IPv6 multicast addresses (more listed in the notes):
  - **ff02::1** - All local nodes
  - **ff02::2** - All local routers

### IPv6 Multicast Addresses

Multicast addresses become critical due to the sheer size of IPv6 subnets: sequential scanning of an entire subnet is not possible (as we will discuss in the upcoming IPv6 scanning section). Multicast addresses are critical for discovering hosts that exist on a network.

Additional Multicast addresses include:

- ff02::5 OSPFIGP
- ff02::6 OSPFIGP Designated Routers
- ff02::7 ST Routers
- ff02::8 ST Hosts
- ff02::9 RIP routers
- ff02::a EIGRP routers
- ff02::c Simple Service Discovery Protocol (SSDP)
- Ff02::16 All MLDv2-capable routers
- ff02::1:2 All-dhcp-agents
- ff02::1:3 Link-local Multicast Name Resolution (LLMNR)
- ff05::1:3 All-dhcp-servers
- ff0x::fb Multicast DNS
- ff0x::101 Network Time Protocol<sup>1</sup>

[1] <https://www.iana.org/assignments/ipv6-multicast-addresses/ipv6-multicast-addresses.xml>

## Types of IPv6 Multicast Addresses

- IPv6 Multicast addresses operate at different scopes:
  - `ff01::` Interface-Local (loopback)
  - `ff02::` Link-Local (same LAN)
  - `ff05::` Site-Local (one location)
  - `ff08::` Organization-Local (one organization)
  - `ff0e::` Global scope
- The Multicast scope has consequences for IPv6 scanning (discussed shortly)
  - The most commonly-used multicast addresses are `ff02::1` (all local nodes) and `ff02::2` (all local routers), which are Link-Local in scope
  - This limits their scope (and usefulness) for scanning purposes

### Types of IPv6 Multicast Addresses

What happens if you ping `ff0e::1` (global internet, all local nodes)? Sadly, this did not result in any responses (sent from a Linux cloud server):

```

ericconrad — root@eic: ~ — ssh -p20100 root@eic.me — 80x5
root@eic:~# ping6 -I eth0 ff0e::1
PING ff0e::1(ff0e::1) from 2604:a880:0:1010::5db:4001 eth0: 56 data bytes

```

This is definitely for the best since ICMPv6 can be forged. Imagine the DoS possibilities!

The global scope multicast address `ff0e::` is used by OLSR (Optimized Link State Routing Protocol): "The default multicast-address used is `ff05::15` for interfaces using a site-local address and `ff0e::1` for interfaces using a global address, if nothing else is specified by the user."<sup>1</sup>

OLSR is described here:

*olsrd and olsrd2 are both Link State Routing Protocol implementations optimized for Mobile ad hoc networks on embedded devices like a commercial of the shelf routers, smartphones or normal computers. Sometimes these networks are called "mesh networks". olsrd and olsrd<sup>2</sup> are the routing daemons which make up the mesh.<sup>2</sup>*

[2] [http://www.olsr.org/docs/report\\_html/node87.html](http://www.olsr.org/docs/report_html/node87.html)

[1] [http://www.olsr.org/mediawiki/index.php/Main\\_Page](http://www.olsr.org/mediawiki/index.php/Main_Page)

## Assigning IPv6 Addresses

- There are a number of methods for assigning IPv6 addresses:
  - Static Assignment
  - Stateless Address Auto Configuration (SLAAC)
    - Assigns the IPv6 host address statelessly, and assigns the network prefix for Global Unicast and Unique Local addresses from a Router Advertisement (RA) daemon (if available)
    - As noted previously, modern systems use IPv6 Privacy Extension addresses to create the Unique Local and Global Unicast addresses
  - DHCPv6
    - Stateful DHCPv6: similar to DHCP (v4), assigns all options (except the default gateway)
    - Stateless DHCPv6: RA daemon assigns an address (and gateway), DHCPv6 assigns other options (such as DNS settings, etc.)
  - Unfortunately: DHCPv6 cannot assign the default gateway
    - This means DHCPv6 must be used in conjunction with a router advertisement daemon

### Assigning IPv6 Addresses

There are a number of methods for assigning IPv6 addresses, described above.

Most organizations that leverage IPv6 use a combination of a Router Advertisement Daemon such as radvd, plus a DHCPv6 server. DHCPv6 cannot assign the default gateway.

The Router Advertisement daemon cannot assign settings such as DNS to some operating systems. radvd supports a "RDNSS" (Recursive DNS Server) setting but, "This feature is not very widely implemented" in clients.<sup>1</sup>

RFC 6106 (Router Advertisement Options for DNS Configuration) describes RDNSS<sup>2</sup> It is not supported by most Windows operating systems. However, Windows 10 Fall Creators Update does (finally) support RDNSS<sup>3</sup>.

The Linux IPv6 Router Advertisement Daemon (radvd) is available at: <http://www.litech.org/radvd/>

It is also available on BSD systems.

[1] <https://github.com/reubenhwk/radvd/blob/master/radvd.conf.example>

[2] <https://tools.ietf.org/html/rfc6106>

[3] <https://blogs.technet.microsoft.com/networking/2017/07/13/core-network-stack-features-in-the-creators-update-for-windows-10/>

## Course Roadmap

- Day 1: Defensible Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- Day 5: Zero Trust Architecture
- Day 6: Capstone: Hands-On Defend the Flag Challenge

### NETWORK SECURITY ARCHITECTURE

1. Layer 3 Attacks and Mitigation
2. Switch and Router Benchmarks
3. EXERCISE: Auditing Router Security
4. Securing SNMP
5. Securing NTP
6. Bogon Filtering
7. Blackholes and Darknets
8. EXERCISE: Router SNMP Security
9. IPv6
- 10. IPv6 Misconceptions**
11. Securing IPv6
12. EXERCISE: IPv6
13. Layer 3/4 Stateful Firewall
14. Web Proxy
15. SMTP Proxy
16. EXERCISE: Proxy Power
17. 530.2 Summary

### Course Roadmap

We will next discuss IPv6 misconceptions.



## IPv6 and NAT

- There is currently no standard IPv6 NAT (Network Address Translation) solution
  - There are non-standard IPv6 NAT solutions such as NAT66, see notes for details
  - Cisco, Juniper, and others offer IPv6 NAT options
- IPv6-enabled systems with a Global Unicast Address (meaning a 'public' address, discussed shortly) generally use their untranslated address to reach the internet
- NAT can perform a number of functions, which we will discuss next
- Note that many-to-one NAT is sometimes called NAPT (Network Address and Port Translation)
  - We will use the term "NAT"

### IPv6 and NAT

There is currently no released RFC that describes NAT for IPv6 (though there are a number of proposed, informational and experimental RFCs). Note there are also some non-standard IPv6 NAT implementations (such as NAT66), but none are standardized, and device support is spotty.

Enterprise-grade equipment, including higher-end Cisco and Juniper devices, support IPV6 NAT. Lower-end equipment, including lots of SOHO (Small Office/Home Office) devices, do not.

Some resist deploying IPv6 due to a lack of standard NAT options. We have quadrillions of IPv6 addresses, so conservation is not needed. There is a decades-old debate in the industry: is NAT a security feature? This has led to fiercely passionate debates, with folks lined up on both sides of the argument, flamethrowers in hand. We will discuss these issues next.

## NAT Functions

- NAT is used for the following functions:
  - Conserve public IP addresses (no longer needed with IPv6)
  - Avoid IP renumbering, for example: when an organization changes ISPs
  - Enable simpler multihoming (see notes for details)
    - Multihoming is the use of multiple internet gateways
  - Allow IP address uniformity (aka homogeneity)
    - Multiple remote sites can use 192.168.1.1 for the router, 192.168.1.2 as a DNS server, etc.
  - (Arguably) provide a layer of security by hiding the host IP addresses, network addresses and topology details from untrusted networks
    - This is a hotly debated subject, see notes for details
    - Note that most modern systems use IPv6 temporary addresses (discussed previously), which hide the host's non-temporary IPv6 address

### NAT Functions

Hiding the source IP address on untrusted connections does (arguably) add a layer of security. Please note: when the course authors 'arguably': we do not want to argue. There are merits to both sides of the argument, and the industry has spent too much time rehashing this point.

What is less arguable: NAT can also break things. Some protocols, such as legacy audio and video conferencing protocols, carry address information in the payload and break via NAT devices.

NAT may also introduce additional attack surface. For example: NAT gateways must maintain an address translation table (adding overhead and complexity). These tables may also be DoS-ed (either accidentally or intentionally): once the NAT table is full, new connections will usually fail until the old one's time out.

If hiding client addresses is the goal: a proxy is a good choice. Most modern proxies (including Squid) now support IPv6. See: <https://wiki.squid-cache.org/Features/IPv6>

## The Internet Engineering Task Force (IETF) on IPV6 NAT

The Internet Engineering Task Force (IETF) provide their thoughts on IPv6 NAT in RFC 5902:

- *While we do not consider IPv6 NATs to be desirable, we understand that some deployment of them is likely unless workable solutions to avoiding renumbering, facilitating multihoming without adversely impacting routing scalability, and homogeneity are generally recognized as useful and appropriate.<sup>1</sup>*

### The Internet Engineering Task Force (IETF) on IPV6 NAT

RFC 5902 contains an excellent summary of IPv6 issues and takes a more balanced approach than other sources (which often come down harshly against IPv6 NAT). Multihoming is a key issue, as they discuss:

*Unfortunately, no solution except NAT has been deployed today that can insulate the global routing system from the growing number of multihomed sites, where a multihomed site simply assigns multiple IPv4 addresses (one from each of its service providers) to its exit router, which is an IPv4 NAT box. Using address translation to facilitate multihoming support has one unique advantage: there is no impact on the routing system scalability, as the NAT box simply takes one address from each service provider, and the multihomed site does not inject its own routes into the system. Intuitively, it also seems straightforward to roll the same solution into multihoming support in the IPv6 deployment. However, one should keep in mind that this approach brings all the drawbacks of putting a site behind a NAT box, including the loss of reachability to the servers behind the NAT box.<sup>2</sup>*

The IETF also mentions that network homogeneity can be addressed via link-local addresses: 'In IPv6, link-local addresses can be used to ensure that all home gateways have the same address, and to provide homogenous addresses to any other devices supported by the service provider.<sup>3</sup>

[1] <https://tools.ietf.org/html/rfc5902>

[2] *ibid.*

[3] *ibid.*

## IPv6 and IPsec

- Some organizations engage in wishful thinking in regard to IPv6
- IPv6 originally required IPsec (IP Security) support
  - Some IPv4 devices support IPsec, but it was never required by the protocol
- As of RFC 6424 (December 2011): IPsec support is no longer required by IPv6
  - 'MUST comply' became 'SHOULD be supported' (see notes for details)
- In reality: most IPv6 devices support IPsec, but it must be configured separately
  - It does not happen automatically

### IPv6 and IPsec

RFC 4301 (Security Architecture for the Internet Protocol, December 2005) stated: "All IPv6 implementations MUST comply with all requirements of this document."<sup>1</sup>

This was later changed in RFC 6434 (IPv6 Node Requirements, December 2011) to "Security Architecture for the Internet Protocol" [RFC4301] SHOULD be supported by all IPv6 nodes.<sup>2</sup>

The Internet Society has a great series of articles on IPv6 myths, here is a quote from 'IPv6 Security Myth #2 – IPv6 Has Security Designed In' by Chris Grundemann:

*The fact that IPv6 requires IPsec does mean that it's available for use on all IPv6 capable devices, which is a step up over IPv4. It does not, however, guarantee the use of IPsec, which is what actually provides security. The responsibility remains with the application developer, the systems administrator, and the end user to actively apply IPsec for authentication and encryption. You must actively use IPsec for it to provide any security whatsoever.<sup>3</sup>*

[1] <https://tools.ietf.org/html/rfc4301>

[2] <https://tools.ietf.org/html/rfc6434#section-11>

[3] <https://www.internetsociety.org/blog/2015/01/ipv6-security-myth-2-ipv6-has-security-designed-in/>

## Course Roadmap

- Day 1: Defensible Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- Day 5: Zero Trust Architecture
- Day 6: Capstone: Hands-On Defend the Flag Challenge

### NETWORK SECURITY ARCHITECTURE

1. Layer 3 Attacks and Mitigation
2. Switch and Router Benchmarks
3. EXERCISE: Auditing Router Security
4. Securing SNMP
5. Securing NTP
6. Bogon Filtering
7. Blackholes and Darknets
8. EXERCISE: Router SNMP Security
9. IPv6
10. IPv6 Misconceptions
11. Securing IPv6
12. EXERCISE: IPv6
13. Layer 3/4 Stateful Firewall
14. Web Proxy
15. SMTP Proxy
16. EXERCISE: Proxy Power
17. 530.2 Summary

### Course Roadmap

We will next discuss securing IPv6.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

## Securing IPv6

- NIST released Special Publication 800-119, "Guidelines for the Secure Deployment of IPv6" in 2010
- They outline the following risks:
  - *The attacker community's use of IPv6*
  - *Unauthorized deployment of IPv6 on existing IPv4 production networks*
  - *Vulnerabilities present in IPv6, including –day zero vulnerabilities that are inherent in any new or revised system*
  - *Complexity added by dual IPv4/IPv6 operations*
  - *Immaturity of IPv6 security products and processes*
  - *Possible lack of vendor support<sup>1</sup>*

### Securing IPv6

NIST SP 800-119 contains an excellent summary of the issues faced the United States federal agencies (and any large organization) in securing IPv6:

*Federal agencies will most likely face security challenges throughout the deployment process, including:*

- *An attacker community that most likely has more experience and comfort with IPv6 than an organization in the early stages of deployment*
- *Difficulty in detecting unknown or unauthorized IPv6 assets on existing IPv4 production networks*
- *Added complexity while operating IPv4 and IPv6 in parallel*
- *Lack of IPv6 maturity in security products when compared to IPv4 capabilities*
- *Proliferation of transition-driven IPv6 (or IPv4) tunnels, which complicate defenses at network boundaries even if properly authorized, and can completely circumvent those defenses if unauthorized (e.g. host-based tunnels initiated by end users)<sup>2</sup>*

[1] <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-119.pdf>

[2] Ibid.

## IPv6 Security Issues

### *Sunlight is the Best of Disinfectants* – Louis Brandeis

- Every organization with Windows Vista or newer is using IPv6 right now
  - Most are ignoring it
  - Malware festers in the darkness
- Some firewalls cannot process IPv6
- Scanning IPv6 is challenging (and quite different than scanning IPv4)
- IPv6 offers robust tunneling options that can evade filtering and detection
- Black hats with internal access can advertise IPv6 routes to internal systems

Let's discuss these issues in detail

### IPv6 Security Issues

Linux has supported IPv6 since kernel version 2.18, released in 1996.<sup>1</sup> The IPv6 code matured, and was available in distributions such as Ubuntu 6.10 in 2006. IPv6 has been supported on Apple Macintosh since OS X 10.2 Jaguar in 2002.<sup>2</sup>

The bottom line: any modern operating system released in the past 10+ years supports IPv6 natively, and it uses IPv6 Local Addresses to communicate on the local LAN.

The full quote by Louis Brandeis (former United States Supreme Court justice) is, "Publicity is justly commended as a remedy for social and industrial diseases. Sunlight is said to be the best of disinfectants; electric light the most efficient policeman."<sup>3</sup>

He was discussing transparency in Wall Street and the government, but it directly applies to information security. We require visibility in the areas where malware can attack and spread. IPv6 presents a large blind spot for many organizations.

[1] <http://ldp.linux.no/HOWTO/Linux+IPv6-HOWTO/basic-history-ipv6-linux.html>

[2] <https://arstechnica.com/gadgets/2012/05/the-future-is-forever-the-state-of-ipv6-in-the-apple-world/>

[3] [https://archive.org/stream/otherpeoplesmone00bran/otherpeoplesmone00bran\\_djvu.txt](https://archive.org/stream/otherpeoplesmone00bran/otherpeoplesmone00bran_djvu.txt)

## IPv6 Firewall Support

- Some firewalls cannot support IPv6 or rely on a separate firewall to do so
- For example, the Linux iptables firewall does not support IPv6, but ip6tables (included with iptables) does
  - This means two independent firewalls must run on the same host in order to filter both IPv4 and IPv6
  - A host running iptables only (including a final DROP rule, see notes for details) will not filter any IPv6 traffic
- Firewalls that support IPv6 are often laxer than IPv4 firewalls
  - Some IPv6 firewalls cannot block inbound ICMPv6

### IPv6 Firewall Support

The Linux iptables firewall could be the world's most common firewall: it is supported in the Linux kernel by default since version 2.4 (released in 2001). It is being supplanted with the nftables firewall (see: [https://wiki.nftables.org/wiki-nftables/index.php/Main\\_Page](https://wiki.nftables.org/wiki-nftables/index.php/Main_Page)), which allows 'one-stop shopping' for IPv4 and IPv6.

Here is a simple iptables input filter, allowing TCP ports 22 and 25, and dropping the rest:

```
-A INPUT -i enp1s0 -p tcp --dport 22 -j ACCEPT
-A INPUT -i enp1s0 -p tcp --dport 25 -j ACCEPT
-A INPUT -j DROP
```

This filter will allow all IPv6 through since the final DROP rule applies to IPv4 only.

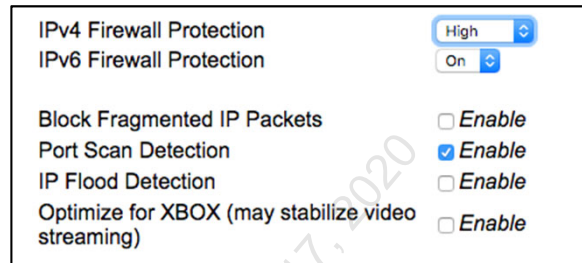
ip6tables is designed to filter IPv6 traffic and needs to be run in conjunction with iptables to filter IPv6. The course authors have seen client firewalls running iptables only and passing all IPv6 traffic.



## Ubee IPv6 Firewall



- This Ubee cable modem firewall cannot block inbound ICMPv6
  - IPv6 Firewall protection is "On"
  - The screenshot on the lower left shows the author's MacBook Pro IPv6 address: `2604:6000:8680:3d00:78:f48c:1b9d:77c4`
  - The screenshot on the lower right shows an ICMPv6 ping from the internet



```
Orion:~ ericconrad$ ifconfig en0
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether a0:99:9b:08:2e:85
    inet6 fe80::471:ddb:d:b001:50ae%en0 prefixlen 64 secured scopeid 0x7
    inet 192.168.0.2 netmask 0xfffff00 broadcast 192.168.0.255
    inet6 2604:6000:8680:3d00:78:f48c:1b9d:77c4 prefixlen 64 autoconf secured
    inet6 2604:6000:8680:3d00:2b12:9bb0:4e45:a4e4 prefixlen 64 autoconf temporary
    inet6 2604:6000:8680:3d00::11 prefixlen 64 dynamic
    nd6 options=281<PERFORMNUD,DAD>
    media: autoselect
    status: active
Orion:~ ericconrad$
```

```
root@eic:~# ping6 -c4 2604:6000:8680:3d00:78:f48c:1b9d:77c4
PING 2604:6000:8680:3d00:78:f48c:1b9d:77c4(2604:6000:8680:3d00:78:f48c:1b9d:77c4) 56 data bytes
64 bytes from 2604:6000:8680:3d00:78:f48c:1b9d:77c4: icmp_seq=1 ttl=52 time=51.5 ms
64 bytes from 2604:6000:8680:3d00:78:f48c:1b9d:77c4: icmp_seq=2 ttl=52 time=49.6 ms
64 bytes from 2604:6000:8680:3d00:78:f48c:1b9d:77c4: icmp_seq=3 ttl=52 time=47.0 ms
64 bytes from 2604:6000:8680:3d00:78:f48c:1b9d:77c4: icmp_seq=4 ttl=52 time=114 ms

--- 2604:6000:8680:3d00:78:f48c:1b9d:77c4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 47.070/65.733/114.689/28.310 ms
root@eic:~#
```

### Ubee IPv6 Firewall

Many SOHO (Small Office/Home Office) firewalls that support IPv6 have limited firewalls. The author's Ubee firewall was deployed by a cable modem technician, and the original configuration that the IPv6 Firewall Protection set to "Off". Every device IPv6 on the author's home network was open to the IPv6 internet: AppleTV, iPhones, tablets, laptops, etc. The author SSHed via IPv6 to his MacBook Pro from the internet to test.

The only other IPv6 firewall setting is "On", so the author chose that. That blocked TCP and UDP traffic via IPv6, but not ICMPv6, as shown above. This means tunneling traffic via ICMPv6 is trivial. This may not be a top concern for a home network, but it is a much bigger concern for sensitive networks.

## Scanning IPv6

- The process for actively scanning IPv6 networks for host discovery is fundamentally different than scanning IPv4 networks
- IPv4-style end-to-end ping sweeps are not possible due to the size of the subnets
  - If you could ping one host per second ...
  - It would take 584 billion years to scan 18+ quintillion IPv6 addresses on a /64 network
- IPv6 multicast address become critical for performing local host discovery
  - Some older methods, such as switch CAM (Content Addressable Memory) inspection, and passive scanning, still work
  - IPv6 does not use ARP (it uses neighbor discovery via multicast, as we will discuss next), but dual-stack systems may be discovered via ARP

### Scanning IPv6

With a tool like nmap (and reasonably fast network), scanning virtually any IPv4 network is feasible in a reasonable amount of time. A ping sweep of a /24 network (256 IP addresses) could take minutes. For example: ping 192.168.0.1, 192.168.0.1, 192.168.0.02... 192.168.0.255).

Scanning /16 (class B, 65,536 hosts) and /8 (class A, 16.7 million hosts) networks are straightforward (but will take more time).

Fast scanning tools like Robert Graham's masscan (<https://github.com/robertdavidgraham/masscan>) are used to scan the entire IPv4 Internet on a routine basis.

The sheer size of IPv6 subnets makes end-to-end scanning infeasible, given the sheer numbers, as the slide above notes. Pinging faster than 1 per second is certainly feasible but speeding the scan up a million-fold (for example) would do nothing to address the overall issue: none of us will be here when the scan completes.

A scanner could narrow down the address space: for example, if a site SLAAC (Stateless Address Auto Configuration) without privacy extensions, then the IPv6 addresses are based on the MAC address, which narrows the address space to scan.

This could help, but the problem is still massive, and most sites use privacy extensions for addresses other than link local. Other scanning methods are required.

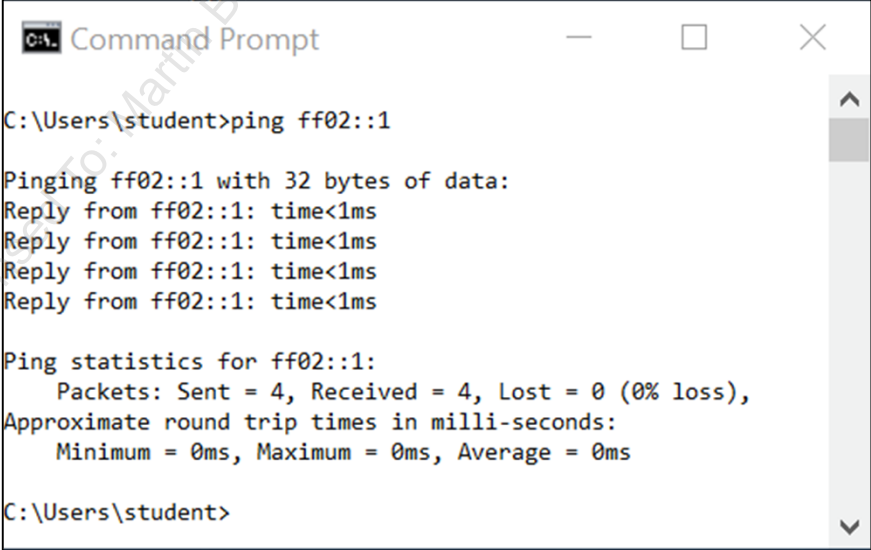
## IPv6 Scanning Tools

- While end-to-end scans of IPv6 networks are not effective, the following methods are helpful
  - IPv6 ping to multicast addresses
  - Inspecting the IPv6 neighbor discovery protocol (NDP) table
  - Inspecting the IPv6 route tables
- The following tools may be used to scan/discover IPv6 (details to follow):
  - ping6 (Linux, UNIX, macOS)
  - ip (Linux, UNIX)
  - netsh.exe (Windows)
  - netstat and ndp (macOS)
  - nmap
  - Metasploit

### IPv6 Scanning Tools

Microsoft Windows operating systems now use ping.exe for both IPv4 and IPv6, but the IPv6 support is minimal. Window's ping.exe can ping IPv6 multicast addresses, for example, but will only show one response (from the multicast address).

For example:



```
C:\Users\student>ping ff02::1

Pinging ff02::1 with 32 bytes of data:
Reply from ff02::1: time<1ms
Reply from ff02::1: time<1ms
Reply from ff02::1: time<1ms
Reply from ff02::1: time<1ms

Ping statistics for ff02::1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\student>
```

netsh.exe can be used to display the IPv6 route and neighbor discovery protocol tables, as we will discuss next

## Scanning IPv6 via Multicast

- The ping6 command can ping multicast addresses
  - If more than one response is received (common when pinging multicast addresses): some versions of ping6 will warn "(DUP!)"
- This command pings the all local nodes multicast address

```
Terminal - econrad@xubuntu-16-04: ~
File Edit View Terminal Tabs Help
[~]$ ping6 -c3 -I eth0 ff02::1
PING ff02::1(ff02::1) from fe80::2a0f:62ab:b0f7:7d46 eth0: 56 data bytes
64 bytes from fe80::2a0f:62ab:b0f7:7d46: icmp_seq=1 ttl=64 time=0.023 ms
64 bytes from fe80::18e6:6f21:253a:a069: icmp_seq=1 ttl=64 time=0.248 ms (DUP!)
64 bytes from fe80::20c:29ff:fe52:4ba6: icmp_seq=1 ttl=64 time=0.562 ms (DUP!)
64 bytes from fe80::22c9:d0ff:fe16:7029: icmp_seq=1 ttl=64 time=0.653 ms (DUP!)
64 bytes from fe80::22a:e3ff:fecc:a22d: icmp_seq=1 ttl=64 time=0.751 ms (DUP!)
64 bytes from fe80::2a0f:62ab:b0f7:7d46: icmp_seq=2 ttl=64 time=0.038 ms
64 bytes from fe80::18e6:6f21:253a:a069: icmp_seq=2 ttl=64 time=0.204 ms (DUP!)
64 bytes from fe80::20c:29ff:fe52:4ba6: icmp_seq=2 ttl=64 time=0.366 ms (DUP!)
64 bytes from fe80::22c9:d0ff:fe16:7029: icmp_seq=2 ttl=64 time=0.524 ms (DUP!)
64 bytes from fe80::22a:e3ff:fecc:a22d: icmp_seq=2 ttl=64 time=0.762 ms (DUP!)
64 bytes from fe80::2a0f:62ab:b0f7:7d46: icmp_seq=3 ttl=64 time=0.041 ms

--- ff02::1 ping statistics ---
3 packets transmitted, 3 received, +8 duplicates, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.023/0.379/0.762/0.273 ms
[~]$
```

### Scanning IPv6 via Multicast

Notes on the command shown above: `ping6 -c3 -I eth0 ff02::1`

As noted previously: most operating systems (except Windows) use separate utilities for sending ICMP echo requests (ping) and ICMPv6 echo requests (ping6).

We send three echo requests ("-c3"), and there are multiple responses per echo request, which is common when pinging multicast requests.

A multicast address could be valid on any interface, so the interface must be specified with "-I <interface name>".

Finally, we are pinging the all nodes multicast address: `ff02::1`.

This command pings all local routers (there is usually 1, so there are no duplicate replies)

```
Terminal - econrad@xubuntu-16-04: ~
File Edit View Terminal Tabs Help
[~]$ ping6 -c3 -I eth0 ff02::2
PING ff02::2(ff02::2) from fe80::2a0f:62ab:b0f7:7d46 eth0: 56 data bytes
64 bytes from fe80::22a:e3ff:fecc:a22d: icmp_seq=1 ttl=64 time=0.748 ms
64 bytes from fe80::22a:e3ff:fecc:a22d: icmp_seq=2 ttl=64 time=0.781 ms
64 bytes from fe80::22a:e3ff:fecc:a22d: icmp_seq=3 ttl=64 time=0.723 ms

--- ff02::2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.723/0.750/0.781/0.039 ms
[~]$
```

### One Limitation with IPv6 Multicast Scanning

- IPv6 Multicast addresses that begin with "ff02::" operate at the Link-Local (LAN) scope
- Local host discovery is simpler due to the use of dual-stack systems (with both IPv4 and IPv6 addresses)
  - In that case, an IPv4 ping or ARP (Address Resolution Protocol) sweep of the IPv4 subnet is likely to identify the same dual-stack systems with a multicast scan of ff02::1 (all local nodes) and ff02::2 (all local routers)
- While IPv6 multicast addresses exist at larger scopes, they are not as commonly used
- This means scanning non-local IPv6 systems can be challenging

### One Limitation with IPv6 Multicast Scanning

As noted above: scanning local IPv6 systems is easy. Most systems are dual-stack, running both IPv4 and IPv6. This means discovering local systems via traditional methods was already easy: a simple ARP sweep or ping scan will likely discover all systems on a local subnet.

Discovering non-local IPv6 systems is much more challenging. Larger-scope IPv6 multicast addresses are rarely used. End-to-end sweeps of /64 networks are not feasible: ping .1, then .2, then .3... and the Sun will supernova before a sweep of the 18+ quintillion addresses on a /64 subnet will complete.

One method for discovering remote IPv6 systems: rely on dual-stack systems and use IPv4 scans.

What happens if an organization does *\*not\** run dual-stack, and has some IPv6-only servers? These will be very difficult to discover if they are not on the local subnet and are not discoverable through other traditional reconnaissance and scanning methods (such as DNS, Google searches, etc.).

## Listing the IPv6 Neighbor Discovery Protocol Table

- Windows **Ethernet0** results for: **netsh interface ipv6 show neighbors**

Internet Address	Physical Address	Type
2001:470:1f11:78e:82d6:cdd3:d485:fd22	00-00-00-00-00-00	Unreachable
fdfe:9e87:9d56:1000:20c:29ff:fe52:4ba6	00-00-00-00-00-00	Unreachable
fe80::20c:29ff:fe52:4ba6	00-0c-29-52-4b-a6	Stale
fe80::22a:e3ff:fecc:a22d	00-2a-e3-cc-a2-2d	Stale (Router)
fe80::18e6:6f21:253a:a069	38-c9-86-1f-9a-d7	Stale
fe80::22c9:d0ff:fe16:7029	20-c9-d0-16-70-29	Stale
fe80::2a0f:62ab:b0f7:7d46	00-0c-29-cb-ff-d8	Stale
ff02::1	33-33-00-00-00-01	Permanent
ff02::2	33-33-00-00-00-02	Permanent
ff02::c	33-33-00-00-00-0c	Permanent
ff02::16	33-33-00-00-00-16	Permanent
ff02::1:2	33-33-00-01-00-02	Permanent
ff02::1:3	33-33-00-01-00-03	Permanent
ff02::1:ff16:7029	33-33-ff-16-70-29	Permanent
ff02::1:ff52:4ba6	33-33-ff-52-4b-a6	Permanent
ff02::1:ff93:637b	33-33-ff-93-63-7b	Permanent
ff02::1:fff9:e1cd	33-33-ff-99-e1-cd	Permanent
ff02::1:ffcc:a22d	33-33-ff-cc-a2-2d	Permanent
ff02::1:fff7:7d46	33-33-ff-f7-7d-46	Permanent

## Listing the IPv6 Neighbor Discovery Protocol Table

The screenshot above shows the results of '**netsh interface ipv6 show neighbors**'. Note that a list of common multicast addresses was shown previously in the IPv6 Multicast Addresses slide. We will show a summary of Windows, Linux, and macOS/BSD commands at the end of this section.

Here are the results of '**ndp -an**' on macOS High Sierra, showing the neighbor discovery protocol table.

Neighbor	Linklayer Address	Netif	Expire	St	Flgs	Prbs
2001:470:1f11:78e:4c:a13:3711:1579	38:c9:86:1f:9a:d7	en4	permanent	R		
2001:470:1f11:78e:a481:cb9:126d:e29f	38:c9:86:1f:9a:d7	en4	permanent	R		
2604:6000:8680:3d00::d	a0:99:9b:8:2e:85	en0	permanent	R		
2604:6000:8680:3d00:78:f48c:1b9d:77c4	a0:99:9b:8:2e:85	en0	permanent	R		
2604:6000:8680:3d00:c9ad:e5e6:202c:adc7	a0:99:9b:8:2e:85	en0	permanent	R		
fdfe:9e87:9d56:1000:4b4:7f27:9a75:2d0d	38:c9:86:1f:9a:d7	en4	permanent	R		
fdfe:9e87:9d56:1000:808c:cb72:fe73:edaa	38:c9:86:1f:9a:d7	en4	permanent	R		
fe80::1%lo0	(incomplete)	lo0	permanent	R		
fe80::14ab:5759:5f85:ce30%en0	40:cb:c0:c8:36:a	en0	20h45m8s	S		
fe80::1cbd:686d:f420:b304%en0	a0:99:9b:8:2e:85	en0	permanent	R		
fe80::3612:98ff:fe01:5535%en0	34:12:98:1:55:35	en0	20h45m13s	S		
fe80::461c:a8ff:fed1:c954%en0	44:1c:a8:d1:c9:54	en0	21h3m2s	S	R	
fe80::7864:cbff:fe0e:69b1%awdl0	7a:64:cb:e:69:b1	awdl0	permanent	R		
fe80::23c0:be87:de84:6781%utun0	(incomplete)	utun0	permanent	R		
fe80::9763:681f:2c4e:b107%utun1	(incomplete)	utun1	permanent	R		
fe80::779c:62da:8f52:897d%utun2	(incomplete)	utun2	permanent	R		
fe80::708:2884:b8ab:5cc8%utun3	(incomplete)	utun3	permanent	R		
fe80::1684:325e:562a:d4ce%utun4	(incomplete)	utun4	permanent	R		
fe80::7129:d8d7:728e:1363%utun5	(incomplete)	utun5	permanent	R		
fe80::22a:e3ff:fecc:a22d%en4	0:2a:e3:cc:a2:2d	en4	3s	R	R	
fe80::9cb:65fb:5d99:e1cd%en4	0:c:29:b5:fc:d5	en4	23h13m37s	S		
fe80::18e6:6f21:253a:a069%en4	38:c9:86:1f:9a:d7	en4	permanent	R		
fe80::22c9:d0ff:fe16:7029%en4	20:c9:d0:16:70:29	en4	20h45m4s	S		
fe80::2a0f:62ab:b0f7:7d46%en4	0:c:29:cb:ff:d8	en4	22h8m15s	S		
fe80::9b43:fb73:ddc5:a6f5%utun6	(incomplete)	utun6	permanent	R		

## Listing the IPv6 Route Table

- Windows results for: `netsh interface ipv6 show route`

```

Command Prompt
C:\Users\student>netsh interface ipv6 show route
-----
Publish Type Met Prefix Idx Gateway/Interface Name
-----
No Manual 256 ::/0 4 fe80::22a:e3ff:fecc:a22d
No System 256 ::1/128 1 Loopback Pseudo-Interface 1
No Manual 256 2001::/32 10 Teredo Tunneling Pseudo-Interface
No System 256 2001:0:9d38:6ab8:e8:1da4:f59c:9c5d/128 10 Teredo Tunneling Pseudo-Interface
No Manual 256 2001:470:1f11:78e::/64 4 Ethernet0
No System 256 2001:470:1f11:78e:9cb:65fb:5d99:e1cd/128 4 Ethernet0
No System 256 2001:470:1f11:78e:49e4:c6c:d193:637b/128 4 Ethernet0
No Manual 256 fdfe:9e87:9d56:1000::/64 4 Ethernet0
No System 256 fdfe:9e87:9d56:1000:9cb:65fb:5d99:e1cd/128 4 Ethernet0
No System 256 fdfe:9e87:9d56:1000:49e4:c6c:d193:637b/128 4 Ethernet0
No System 256 fe80::/64 8 Bluetooth Network Connection
No System 256 fe80::/64 4 Ethernet0
No System 256 fe80::/64 5 Ethernet
No System 256 fe80::/64 10 Teredo Tunneling Pseudo-Interface
No System 256 fe80::5efe:10.99.99.162/128 7 isatap.12colonies.org
No System 256 fe80::e8:1da4:f59c:9c5d/128 10 Teredo Tunneling Pseudo-Interface
No System 256 fe80::9cb:65fb:5d99:e1cd/128 4 Ethernet0
No System 256 fe80::34d6:d586:efb3:50f5/128 5 Ethernet
No System 256 fe80::e9b5:75d:6b37:76bd/128 8 Bluetooth Network Connection
No System 256 ff00::/8 1 Loopback Pseudo-Interface 1
No System 256 ff00::/8 8 Bluetooth Network Connection
No System 256 ff00::/8 4 Ethernet0
No System 256 ff00::/8 5 Ethernet
No System 256 ff00::/8 10 Teredo Tunneling Pseudo-Interface
-----
C:\Users\student>

```

## Listing the IPv6 Route Table

The screenshot above shows the Windows results of '`netsh interface ipv6 show route`'.

Here are the results of `netstat -A inet6 -rn` on Ubuntu Linux, showing the IPv6 route table. The "`-rn`" flags choose the routing ("r") table and do not ("n") resolve names.

```

Terminal - econrad@xubuntu-16-04:~
File Edit View Terminal Tabs Help
[~]$ netstat -A inet6 -rn
Kernel IPv6 routing table
Destination Next Hop Flag Met Ref Use If
2001:470:1f11:78e::/64 :: Ue 256 1 818 eth0
fdfe:9e87:9d56:1000::/64 :: Ue 256 1 302 eth0
fe80::/64 :: U 256 1 651 eth0
::/0 fe80::22a:e3ff:fecc:a22d UG 100 1 2 eth0
::/0 !n -1 1 2048 lo
::1/128 :: Un 0 2 8 lo
2001:470:1f11:78e:40aa:7b2c:282d:2422/128 :: Un 0 2 794 lo
2001:470:1f11:78e:82d6:cdd3:d485:fd22/128 :: Un 0 2 3 lo
fdfe:9e87:9d56:1000:40aa:7b2c:282d:2422/128 :: Un 0 2 106 lo
fdfe:9e87:9d56:1000:448c:9f44:d81d:d7f6/128 :: Un 0 2 2 lo
fe80::2a0f:62ab:b0f7:7d46/128 :: Un 0 2 904 lo
ff00::/8 :: U 256 1 302 eth0
::/0 !n -1 1 2048 lo
[~]$

```

## Summary of Native Operating System IPv6 Discovery Tools

- Here are a summary of the Windows, Linux and MacOS/BSD commands to perform IPv6 pings, and to show the IPv6 NDP (Network Discovery Protocol) and IPv6 route tables<sup>1</sup>

	IPv6 ping	IPv6 NDP Table	IPv6 Route Table
Windows	<code>ping</code>	<code>netsh interface ipv6 show neighbors</code>	<code>netsh interface ipv6 show route</code>
Linux	<code>ping6</code>	<code>ip -6 neighbor show</code>	<code>netstat -A inet6 -rn</code>
macOS/BSD	<code>ping6</code>	<code>ndp -an</code>	<code>netstat -f inet6 -rn</code>

### Summary of Native Operating System IPv6 Discovery Tools

As noted above: Windows, Linux and macOS/BSD use a variety of (mostly) different tools to ping, list the IPv6 neighbor discovery table, and the IPv6 route table.

Thanks to the University of Wisconsin, for their excellent summary of IPv6 tools. The table shown above is based on their "Network Troubleshooting Tools, IPv4 and IPv6", for more information:  
<https://kb.wisc.edu/ns/page.php?id=12364>

Universities have long been leaders in IPv6 deployment, Virginia Tech University was one of the earliest and biggest adopters of IPv6. Randy Marchany (VTU University Information Technology Security Officer and SANS instructor) mentioned this fun fact to a course author:

*Virginia Tech is currently a leader in a large-scale production deployment of IPv6, with nearly thirteen years of experience and thousands of native IPv6 clients. In 2010, the university was ranked by Google as one of the largest deployments worldwide, behind only nations like France and China. Globally reachable by IPv6, Virginia Tech faculty are able to conduct research around the world using IPv6-only networks.<sup>2</sup>*

[1] <https://kb.wisc.edu/ns/page.php?id=12364>

[2] <http://ipv6.cns.vt.edu/>



## Scanning IPv6 with nmap

- nmap's IPv6 support was strongly improved beginning with nmap 7.0
- nmap now has a number of NSE (nmap scripting engine) scripts for discovering IPv6

```

ericconrad — root@eic: ~ — -bash — 116x14
Orion:~ ericconrad$ sudo nmap -6 --script=targets-ipv6-multicast-echo.nse --script-args 'newtargets,interface=en0'

Starting Nmap 7.31 ( https://nmap.org ) at 2018-02-10 16:32 EST
Pre-scan script results:
| targets-ipv6-multicast-echo:
| IP: fe80::3612:98ff:fe01:5535 MAC: 34:12:98:01:55:35 IFACE: en0
| IP: fe80::461c:a8ff:fed1:c954 MAC: 44:1c:a8:d1:c9:54 IFACE: en0
| IP: fe80::20e:c4ff:fece:2b82 MAC: 00:0e:c4:ce:2b:82 IFACE: en0
| IP: fe80::c72:84e8:cf63:4206 MAC: 28:f0:76:2f:c3:ce IFACE: en0
| IP: fe80::76c2:46ff:fe3c:2282 MAC: 74:c2:46:3c:22:82 IFACE: en0
| IP: fe80::14e8:91bc:fc10:d4e3 MAC: 68:ef:43:39:3e:91 IFACE: en0
| IP: fe80::1001:2570:761d:f020 MAC: 48:a1:95:1a:48:cd IFACE: en0
| IP: fe80::10f8:94f8:8cae:6261 MAC: 84:fc:ac:e5:34:41 IFACE: en0
| IP: fe80::32a9:deff:fee7:2014 MAC: 30:a9:de:e7:20:14 IFACE: en0
|_

```

### Scanning IPv6 with nmap

nmap ipv6 scanning scripts include:

targets-ipv6-multicast-echo.nse

- *"ends an ICMPv6 echo request packet (ICMPv6 Type 128) to the all-nodes link-local multicast address (ff02::1) to discover responsive hosts on a LAN without needing to ping each IPv6 address individually."*<sup>1</sup>

targets-ipv6-multicast-mld.nse

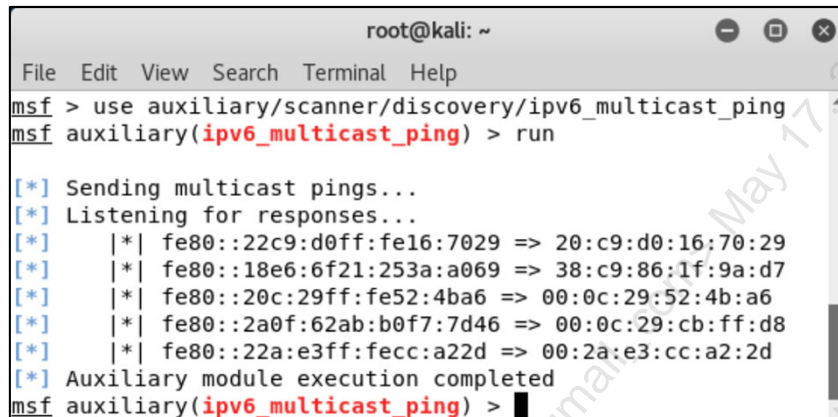
- *Attempts to discover available IPv6 hosts on the LAN by sending an MLD (multicast listener discovery, ICMPv6 Type 130) query to the link-local multicast address(ff02::1) and listening for any responses."*<sup>2</sup>
- Note: this is the same approach as targets-ipv6-multicast-echo.nse, but using a different ICMPv6 request type

[1] <https://nmap.org/nsedoc/scripts/targets-ipv6-multicast-echo.html>

[2] <https://nmap.org/nsedoc/scripts/targets-ipv6-multicast-mld.html>

## Scanning IPv6 with Metasploit

- Metasploit can perform IPv6 multicast scans with the "ipv6\_multicast\_ping" module
  - Other Metasploit IPv6 scanning modules are described in the notes



```

root@kali: ~
File Edit View Search Terminal Help
msf > use auxiliary/scanner/discovery/ipv6_multicast_ping
msf auxiliary(ipv6_multicast_ping) > run

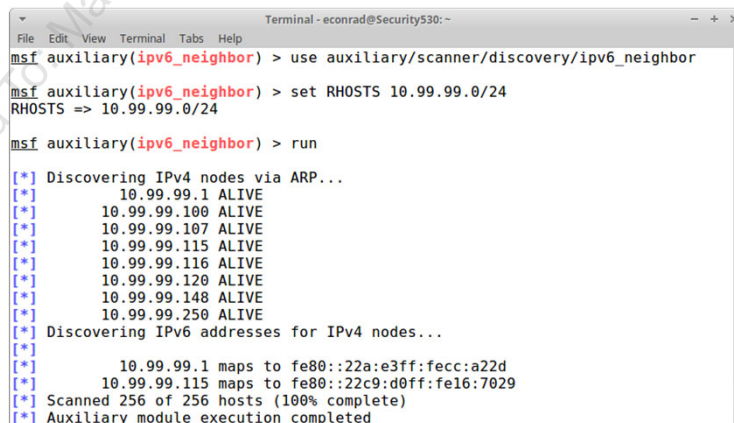
[*] Sending multicast pings...
[*] Listening for responses...
[*]   [*] fe80::22c9:d0ff:fe16:7029 => 20:c9:d0:16:70:29
[*]   [*] fe80::18e6:6f21:253a:a069 => 38:c9:86:1f:9a:d7
[*]   [*] fe80::20c:29ff:fe52:4ba6 => 00:0c:29:52:4b:a6
[*]   [*] fe80::2a0f:62ab:b0f7:7d46 => 00:0c:29:cb:ff:d8
[*]   [*] fe80::22a:e3ff:fecc:a22d => 00:2a:e3:cc:a2:2d
[*] Auxiliary module execution completed
msf auxiliary(ipv6_multicast_ping) >

```

## Scanning IPv6 with Metasploit

Other Metasploit IPv6 scanners include:

- `ipv6_neighbor_router_advertisement`: "Send a spoofed router advertisement with high priority to force hosts to start the IPv6 address auto-config. Monitor for IPv6 host advertisements, and try to guess the link-local address by concatenating the prefix, and the host portion of the IPv6 address. Use NDP host solicitation to determine if the IP address is valid"<sup>1</sup>
- `ipv6_neighbor` (screenshot below): "Enumerate local IPv6 hosts which respond to Neighbor Solicitations with a link-local address. Note, that like ARP scanning, this usually cannot be performed beyond the local broadcast network."<sup>2</sup>



```

Terminal - econrad@Security530: ~
File Edit View Terminal Tabs Help
msf auxiliary(ipv6_neighbor) > use auxiliary/scanner/discovery/ipv6_neighbor
msf auxiliary(ipv6_neighbor) > set RHOSTS 10.99.99.0/24
RHOSTS => 10.99.99.0/24
msf auxiliary(ipv6_neighbor) > run

[*] Discovering IPv4 nodes via ARP...
[*] 10.99.99.1 ALIVE
[*] 10.99.99.100 ALIVE
[*] 10.99.99.107 ALIVE
[*] 10.99.99.115 ALIVE
[*] 10.99.99.116 ALIVE
[*] 10.99.99.120 ALIVE
[*] 10.99.99.148 ALIVE
[*] 10.99.99.250 ALIVE
[*] Discovering IPv6 addresses for IPv4 nodes...
[*] 10.99.99.1 maps to fe80::22a:e3ff:fecc:a22d
[*] 10.99.99.115 maps to fe80::22c9:d0ff:fe16:7029
[*] Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed

```

[1] [https://github.com/rapid7/metasploit-framework/blob/master/modules/auxiliary/scanner/discovery/ipv6\\_neighbor\\_router\\_advertisement.rb](https://github.com/rapid7/metasploit-framework/blob/master/modules/auxiliary/scanner/discovery/ipv6_neighbor_router_advertisement.rb)

[2] [https://github.com/rapid7/metasploit-framework/blob/master/modules/auxiliary/scanner/discovery/ipv6\\_neighbor.rb](https://github.com/rapid7/metasploit-framework/blob/master/modules/auxiliary/scanner/discovery/ipv6_neighbor.rb)

## IPv6 Tunneling Options

There are a wide variety of IPv6 tunneling options, including:

- 6to4, 6in4 and 6over4
  - 6rd
  - 4 over 6
  - Teredo
  - ISATAP
  - GRE
- The wide variety and types of IPv6 tunnels make both preventing and detecting them difficult
  - These tunnels can also be used to evade or bypass controls such as IDSes and IPSes

### IPv6 Tunneling Options

Here is a brief summary of the various IPv6 tunneling options described above. They all use IP protocol 41 (IPv6 encapsulation):

- 6to4: Tunnels IPv6 packets via IPv4 networks, without setting up a tunnel
- 6in4: Uses pre-configured tunnels to carry IPv6 traffic via IPv4
- 6over4: Similar to 6in4, requires IPv4 multicast. Designed for intranet (private) use
- 6rd (rapid deployment): Based on 6to4, used to carry IPv6 packets via private IPv4 networks
- 4over6: Tunnels ipv4 packets via IPv6-only networks
- Teredo: Similar to 6to4 and considered an improvement. Carries IPv6 packets via UDP (and IPv4). Able to traverse NAT gateways. It's considered high cost due to packet overhead
- ISATAP (Intra-Site Automatic Tunnel Addressing Protocol): similar to 6over4, uses DNS instead of multicast. Designed for intranet (private) use
- GRE: can tunnel both IPv4 and IPv6 traffic
- Most SSL/TLS VPN solutions (including OpenVPN) can tunnel both IPv4 and IPv6

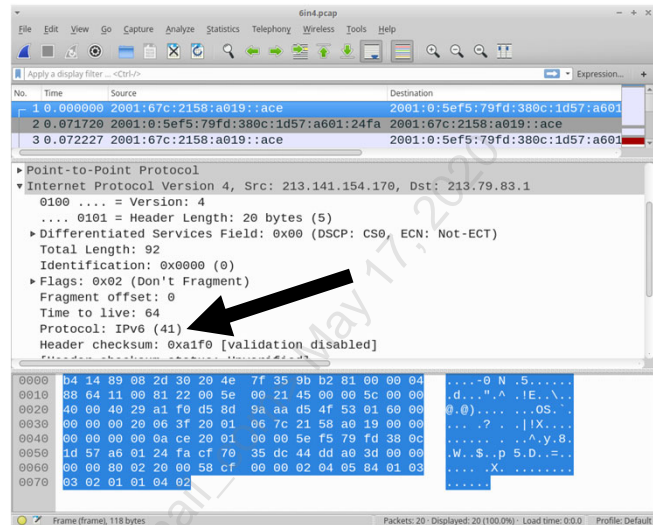
Note that 6to4, 6in4, and 6rd cannot traverse NAT gateways without workarounds

RFC 7059 ("A Comparison of IPv6-over-IPv4 Tunnel Mechanisms"<sup>1</sup>) has a great summary of the various IPv6 tunneling mechanisms.

[1] <https://tools.ietf.org/html/rfc7059>

## Preventing and Detecting IPv6 Tunneling

- Many forms of IPv6 via IPv4 tunnels carry IPv6 where TCP or UDP would normally be
  - The layer 3 header "Protocol" field would be 41 (IPv6) in this case
- Configure Next-Gen Firewalls, IDSes and/or IPSes to block/alert protocol 41
  - Snort syntax: `ip_proto:41`



### Preventing and Detecting IPv6 Tunneling

Many forms of IPv6 via IPv4 tunnels carry IPv6 where TCP (protocol 6) or UDP (protocol 17) would normally be. This includes both 6in4 and 6to4 tunnels.

In this case, the next layer protocol will be 41 (IPv6). The snort syntax option "`ip_proto:41`" will detect these types of tunnels.

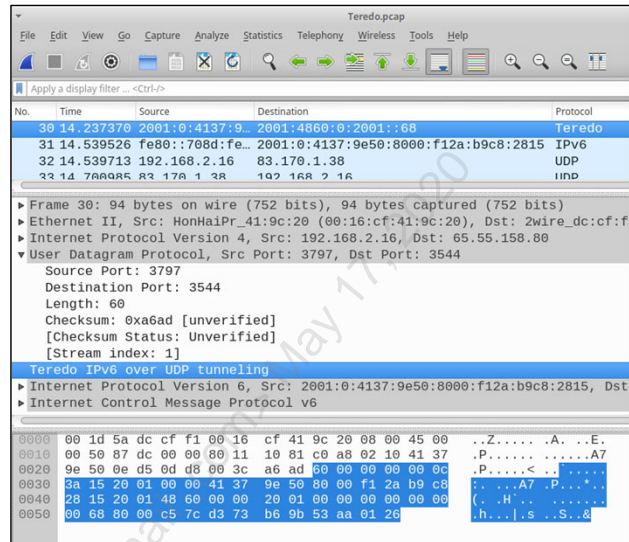
Here is an Emerging Threat open rule for detecting or blocking protocol 41 (IPv6 carried by IPv4) using Suricata, Snort, Sourcefire, Cisco Firepower (and others):

```
alert ip $HOME_NET any -> $EXTERNAL_NET any (msg:"ET POLICY Protocol
41 IPv6 encapsulation potential 6in4 IPv6 tunnel active";
ip_proto:41; threshold:type both,track by_dst, count 1, seconds 60;
reference:url,en.wikipedia.org/wiki/6in4; classtype:policy-violation;
sid:2012141; rev:2; metadata:created_at 2011_01_05, updated_at
2011_01_05;)
```

[1] <https://rules.emergingthreats.net/open/snort-2.9.0/emerging-all.rules>

## Preventing and Detecting Teredo Tunneling

- Teredo was originally developed by Microsoft and is standardized in RFC 4380<sup>1</sup>
- It uses UDP port 3544 by default, but other UDP ports may be used
- The Wireshark display filter "teredo" will detect Teredo tunnels via any UDP port
  - See notes for the Snort rule



### Preventing and Detecting Teredo Tunneling

Here is an Emerging Threat rule for detecting Teredo tunneling:

```
alert udp $HOME_NET any -> $EXTERNAL_NET 3544 (msg:"ET POLICY
Microsoft TEREDO IPv6 tunneling"; content:"|FE 80 00 00 00 00 00 00
80 00|TEREDO"; offset:21; depth:16;
reference:url,doc.emergingthreats.net/2003155; classtype:misc-
activity; sid:2003155; rev:4; metadata:created_at 2010_07_30,
updated_at 2010_07_30;)2
```

In the author's experience: the above rule will sometimes have false negatives when detecting Teredo. This Snort rule will detect more Teredo, at the risk of more false positives:

```
alert udp $HOME_NET any -> $EXTERNAL_NET 3544 (msg:"POLICY-OTHER
Outbound Teredo traffic detected"; flow:to_server; content:" |01|";
depth:2; offset:8; byte_test:1,&,96,0; reference:cve,2007-3038;
reference:url,technet.microsoft.com/en-us/security/bulletin/MS07-038;
classtype:policy-violation; sid:12065; rev:5;)3
```

Change "\$EXTERNAL\_NET 3544" to "\$EXTERNAL\_NET any" in either rule to detect Teredo via other UDP ports.

[1] <https://tools.ietf.org/html/rfc4380>

[2] <https://doc.emergingthreats.net/2003155>

[3] <https://github.com/John-Lin/docker-snort/blob/master/snortrules-snapshot-2972/rules/policy-other.rules>

## Preventing and Detecting IPv6 via IPv4 Tunnels with Cisco

- This Cisco IOS ACL will allow and log Protocol 41 (IPv6 via IPv4) and UDP port 3544 traffic<sup>1</sup>:

```
Router(config)# interface GigabitEthernet0/1
Router(config-if)# ip address 10.5.30.1 255.255.255.0
Router(config-if)# ip access-group DetectIPv6 out
Router(config-if)# ip access-list extended DetectIPv6
Router(config-if)# permit 41 any any log
Router(config-if)# permit udp any any eq 3544 log
Router(config-if)# permit ip any any
```

### Preventing and Detecting IPv6 via IPv4 Tunnels with Cisco

For protocol 41: change "permit" to "deny" to drop the traffic

```
Router(config-if)# deny 41 any any log
```

Denying UDP port 3544 traffic is likely to cause collateral damage, since any protocol may choose 3544 as an ephemeral (temporary-use) port.

Cisco has a great guide called "Detecting IPv6 Tunnels in an Enterprise Network" which is well worth checking out:

*An important consideration is that IPv6 is quite likely to be already running on the enterprise network, whether that implementation was planned or not. Some important characteristics of IPv6 include:*

- *IPv6 has a mechanism to automatically assign addresses so that end systems can easily establish communications.*
- *IPv6 has several mechanisms available to ease the integration of the protocol into the network.*
- *Automatic tunneling mechanisms can take advantage of the underlying IPv4 network and connect it to the IPv6 Internet.*<sup>2</sup>

[1] [https://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/enterprise-ipv6-solution/white\\_paper\\_c11-629391.html](https://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/enterprise-ipv6-solution/white_paper_c11-629391.html)

[2] Ibid.

## Unauthorized IPv6 Router Advertisements

- In this scenario: a black hat compromises an internal client system via a phishing attack via IPv4
  - The black hat then creates a 6to4 tunnel from the compromised client to the IPv6 internet
  - The compromised client then sends IPv6 router advertisements to the local subnet, identifying the client PC as an IPv6 router
  - The local systems create a global unicast address, using the network prefix assigned by the rogue IPv6 router
- That local subnet is now directly exposed to the public IPv6 internet
  - We will illustrate this attack on the next slide
- Rogue Advertisement (RA) Guard mitigates this risk, see notes for details
  - RA Guard also mitigates DoS via IPv6 Route Advertisement flooding

### Unauthorized IPv6 Router Advertisements

Cisco describes the IPv6 RA Guard:

*The IPv6 RA Guard feature provides support for allowing the network administrator to block or reject unwanted or rogue RA guard messages that arrive at the network device platform. RAs are used by devices to announce themselves on the link. The IPv6 RA Guard feature analyzes these RAs and filters out RAs that are sent by unauthorized devices. In host mode, all RA and router redirect messages are disallowed on the port. The RA guard feature compares configuration information on the Layer 2 (L2) device with the information found in the received RA frame. Once the L2 device has validated the content of the RA frame and router redirect frame against the configuration, it forwards the RA to its unicast or multicast destination. If the RA frame content is not validated, the RA is dropped.<sup>1</sup>*

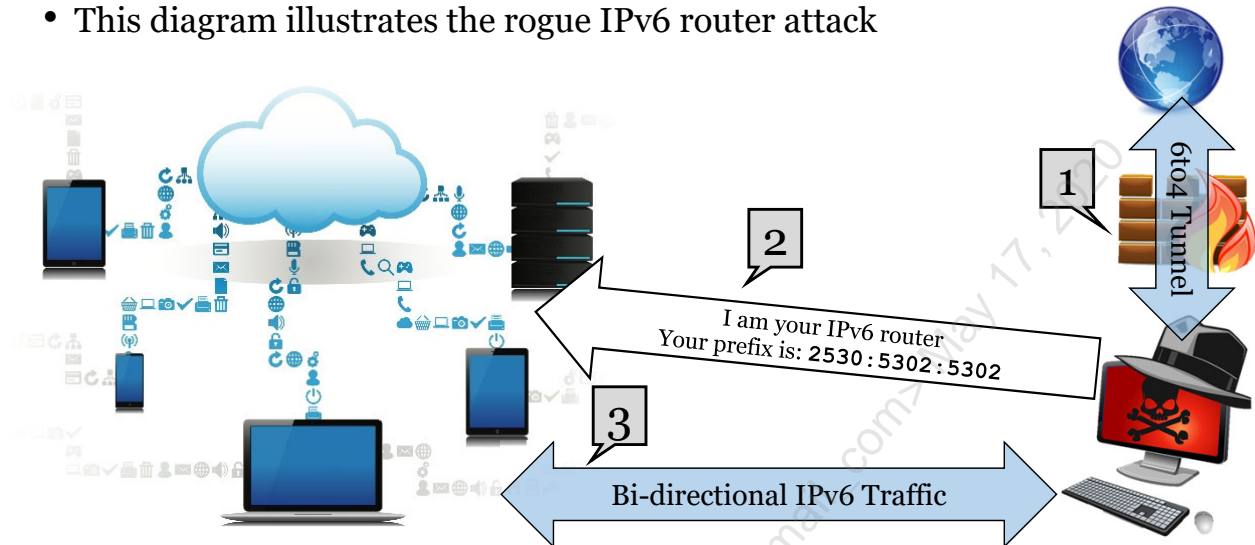
RA Guard defines 'host' and 'router' policies. Hosts are not allowed to send IPv6 Router Advertisements. Here is the syntax for a host:

```
Switch(config)#ipv6 nd rguard policy hostdevice
Switch(config-nd-raguard)#device-role host
Switch(config-nd-raguard)#exit
Switch(config)#int gig0/1
Switch(config-if)#ipv6 nd rguard attach-policy hostdevice
```

[1] [https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipv6\\_fhsec/configuration/15-s/ipv6f-15-s-book/ipv6-ra-guard.pdf](https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipv6_fhsec/configuration/15-s/ipv6f-15-s-book/ipv6-ra-guard.pdf)

## Rogue IPv6 Router Attack Illustrated

- This diagram illustrates the rogue IPv6 router attack



### Rogue IPv6 Router Attack Illustrated

The rogue IPv6 router attack is shown above.

Assume the network shown above has not configured IPv6 but has a large Windows environment, all of which is Vista or newer. This means they all automatically support IPv6.

Step 0 (not shown): compromise a host via a traditional IPv6 phishing attack.

Step 1: create a 6-to-4 tunnel to the internet, which will carry IPv6 traffic via IPv4 packets.

Step 2: send IPv6 Router Advertisements (RA) to the local subnet, advertising a public IPv6 network prefix.

Step 3: All IPv6-enabled systems that receive this router advertisement will automatically assign themselves an IPv6 global unicast address, using the provided IPv6 network prefix.

All systems on that network are now "on" the IPv6 internet. Any inbound internet traffic will pass to them unfiltered (assume the black hat is not going to firewall them). Also: the black hat has successfully performed a Man-in-the-Middle attack on all IPv6 traffic to/from the Internet.



## Hands-On IPv6

- The best way to learn (and secure) IPv6 is to use it
  - If you do not currently have public IPv6 access: ask your ISP
  - To get started: consider deploying it on test networks
- Another option: tunnel IPv6 via Hurricane Electric
  - Free service allowing up to five IPv6 tunnels via IPv4
  - Static IPv4 address is preferred on the local tunnel broker, but a dynamic address may be used
- Hurricane Electric also has a nifty (and free) IPv6 certification



### Hands-On IPv6

Note that the authors have no commercial connection to Hurricane Electric; we are simply fans of their service.

Hurricane Electric (<http://he.net>) describes their free tunnel service

*Our free tunnel broker service enables you to reach the IPv6 Internet by tunneling over existing IPv4 connections from your IPv6 enabled host or router to one of our IPv6 routers. To use this service you need to have an IPv6 capable host (IPv6 support is available for most platforms) or router which also has IPv4 (existing Internet) connectivity. Our tunnel service is oriented towards developers and experimenters that want a stable tunnel platform.<sup>1</sup>*

Once the IPv6 tunnel is set up (or you have IPv6 access via your ISP), you may attempt their free IPv6 certification, available at: <https://ipv6.he.net/certification/cert-main.php>

It's easy to set up an IPv6 tunnel with most systems, including Windows, Linux, or MacOS. You can even use a RaspberryPi: <https://www.raspberrypi.org/forums/viewtopic.php?f=36&t=88054>

[1] <https://www.tunnelbroker.net>

Re-Design & Implement on 530.2 – IPv6



- Follow NIST SP 800-119 “Guidelines for the Secure Deployment of IPv6”
- Know IPv6 capabilities of your prevention and detection tools
- Configure NGFW, IDS and IPS to block/alert on protocol 41
- Use Cisco IOS ACL to log protocol 41 and UDP port 3544
- Use Rogue Advertisement Guard to mitigate unauthorized IPv6 router advertisements

This page intentionally left blank.

## Course Roadmap

- Day 1: Defensible Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- Day 5: Zero Trust Architecture
- Day 6: Capstone: Hands-On Defend the Flag Challenge

### NETWORK SECURITY ARCHITECTURE

1. Layer 3 Attacks and Mitigation
2. Switch and Router Benchmarks
3. **EXERCISE: Auditing Router Security**
4. Securing SNMP
5. Securing NTP
6. Bogon Filtering
7. Blackholes and Darknets
8. **EXERCISE: Router SNMP Security**
9. IPv6
10. IPv6 Misconceptions
11. Securing IPv6
12. **EXERCISE: IPv6**
13. Layer 3/4 Stateful Firewall
14. Web Proxy
15. SMTP Proxy
16. **EXERCISE: Proxy Power**
17. 530.2 Summary

### Course Roadmap

We will next conduct an exercise on IPv6.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020



## Exercise 2.3: IPv6

- Exercise 2.3 is in the digital wiki found in your student VM (recommended)
- Alternatively, you may use your Workbook

### SEC530 Exercise: IPv6

We will now use IPv6 – hands on. Please go to the SEC530 lab workbook, section 2.3.

## Course Roadmap

- Day 1: Defensible Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- Day 5: Zero Trust Architecture
- Day 6: Capstone: Hands-On Defend the Flag Challenge

### NETWORK SECURITY ARCHITECTURE

1. Layer 3 Attacks and Mitigation
2. Switch and Router Benchmarks
3. EXERCISE: Auditing Router Security
4. Securing SNMP
5. Securing NTP
6. Bogon Filtering
7. Blackholes and Darknets
8. EXERCISE: Router SNMP Security
9. IPv6
10. IPv6 Misconceptions
11. Securing IPv6
12. EXERCISE: IPv6
13. Layer 3/4 Stateful Firewall
14. Web Proxy
15. SMTP Proxy
16. EXERCISE: Proxy Power
17. 530.2 Summary

### Course Roadmap

We will next discuss layer 3 and 4 stateful firewalls.

## Layer 3/4 Stateful Firewalls

- Stateful firewalls primarily inspect layers 3 (IP addresses) and layer 4 (ports, ICMP types and codes, etc.)
  - Some stateful firewalls provide limited application inspection
  - Examples include Checkpoint's SmartDefense
- Robust application inspection is provided by Next-Generation firewalls (discussed during 530.3)
- Stateful firewalls allow simple (and often inexpensive) filtering solutions, often used to augment primary firewalls

### Layer 3/4 Stateful Firewalls

Stateful firewalls are a mature technology, first introduced by Checkpoint in 1994 (their patent was filed in December 1993<sup>1</sup>). Their 'killer feature' was the state table, which tracked connections, and allowed the firewall to match responses to previous requests. Older packet filter firewalls lacked this capability and were quite porous as a result (they typically allowed all replies, and the trusting assumption that there must have been a matching request).

They are primarily a layer 3 and layer 4 control, filtering on IP addresses and ports. Many stateful firewalls offer limited application inspection, such as the old Cisco PIX 'fixup' feature, and Checkpoint's SmartDefense feature of their stateful firewalls (note that Checkpoint also makes next-generation firewalls).

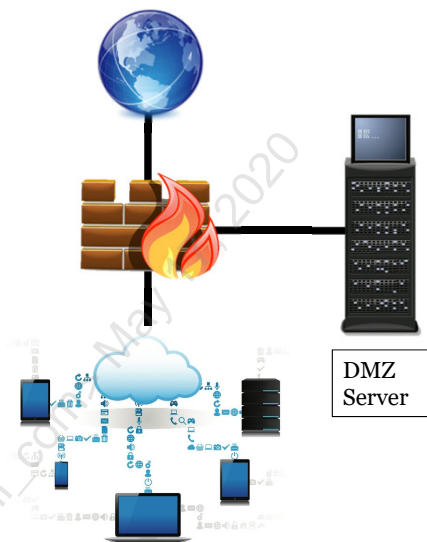
Next-generation firewalls (and/or other advanced technologies such as malware detonation devices) are required for robust application inspection.

Stateful firewalls are now built into most kernels, allowing very flexible, robust and inexpensive firewall solutions.

[1] <https://www.google.com/patents/US5606668>

## Firewall Architecture

- Simple networks with no public services can use a two-legged design, with Internet (often labeled WAN) and LAN ports
- Simple networks offering limited public services may use a three-legged firewall (adding a DMZ port)
  - DMZ (Demilitarized Zone) networks are a "no-man's land" between untrusted networks and trusted
  - More complex networks require more complex designs (and more ports)



### Firewall Architecture

Two-legged firewalls are adequate for small networks offering no incoming internet services, such as homes and some small offices. Three-legged firewalls are suitable for small networks offering limited inbound internet connectivity, such as a public web server.

More complex needs require more complex architecture. A single DMZ hosting dozens or more servers that provide public access is a risk: more DMZ segmentation is usually required at that point (to mitigate the risk of one DMZ system compromising another). This segmentation can be provided with a larger firewall providing multiple separate DMZ interfaces. It may also be provided via a DMZ switch with multiple VLANs. We will discuss DMZ design next.

NIST 800-141 (Guidelines on Firewalls and Firewall Policy) describes DMZs:

*Many hardware firewall devices have a feature called DMZ, an acronym related to the demilitarized zones that are sometimes set up between warring countries. While no single technical definition exists for firewall DMZs, they are usually interfaces on a routing firewall that are similar to the interfaces found on the firewall's protected side. The major difference is that traffic moving between the DMZ and other interfaces on the protected side of the firewall still goes through the firewall and can have firewall protection policies applied.<sup>1</sup>*

[1] <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-41r1.pdf>

## DMZ Design

- The risk of a compromised DMZ system pivoting into internal systems (or other DMZ systems) must be mitigated
  - Untrusted->DMZ access should be tightly filtered, plus DMZ->trusted
- DMZs with multiple servers should be broken up into individual trust zones (or separate DMZs)
- Private VLANs may also be used
  - Promiscuous port: the firewall DMZ interface
  - Isolated ports: DMZ servers that only need to send traffic via the firewall
  - Community ports: when multiple DMZ systems need to communicate with each other (and via the firewall)

### DMZ Design

DMZs are designed to mitigate the risk of a compromise of a DMZ system leading to the compromise of internal systems. They should also be designed to mitigate the risk of one DMZ compromise leading to the compromise of another.

In the author's experience: filtering from untrusted to DMZ is usually quite good. However, poor filtering from DMZ to internal is sadly common. A good security maxim is: any system offering inbound internet access may be compromised, and systems should be designed accordingly. There could be a zero-day exploit, for example, leading to the compromise of a public web server. The DMZ is designed to contain that compromise: securing all other trusted systems—including other DMZ systems.

Private VLANs are not normally associated with servers, but they offer an elegant solution for DMZ containment. It is quite common to have DMZ networks containing multiple servers with no need to communicate with each other, such as a public inbound mail server and a public web server. They only need to send unicast traffic via the firewall. This makes a private VLAN an ideal (and simple) solution. Configure a private VLAN on the DMZ switch, configure the firewall interface as a promiscuous port, and configure the mail and web server ports as isolated. DMZ containment is now complete.



## Beyond DMZ: Segmentation is More Than 2 Zones

- Security zones should be established according to:
  - Business and regulation requirements
    - PCI DSS requires segmentation of systems processing credit card data
  - Criticality of assets
    - Domain controllers should be segregated off user workstations
  - Threats
    - Legacy systems should be segmented off
  - Risk appetite
    - Wanna cry, anyone?

This page intentionally left blank.

## Network Segmentation Principles

- Segmentation should facilitate prevention & detection
- Systems and data with different classification levels (tiers) must reside in different zones
- Control points are implemented at "gates" where all ingress & egress traffic is inspected and access control policies enforced
- Balance security with usability
  - Higher segmentation adds complexity and administrative burden. Insufficient segmentation can make the network indefensible

This page intentionally left blank.

## Example of Tiers – Based on Criticality and Business Impact

### - Tier 1:

- Critical components to maintain operations, including domain controllers, exchange servers, and network infrastructure devices.

### - Tier 2:

- Internal systems containing PII and associated data, including databases, sharepoint servers and other web servers.

### - Tier 3:

- External facing data-providing services



This page intentionally left blank.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

## Router ACLs

- Modern routers provide layer 3/4 firewall capabilities
- Modern Cisco routers support standard and extended ACLs
  - Standard: filters on source only (layer 3)
  - Extended: filters on source or destination, as well as based on ICMP types/codes and TCP/UDP ports
- ACLs may be inbound or outbound
  - Inbound: applied to packets entering the router
  - Outbound: applied to packets before routing a packet to an outbound interface
- Syntax example is in the notes

### Router ACLs

The first step is to create an ACL. The second step is applying it to an interface.

This example will allow SSH to 10.5.30.0/24 and deny all other traffic. As previously noted: Cisco ACLs use wildcard netmasks: in this case 0.0.0.255, which is the same as 255.255.255.0 in normal netmask form,

Create the ACL:

```
Router(config)# access-list 530 permit tcp any 10.5.30.0 0.0.0.255 eq 22
Router(config)# access-list 530 deny ip any any
```

Then apply it to an interface (incoming packets):

```
Router(config)# interface GigabitEthernet0/0
Router(config-if)# ip access-group 530 in
```

## Linux and BSD Firewalls

- We will now look at Linux and BSD layer 3/4 firewall options
- Understanding these firewalls is important, even for organizations that don't think of themselves as Linux or BSD shops
  - Why? The Internet of Things (IoT) is mostly Linux, as are most appliances
  - Many of these devices do not receive the same level of scrutiny that other servers receive, which can lead to significant security issues
- These firewalls may also be run on small/inexpensive hardware, allowing flexible segmentation options
- Some Linux appliances lack an (officially-supported) firewall but have iptables kernel support
  - This allows organizations to configure iptables to protect the device

### Linux and BSD Firewalls

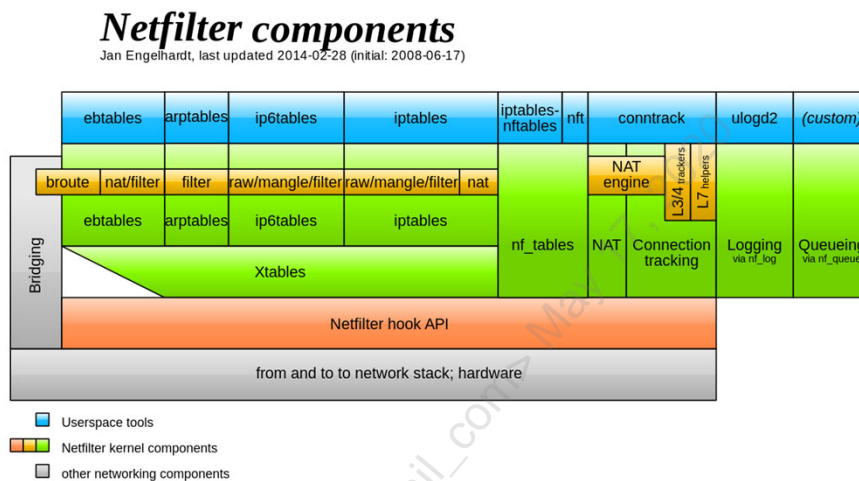
The Internet of Things (IoT) is mostly comprised of Linux. Most cell phones and tablets run Android, which is Linux. Apple macOS and iOS are BSD UNIX-based (in 'userland', they use the Mach kernel). Most of the cloud is Linux-based.

This means most organizations are Linux (or UNIX) shops, whether they know that or not. Both Linux and BSD offer kernel-based firewalls that are available in a variety of devices and products mentioned above, but not always used. Any Linux 2.4 system supports iptables by default (Linux 2.2 supports ipchains).

This means knowing how to configure iptables on Linux and PF on BSD often allows organizations to better secure equipment they have already deployed. It also allows flexible (and often inexpensive) firewall solutions for devices that may be difficult to firewall otherwise, such as legacy equipment on the WAN.

## Netfilter

- Netfilter is a packet filtering framework supported by Linux 2.4+
- Includes arptables, iptables, conntrack and much more



### Netfilter

Netfilter is a Linux 2.4 packet filtering framework. It is the successor to the ipchains (Linux 2.2) and ipfwadm (Linux 2.0) projects. Netfilter is tied directly to the Linux kernel, so it is Linux-only software. Modern firewalls are kernel-based, so they are tied directly to the underlying operating system. For example, PF is the BSD firewall (pfSense is built on top of PF).

Linux iptables is the most notable netfilter software. As the images above show: netfilter has many other components.

From the netfilter.org project:

*netfilter.org is home to the software of the packet filtering framework inside the Linux 2.4.x and later kernel series. Software commonly associated with netfilter.org is iptables.*

*Software inside this framework enables packet filtering, network address [and port] translation (NAT) and other packet mangling. It is the re-designed and heavily improved successor of the previous Linux 2.2.x ipchains and Linux 2.0.x ipfwadm systems.*

*netfilter is a set of hooks inside the Linux kernel that allows kernel modules to register callback functions with the network stack. <sup>1</sup>*

[1] <https://www.netfilter.org/>

## Linux iptables

- Linux iptables is one of the most popular firewalls
- It supports per-interface input, output, and forward filter 'chains'
  - Input: inbound traffic to an interface
  - Output: outbound traffic from an interface
  - Forward: traffic between interfaces (routed packets)
  - A chain is a list of firewall rules applied in order
- The default behavior is accept (unless explicitly denied)
  - This means a system with no configured output chain will allow all output

### Linux iptables

iptables is quite powerful and versatile. The configuration has a learning curve and can seem a bit intimidating for network personnel used to configuring firewalls via GUIs.

Iptables is comprised of tables; chains are rules applied in each table. The filter table is shown above. Iptables also supports NAT and mangle tables:

*TABLES are the major pieces of the packet processing system, and they consist of FILTER, NAT, and MANGLE. FILTER is used for the standard processing of packets, and it's the default table if none other is specified. NAT is used to rewrite the source and/or destination of packets and/or track connections. MANGLE is used to otherwise modify packets, i.e. modifying various portions of a TCP header, etc.*

CHAINS are then associated with each table. Chains are lists of rules within a table, and they are associated with "hook points" on the system, i.e. places where you can intercept traffic and take action.

[1] <https://danielmiessler.com/study/iptables/>

## Iptables Simple Ruleset

- Here is a simple iptables ruleset for a two-legged firewall

- LAN is eth0, WAN is eth1 (details are in the notes)

### # Input rules

```
iptables -A INPUT -i eth0 -p all -j ACCEPT
```

```
iptables -A INPUT -i eth1 -p tcp --dport 22 -j ACCEPT
```

```
iptables -A INPUT -j DROP
```

### # Forwarding rules

```
iptables -A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
```

```
iptables -A FORWARD -i eth0 -o eth1 -j ACCEPT
```

```
iptables -A FORWARD -j DROP
```

```
# No output rules (all output is allowed)
```

## Iptables Simple Ruleset

Here is a detailed explanation of the rules:

Accept all LAN (eth0) packets:

- **-A INPUT -i eth0 -p all -j ACCEPT**

Accept SSH from the WAN:

- **-A INPUT -i eth1 -p tcp --dport 22 -j ACCEPT**

Drop all other incoming traffic

- **-A INPUT -j DROP**

Forward established connections

- **-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT**

Forward traffic from LAN to WAN:

- **-A FORWARD -i eth0 -o eth1 -j ACCEPT**

Do not forward any other traffic

- **-A FORWARD -j DROP**

Since there are no output rules, all output is allowed (default allow)



## PF and pfSense

- PF (Packet Filter) is the BSD kernel firewall, first designed as part of OpenBSD
  - Also supported by FreeBSD, NetBSD, and many others
  - Includes advanced features such as QoS, and HA redundancy
  - PF can log in pcap format
- pfSense is a FreeBSD-based open source firewall distribution that uses PF
  - Includes an easy-to-use GUI
  - pfSense runs on many low-cost hardware appliances

### PF and pfSense

PF is BSD's firewall, first developed by Daniel Hartmeier for OpenBSD 3.0. It has since been added to many other BSD-based operating systems, including macOS.

QoS is supported through ALLTQ (Alternate Queuing). High Availability is provided by CARP (Common Address Redundancy Protocol) and pfsync.

For network engineers, one of the most intriguing features of PF is logging pcap format. PF logs are read with tcpdump, which has been extended on BSD systems for this use:

*Rather than reinvent the wheel, the PF firewall for BSD Unix systems adopted the binary data file format associated with pcap for its own logging facility, known as pflog. Thanks to this fact and the fact that a number of network traffic analysis and other activity and logging capture, filtering, and visualization tools have adopted the same format, a surprisingly large number of tools are well-suited to parsing and sorting logs for PF firewalls, including tcpdump itself.<sup>1</sup>*

[1] <https://www.techrepublic.com/blog/it-security/filtering-pf-firewall-logs/>

## pfSense Console

- The pfSense console is commercial quality, providing GUI access to all features

The screenshot shows the pfSense console interface. On the left, there's a 'Rules (Drag to Change Order)' table with columns for States, Protocol, Source, Port, Destination, Port, Gateway, Queue, Schedule, and Description. The first rule is 'Block bogon networks' with 0/3 KIB states. Below it are two rules for IPv4 TCP: one for port 443 (HTTPS) and one for port 22 (SSH). On the right, the 'System Information' panel displays details like Name (pfSense.sec530.com), System (VMware Virtual Machine), BIOS (Phoenix Technologies LTD), Version (2.4.2-RELEASE), CPU Type (Intel(R) Core(TM) i7-4980HQ), and Uptime (00 Hour 31 Minutes 28 Seconds).

### pfSense Console

pfSense is available pre-installed on many small network appliances. It's also available as a 64-bit AMD image (with ISO and memory stick installation options) and Netgate ADI (for installation on Netgate hardware).

Downloads are available from: <https://www.pfsense.org/download/>

The ISO image installs quickly in VMware (<5 minutes), allowing easy testing and 'kicking the tires' before installing on hardware. The text configuration screen (shown below) allows simple configuration, leading to the GUI console shown above.

```
FreeBSD/amd64 (pfSense.sec530.com) (ttyv0)

VMware Virtual Machine - Netgate Device ID: 861198f8ad9d657e28c4

*** Welcome to pfSense 2.4.2-RELEASE (amd64) on pfSense ***

WAN (wan)      -> em0      -> v4/DHCP4: 10.99.99.173/24
                v6/DHCP6: 2001:470:1f11:78e:20c:29ff:fe6b:879a
/64
LAN (lan)      -> em1      -> v4: 192.168.198.211/24

0) Logout (SSH only)          9) pfTop
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults  13) Update from console
5) Reboot system              14) Enable Secure Shell (sshd)
6) Halt system                 15) Restore recent configuration
7) Ping host                   16) Restart PHP-FPM
8) Shell

Enter an option: █
```

## Let's Get Small

- Organizations are often faced with legacy systems that lack vendor support
- All access (including internal) to unsupported systems should be filtered
- Options include:
  - Host-based firewall
  - VLAN ACLs
  - Router or Firewall filtering
- Another option: Velcro a tiny USB-powered firewall to the device



### Let's Get Small

Enterprise solutions such as host-based firewalls, VLAN ACLs, and enterprise router or firewall filtering are normally preferred to the small options shown above. There are cases where the typical enterprise solutions are not available, such as a Windows NT medical device running in a small clinic with an unmanaged switch and SOHO firewall (this is directly from an author's experience). In that case, 'plan A' is upgrading everything (medical device, switch, and firewall), but that may not be feasible (or fast). A perfectly valid 'plan B' is: configure a compensating control, such as small hardware firewall attached to the device.

This is often a perfectly valid solution, assuming there are no high availability concerns. Organizations often bristle at support costs (such as firewall rule updates), but these devices tend not to change: the initial firewall rules for the NT device mentioned above were: allow inbound TCP ports 443 (HTTPS) and 3389 (RDP), and deny the rest. These rules remained unchanged for the life of the device.

The 'maker revolution' has led to a wealth of inexpensive devices with amazing features.

- The device shown on the upper right is an SG-1000 microFirewall Security Appliance running pfSense. It lists for \$149.<sup>1</sup>
- The device on the lower right is a NEXX WT1520F<sup>2</sup>, capable of running OpenWRT<sup>3</sup>. It costs \$15 shipped. <https://www.cnx-software.com/2015/03/29/nexx-wt1520-router-openwrt/>

[1] <https://store.netgate.com/SG-1000.aspx>

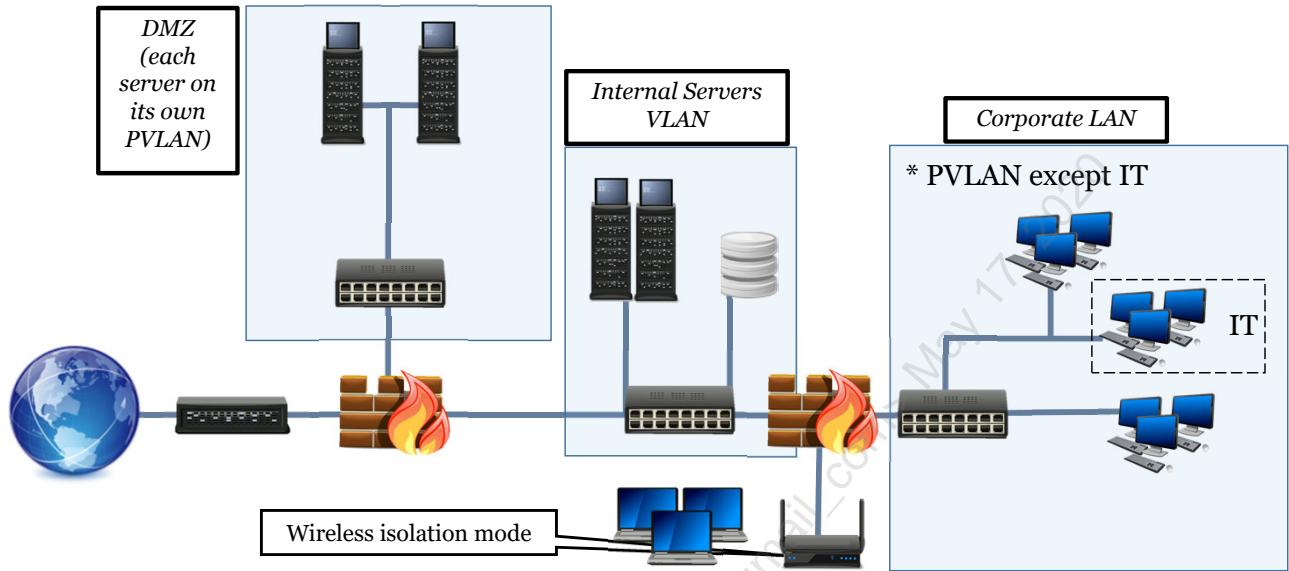
[2] <https://www.cnx-software.com/2015/03/29/nexx-wt1520-router-openwrt/>

[3] <https://wiki.openwrt.org/toh/nexx/wt1520>

[4]

[https://login.aliexpress.com/?return\\_url=http://www.aliexpress.com/snapshot/0.html?orderId=89080001654792&from=aliexpress](https://login.aliexpress.com/?return_url=http://www.aliexpress.com/snapshot/0.html?orderId=89080001654792&from=aliexpress)

### Case Study: Tyrell Corporation



This page intentionally left blank.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

## Course Roadmap

- Day 1: Defensible Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- Day 5: Zero Trust Architecture
- Day 6: Capstone: Hands-On Defend the Flag Challenge

### NETWORK SECURITY ARCHITECTURE

1. Layer 3 Attacks and Mitigation
2. Switch and Router Benchmarks
3. **EXERCISE: Auditing Router Security**
4. Securing SNMP
5. Securing NTP
6. Bogon Filtering
7. Blackholes and Darknets
8. **EXERCISE: Router SNMP Security**
9. IPv6
10. IPv6 Misconceptions
11. Securing IPv6
12. **EXERCISE: IPv6**
13. Layer 3/4 Stateful Firewall
14. **Web Proxy**
15. SMTP Proxy
16. **EXERCISE: Proxy Power**
17. 530.2 Summary

### Course Roadmap

The next section covers the importance of a Web Proxy.

## Application Layer Security

Layer 7 security refers to application layer security

- Means security based on full network protocol knowledge
- And mainstream use of standard network protocols
- **Example:** Microsoft Update uses HTTP and HTTPS

Use of application layer security differs by product and implementation

- Necessary to balance network and host protection

### Application Layer Security

Starting in book three the focus is shifting to application layer security. In regard to the OSI model, this means application-level inspection. What this means is simply that a security decision is made with the full knowledge of what an application is and does. Take for instance Microsoft Update. Microsoft Update is an application, but it operates using specific DNS domains and makes connections on top of HTTP and HTTPS.

Analysis achieves precision and making decisions at the application layer increases your overall security posture. However, this requires a lot more effort as applications have to be understood.

## Application Proxies

A proxy is a system that brokers traffic between systems

- Type of proxy is specific to the application
- Example: HTTP, SMTP, SOCKS, FTP

The goal is to funnel traffic through a proxy, so it can:

- Control the flow of data
- Analyze traffic for unauthorized or malicious use
- Cache content

Deployment and direction of proxy changes use

### Application Proxies

A proxy is a system that handles connections on behalf of another system. By design, a proxy acts as a mediator of an application. Two common examples would be a web proxy for HTTP and HTTPS traffic, a spam gateway for SMTP. Both of these handle application-level analysis of a given protocol.

The key advantage of a proxy is that it is purpose-built to provide security for a particular protocol and the applications that sit on top of that protocol. For analysis to be performed traffic must be sent through a proxy. For a proxy to analyze this traffic, the proxy requires systems to be configured to send data through the proxy.

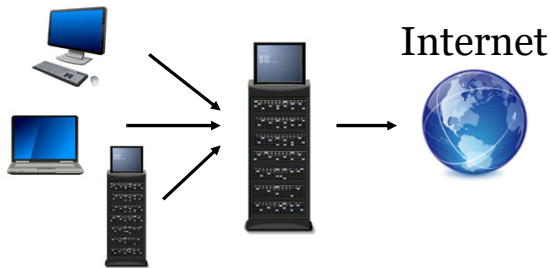
How you deploy a proxy changes its functionality. For example, forcing web traffic through a proxy out to the internet means the proxy will protect outbound internet requests. Putting a proxy in front of a web server and forcing all traffic to it will protect one or more servers.

## Proxy Types

### Forward

Systems request access through a proxy to access a resource

- Example: Web Proxy



### Reverse

Service protected by forcing connections through a proxy

- Example: Web Application Firewall / Load Balancer



### Proxy Types

A proxy protects assets by analyzing information flowing through the proxy. A workstation sending traffic through a proxy to the internet is an example of a forward proxy<sup>1</sup>. All systems sending traffic to a web application firewall that is protecting web servers is an example of a reverse proxy<sup>2</sup>. What a proxy is protecting and how it is deployed will dictate the proxy's purpose.

This module will help you focus on how a forward web proxy can significantly aid in protecting your organization.

[1] <http://www.jscape.com/blog/bid/87783/Forward-Proxy-vs-Reverse-Proxy>

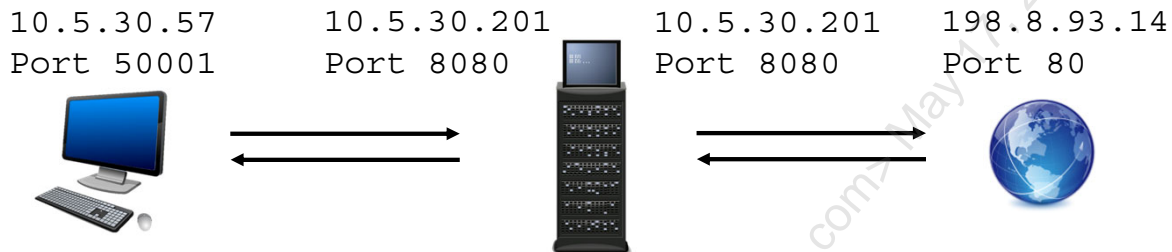
[2] Ibid.



## Web Proxy

A web proxy acts as an intermediary for web access

- Primarily used to protect internal assets
- Often underutilized and completely underestimated



Use of web proxy **forces** analysis of web traffic

### Web Proxy

A web proxy protects an organization's assets as they access the internet. Specifically, the proxy is performing analysis of HTTP and HTTPS traffic. This type of forward proxy focuses heavily on web application traffic. With so many attacks such as client-side phishing attacks, watering hole attacks, and drive-by malware, a traditional web proxy goes a long way in securing your organization.

How a web proxy works is endpoint systems connect to the web proxy, and then the web proxy makes a separate connection out to the destination server. Therefore, two separate connections are involved rather than one. The beauty of this is that the endpoint connection terminates at the web proxy and the web proxy has full visibility into the web connection request and response. In turn, this allows security decisions to be made by the proxy on behalf of the endpoint system.

A web proxy brokering a connection for an endpoint makes tracking down connections difficult. To help solve this problem a web proxy adds an HTTP header field called X-Forwarded-For or XFF. The XFF field contains the originating client's IP address. The XFF field is not mandatory, but web proxies almost universally support it. When an XFF header is in use, it is important that it is removed at the perimeter so that it does not give away information about internal systems and IP spaces to external systems.

## Web Proxy Capabilities

Inspection of web traffic includes filtering based on:

- Site category
- URLs
- File contents
- Data loss prevention
- MIME Types
- User Agents
- Global reputation
- Status codes
- Cookies
- Form values
- Protocol anomalies
- Certificates
- AV Signatures
- Sandbox analysis

### Web Proxy Capabilities

A web proxy will secure the environment by looking at various application level details about HTTP and HTTPS connections. It will take into account the website a user is requesting, the status and error codes associated with the request, and additional information such as the MIME type, user agent, and file contents involved in every web connection. If malware attempts to communicate over HTTP or HTTPS but is not properly using these protocols, the web proxy will break the connection.

Also, a web proxy will perform lookups on a site that a system is attempting to access. This lookup includes identifying the site's category such as education or gambling and may also include global reputation scores. Ultimately, a site's category or reputation is used to allow or deny the site.

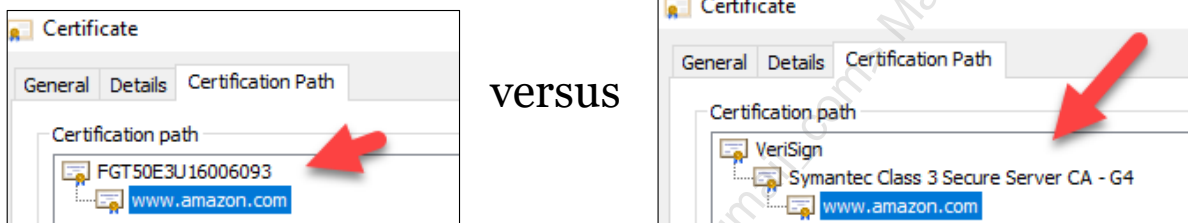
## SSL Interception

Encryption blinds a proxy by default

- Interception of traffic would cause errors and break sites

**SSL Interception** allows analysis of encrypted sites

- Requires proxy to act as a trusted certificate authority
- Proxy generates certificates per site accessed



### SSL Interception

A web proxy's benefit to security is its ability to analyze web connections. Unfortunately, the rising trend of encryption breaks a proxy's ability to analyze connections. Encryption requires an SSL handshake that verifies a server's certificate is valid. A proxy intercepting a client's request to a site using SSL would invalidate the end server's certificate trust. As a result, using a web proxy will not inspect requests to encrypted sites.

With the web quickly switching to the "all things encrypted" mentality, defenders need a way to allow inspection. The answer to this problem is implementing SSL inspection or at a minimum SSL certificate analysis. Both SSL inspection and SSL certificate analysis are broken down in a later module. A quick summary, SSL inspection uses a certificate an endpoint trusts between the endpoint and the proxy, and the proxy establishes its connection using the end web server's original certificate. SSL certificate analysis does not allow for content inspection but can make rudimentary decisions based on a certificate's information.

## Site Categories

Web proxy often associated with site category filtering

- Quick win for controlling access
- Allows sites by **category**

You should block unknown sites

- Similar to default deny firewall

Not a replacement for whitelist

- How might an adversary get through category-based filters?



### Site Categories

Web proxies are used to apply content filtering. Content filtering blocks sites using a website's reputation and category. New sites fall into an unknown category. Established sites fall into a category based on the site's content. For example, the category of education contains sites such as universities and school websites. The category of gambling would include sites about casinos or online gambling.

A key advantage of commercial proxies is the inclusion of current and accurate category classifications and reputations. Categories make allowing or blacklisting sites easy. Simply select a category and set it to either allowed or denied.

However, security administrators may be overconfident in category-based allowances. An adversary or pen tester can easily evade category-based filters. These filters are simply too broad. If an organization only accesses a hundred healthcare websites, then that organization may not want to allow the other 1 million plus healthcare sites that are also in the healthcare category.

## Bypassing Site Categories

Domains often go up for auction

- Businesses close or sites or no longer needed

Adversaries buy these and use them for phishing

- Great for buying pre-categorized sites

The screenshot shows the Auctions Domain Name Aftermarket interface. At the top, there is a search bar with the text "Enter keywords to begin search". Below the search bar, there is a "Featured Domains" section with a table of domains for sale. The table has columns for Name, Bids/Offers, Traffic, Estimated Value, Enter Bid/Offer, and Time Left. The domain "mothersrights.com" is highlighted with a red arrow. To the right of the table, there is a callout box titled "WF Rating History" which shows "Apr 14th, 2013 @ 21:21:27 PDT added as Society and Lifestyles".

✓	Name	Bids/Offers	Traffic	Estimated Value	Enter Bid/Offer	Time Left
<input type="checkbox"/>	+ mothersrights.com	0	-	-	USDS Offer \$3,000 or more	89D 12H

### Bypass Site Categories

How hard is it to bypass filters that use categories? The truth, bypassing category-based filtering is as easy as 1-2-3. The easiest way is to purchase a domain name that is up for auction. Domain registrars such as GoDaddy allow the purchase of previously owned domains. These domains come with their previous category. This slide shows the domain mothersrights.com as being for sale in 2017 and that the domain has a category of Society and Lifestyles. This site had the category established back in 2013.

An alternative way to bypass category-based filters is to establish a website with content specific to a category. Eventually, a content filtering service scans the site and establish a category. To speed things up the site can be submitted for review with a recommended categorization.

[1] <https://auctions.godaddy.com/>

[2] <https://fortiguard.com/webfilter>

## Website Whitelisting

A more secure approach is to **whitelist** all sites

- Requires gathering authorized sites and allowing them
- Then denying anything else regardless of category

This involves **more work and maintenance**

- One way to get started is to add every site accessed today to a whitelist
- May include evil inside but starts the whitelist

**Whitelisting** provides a **significant security boon**

### Website Whitelisting

A more secure way to allow or deny sites is using a whitelist. A whitelist is a list that only contains authorized items. For a web proxy, this would be a list of all authorized websites. Whitelisting works by approving websites that have a business need and then blocking all other sites.

The downside to whitelisting is it requires a lot more hands-on maintenance. In fact, if an organization implements whitelisting but cannot approve new sites within 4 hours, then they may not be candidates for whitelisting. Many whitelisting projects fail, not because of the technology but because of the lack of speed in authorizing new entries.

Successfully implementing whitelisting does not require every site is validated. Doing so would take a considerably long time and would delay the security advantages of having whitelisting applied. A quick way to implement whitelisting is to log all sites accessed over a period, such as a month or more, and whitelist all of them. Moving forward new sites would need to be approved, but everything previous would be authorized. Fast track whitelisting includes possible malware but at least gets controls in place moving forward.

## Web Proxy Alternatives

**Terms and Conditions** and **Authentication** can be required either entirely or conditionally

- Malware highly likely to fail when against both of these
- Could be required before internet access works
- Some organizations require human interaction for web access
  - Or could be applied to unknown or not whitelisted sites

### **WARNING - Responsibility Acceptance Required**

You are about to access a site that is **unknown or has not been accessed before**. This poses a security risk. By **logging** in below you are **accepting responsibility** for your actions.

### Web Proxy Alternatives

A web proxy controls web access. One method of limiting access is to require an end user to accept terms and conditions or possibly authenticating to the proxy before allowing access. Both terms and conditions and authentication can be applied holistically or only to specific website categories. Access controls also can be applied per user or group.

A key advantage of these access controls is that malware often is unaware of how to get around them. An infected machine may have malware actively running but because it does not know to accept terms and conditions or authenticate it cannot connect back home through the web proxy. These access controls are a superb and easy method to stop malware in its tracks.

Many organizations take and enforce either terms and conditions or authentication to all sites. While this is a strong security control, it greatly inconveniences end users. It also allows malware to ride a previously authenticated session. A different approach to this could be to require accepting terms and conditions or authenticating only if a site is new, unknown, or within specific categories. Now an end user may have an infected machine, but the malware would reach out to a site requiring human interaction before traffic is able to proceed. Now an organization would have prevention plus detection via the end user. The terms and conditions or authentication popup work best when it is clear to the end user that they are responsible for allowing access to whatever site they or their computer is trying to access.

## Proxy Deployment

Proxies are deployed in one of two modes

- **Transparent** - Traffic goes through proxy regardless of endpoint configuration
- **Explicit** - Endpoints must be configured to use the proxy

Many proxies support both deployment types

- Transparent is easier to deploy
- But explicit has some significant security advantages

### Proxy Deployment

There are two main categories of proxies. They are forward and reverse proxies. A forward proxy protects systems that are connecting out to another system. An example of this is a web proxy. A reverse proxy is an opposite in that it protects the asset systems are connecting to. An example of this is a web application firewall. Both a forward proxy or reverse proxy can be deployed in one of two modes: transparent or explicit.

An explicit web proxy requires endpoints to be configured to use the proxy. A transparent proxy sits inline on the internet and does not require endpoints to know it exists. As such, transparent proxies are easy to deploy and maintain.

An explicit proxy provides significant advantages simply because it requires endpoints to know it exists and to require configuration to use.



## Explicit Proxy Advantages

Malware is often not proxy aware

- Under an **explicit proxy**, this means **no** internet access
- With a transparent proxy, malware has internet access
- This alone justifies having an explicit proxy

Only works if outbound web access is denied except for the web proxy

- Standard firewall rules for the win
- You will have to make exceptions for some assets

### Explicit Proxy Advantages

An explicit proxy works best when it is the only method to access websites found on the internet. All attempts to access websites directly need to be blocked by a firewall. Requiring web access to funnel through a web proxy alone is enough to stop a large percentage of malware. An end user may click a malicious link and become infected, but if the malware is not a proxy-aware, then the malware will be unable to access the internet. Proxy-aware means a program, whether benign or malicious, is programmed to inherit proxy settings from the operating system or manually specifies a proxy.

In truth, not all assets support the use of a web proxy. Assets not supporting the use of a web proxy will need to either use a transparent web proxy or have firewall rule exceptions. Taking the time to handle exceptions is well worth the advantage of having an explicit proxy.

## Authenticated vs. Unauthenticated Proxy

Malware commonly is not proxy aware

- But malware occasionally is aware



Explicit proxy plus authentication is the most secure

- Even proxy aware malware struggles with credentials
  - Malware would first have to steal credentials
  - Then use them in a proxy-aware fashion for explicit mode

Unauthenticated proxies imply trust without verification

- Connect to a proxy then device equals trusted = **BAD**

### Authenticated vs Unauthenticated Proxy

The use of an authentication proxy means that clients attempting to use a proxy must prove themselves before use is authorized. The alternative is unauthenticated proxy access. Unauthenticated proxy access means that any device that can make a network connection to a proxy is authorized. Thus, every client is trusted by default. While this sounds crazy, it is the default rollout strategy of many web proxy deployments.

Instead, web proxies should require authentication and be deployed in explicit mode. Combining these two achieves strong malware protection. A majority of malware is not proxy aware. Those pieces of malware that are proxy aware often fail to connect through a web proxy that requires authentication. For malware to successfully do this would require automated intelligence to steal credentials and then use them as a proxy prior to attempting any outbound internet access.

[1] <https://itstillworks.com/proxy-authentication-types-3269.html>

## Control of Flow

An explicit proxy also forces internet flow

- Must go through the funnel to access the web

Will **prevent** and **detect** internet access from:

- Internet of things devices
- Personal devices
- Unauthorized devices
- Malware
- Misconfigured assets



### Control of Flow

Another key advantage of an explicit web proxy revolves around the fact that authorized access must go through it. Direct access from an endpoint to the internet should not happen by design with an explicit proxy. Requiring access through the proxy will prevent and detect connections from unauthorized, misconfigured, or malicious endpoints. Identifying and dealing with all non-proxy aware devices is an advantage for the conscious security defender.

## Proxy Placement

Ideally, everything would go through an explicit proxy

- What about devices that do not support proxies?
- What about devices that enter and leave the network?

**Segmentation** should be considered for "dumb" devices

- And possibly use a transparent proxy to limit access

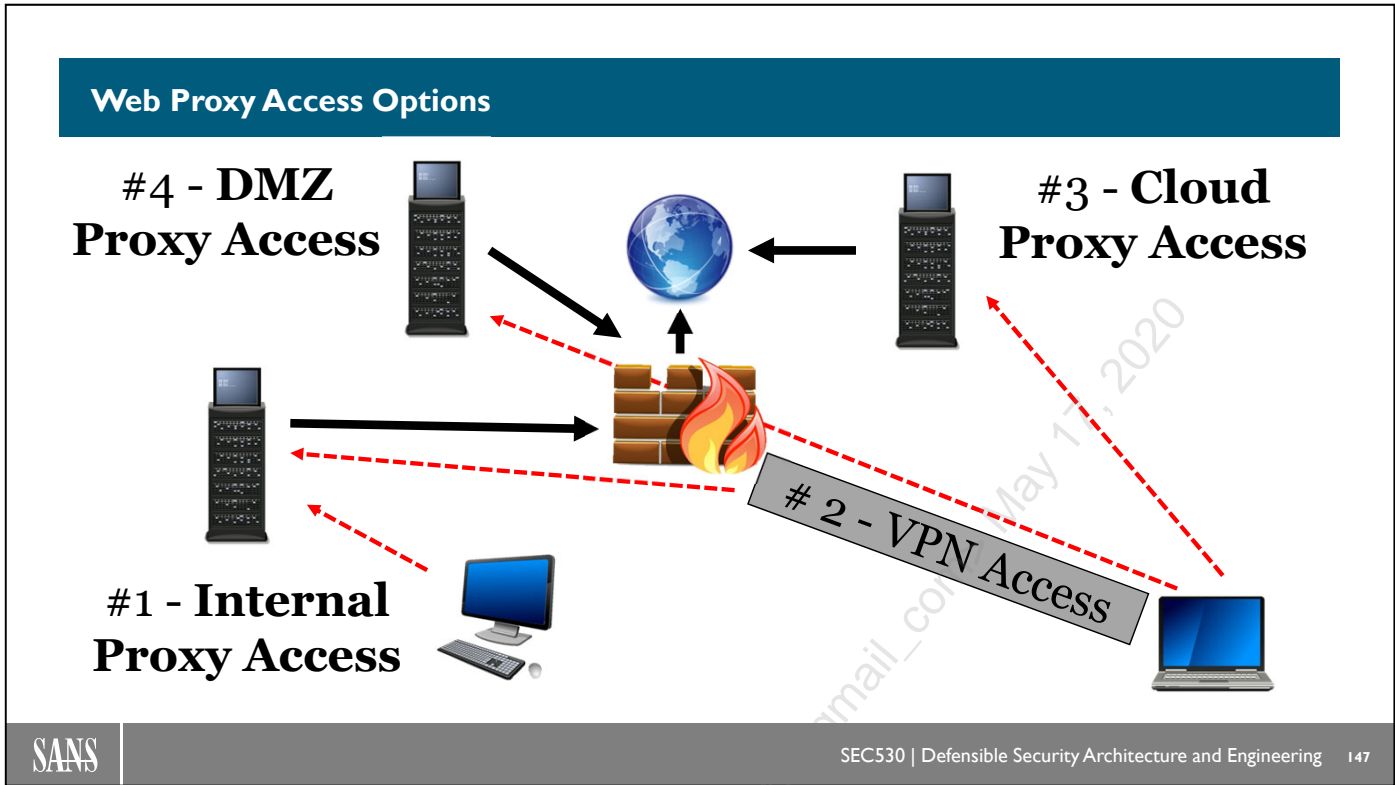
Systems supporting proxy need access to the proxy

- Through direct access via internal or VPN access
- Or via proxy in the cloud or internet facing DMZ system

### Proxy Placement

The ideal goal is to use an explicit web proxy. The problem is some assets cannot specify a proxy and other assets are constantly coming into and out of an organizations network. Devices that cannot specify a proxy require either the use of a transparent web proxy and/or placement into a segmented network. A segmented network makes controlling and protecting assets that have limited controls an easier task.

Devices that enter and leave a network routinely are more challenging. Devices such as laptops can use an explicit web proxy. However, mobile devices require a means to access the proxy. This access can be through the use of an always-on VPN tunnel, a cloud-hosted proxy service, or a DMZ internet facing web proxy.



**Web Proxy Access Options**

This diagram represents four methods of accessing an explicit web proxy. The solid arrow lines reflect how traffic from the web proxy reaches the internet. In most cases, a web proxy is behind a corporate firewall and will route out to the internet through the perimeter firewall. In the case of a cloud proxy, web traffic does not pass through a corporate firewall. The arrow lines with dashes reflect how clients connect to a proxy. The traditional approach is to connect to the proxy over the internal network. Internal network access works for desktops and servers but does not work for mobile devices.

Mobile devices use an explicit proxy in one of three ways. One way is to have an always-on VPN connection. If a laptop uses an always-on VPN, then it can point to the proxy server directly using the proxy's internal IP address. Another way a mobile device can use an explicit proxy is by accessing it directly over the internet. Direct access to a web proxy requires either a cloud-hosted solution or port forwarding through a corporate firewall.

## Squid

You do not need money to benefit from a web proxy

- Squid provides an open source solution

Supports explicit and transparent configurations

- Can use **ClamAV** for antivirus checks
- Or can integrate with **ICAP** servers
- Supports web caching and acceleration
- Supports SSL interception

Commercial solutions are more robust



### Squid

Squid<sup>1</sup> is an open source web proxy that supports both transparent and explicit mode deployment options. Squid runs on commodity hardware or virtualization platforms and includes the ability to perform antivirus checks, web caching, optional SSL interception, and the ability to make calls to an Internet Content Adaptation Protocol (ICAP) server.

Commercial web proxies are much more mature and feature-rich in comparison to Squid. However, Squid is free and is better than not having a web proxy. Antivirus checks in Squid use ClamAV<sup>2</sup>. ClamAV is also not as mature and feature-rich, but it is free and easy to implement.

[1] <http://www.squid-cache.org/>

[2] <http://squidclamav.darold.net/>

## Internet Content Adaptation Protocol (ICAP)

**ICAP** is used to extend the capabilities of a proxy

- Offloads tasks to another system for processing
- Typically includes antivirus and malware analysis
- Often includes multiple antivirus engines

Custom integrations can be built using the ICAP service

- Recommend sticking with commercial solutions
- ICAP requires advanced knowledge and programming

Squid uses ICAP to establish filtering with **SquidGuard**

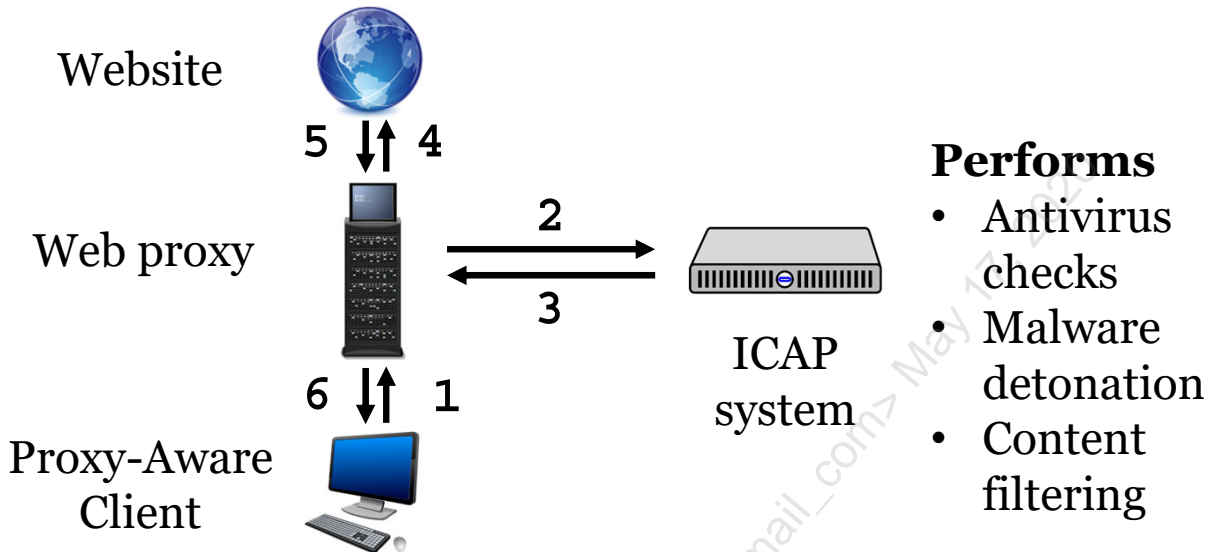
### Internet Content Adaptation Protocol (ICAP)

An interesting capability of modern web proxies is their ability to integrate with other solutions with the Internet Content Adaptation Protocol, or ICAP for short. ICAP allows additional security checks such as multiple antivirus engine scans, malware detonation, or custom analysis to run on a connection or file from a connection. ICAP analysis means more security checks, and it also means offloading to different hardware.

Vendor solutions use ICAP to integrate multiple products. ICAP integration is not as common outside of commercial products because ICAP is difficult to customize and integrate. Squid supports ICAP and easily integrates with SquidGuard. SquidGuard allows implementing access controls based on lists of authorized or unauthorized sites.

[1] <http://www.squidguard.org/>

## ICAP Diagram



### ICAP Diagram

This slide demonstrates the flow of a connection when an ICAP service is in use. In this diagram, the ICAP system helps offload certain tasks such as antivirus scans, malware detonation of files, and content filtering. ICAP is an optional service that helps to offload or extend the capabilities of a proxy.

[1] [https://farm3.staticflickr.com/2831/10420027034\\_6b94402287.jpg](https://farm3.staticflickr.com/2831/10420027034_6b94402287.jpg)



## Web Proxy Review

If you do not have a web proxy... get one!

- Provides central control and visibility of web traffic
- Stops stage two malware downloads from succeeding
- Can give users a visual warning about new sites

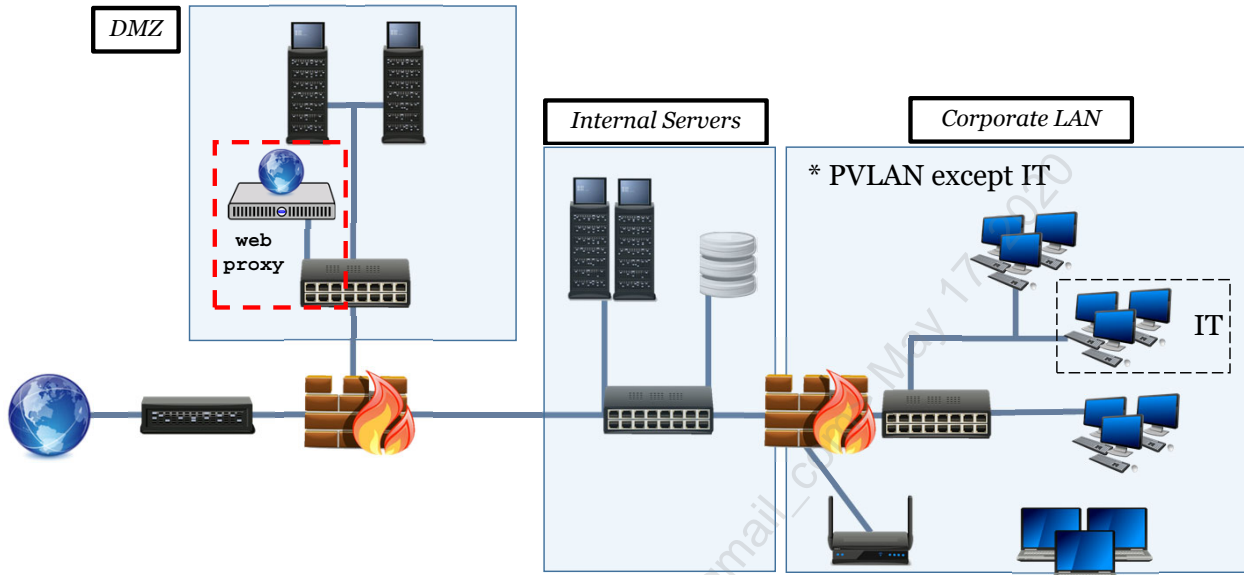
Consider the following web proxy strategies:

- Full explicit proxy over transparent proxy
- Full whitelisting vs unknown category sites
- Applying authentication or terms and conditions

### Web Proxy Review

Organizations that do not require a web proxy to access the internet need to consider adding it to their overall security architecture. The advantages of a web proxy especially in combating malware or data exfiltration are vast. Under ideal circumstances deploy a web proxy in an explicit mode with full whitelisting. Also, implement either authentication or terms and conditions for unknown sites to be accessed or require IT to authorize new sites.

## Case Study: Tyrrell Corporation



### Case Study: Tyrrell Corporation

This diagram shows the addition of an explicit web proxy to the Tyrrell Corporation architecture. In this example, web requests going outbound to the internet should come from the explicit web proxy.

## Course Roadmap

- Day 1: Defensible Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- Day 5: Zero Trust Architecture
- Day 6: Capstone: Hands-On Defend the Flag Challenge

### NETWORK SECURITY ARCHITECTURE

1. Layer 3 Attacks and Mitigation
2. Switch and Router Benchmarks
3. **EXERCISE: Auditing Router Security**
4. Securing SNMP
5. Securing NTP
6. Bogon Filtering
7. Blackholes and Darknets
8. **EXERCISE: Router SNMP Security**
9. IPv6
10. IPv6 Misconceptions
11. Securing IPv6
12. **EXERCISE: IPv6**
13. Layer 3/4 Stateful Firewall
14. Web Proxy
15. **SMTP Proxy**
16. **EXERCISE: Proxy Power**
17. 530.2 Summary

### Course Roadmap

The next section covers the use of an SMTP proxy.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

## SMTP Proxy

Web and email traffic are common ways to enter a network

- Due to client-side phishing attacks
- End users routinely use web and email, so the attack vector is massive

An SMTP proxy is an effective means to control email

- This is more commonly called a spam appliance
- Selling point is handling spam, but it does much more

**Spam = Nuisance**      **Malware = Compromise**

### SMTP Proxy

Spam is knocking at your gates demanding to be let in. Put frankly; spam is a nuisance. Spam appliances, which are SMTP proxies, combat spam by identifying key SMTP characteristics to drop an email. Dropping spam is important because while it is always annoying, it sometimes contains malware.

Also, adversaries attack organizations most commonly through websites and email. A phishing attack is difficult to prevent, but an SMTP proxy specializes in handling emails. Design a spam appliance to handle more than spam. Treat the appliance as what it is, an SMTP proxy.

## SMTP Prevention and Detection

Balancing security controls is difficult

- Focus on spam is pure prevention
- More serious emails need prevention and detection

**SMTP proxy capabilities** include but not limited to:

- Bayesian Analysis
- Per email encryption
- Handling phishing domains
- Modifying and auto-routing emails
- Sender authentication
- Rate limiting
- Sender blocking

### SMTP Prevention and Detection

SMTP proxies focus heavily on prevention. Prevention is important because a majority of email is spam. However, a good SMTP proxy prevents and identifies a targeted attack. This module focuses on key features that SMTP proxies deploy. The features include statistical analysis through Bayesian Analysis, sender verification to prevent spoofing, special ways to identify targeted phishing domains, and key ways to modify emails to empower your end users.

## Bayesian Analysis

Key prevention technique is learning normal from email

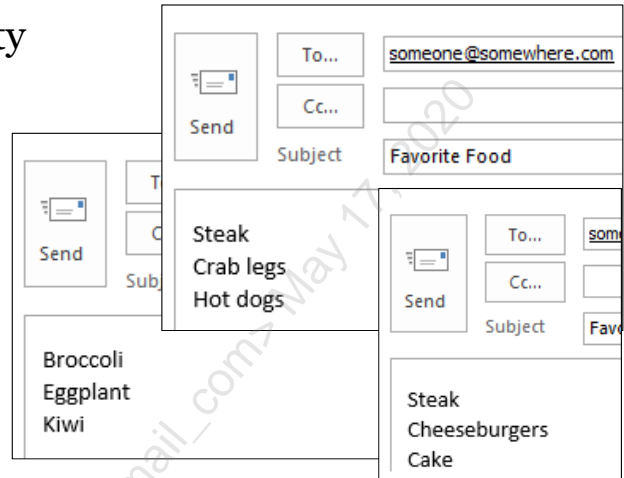
- Works with statistical probability
- Not fancy machine learning

Works by checking words, email headers, and metadata

- Ideally calculates per user
- Changes over time

Spam/attackers can bypass this

- Trial and error + read receipts



### Bayesian Analysis

Machine learning is either unsupervised learning, which deals with classifying datasets without any prior knowledge or supervised learning, which involves feeding known good and known bad data to learn how to classify new data. Machine learning is fantastic, but it is not the most appropriate method, and vendors are overselling the idea. Bayesian Analysis is an old yet highly effective way to identify spam or malicious email. Bayesian Analysis operates by performing an ongoing statistical analysis which results in a probability score that something is going to happen.

This level of statistical analysis sounds mysterious and confusing, but multiple online resources are available to break it down<sup>12</sup>. Assume that two colleagues email each other every day. On Monday, they go back and forth on going to a steakhouse or a Mexican restaurant. On Tuesday, they discuss a cheeseburger and seafood restaurant. Then Wednesday, they decide between Mexican and Korean. Each day multiple emails contain different types of restaurants. Now, what is the probability that one of them will suggest going to a vegan restaurant?

Practically zero. Now, would the probability remain close to zero if one of them started to discuss the need for eating healthy and going on a diet? No. It may not jump significantly but because of the additional "keywords," it is more likely that an email may suggest a vegan restaurant. This example is effectively how Bayesian Analysis operates.

Because analysis considers various keywords, header information, and any other metadata over time the scoring system adapts over time. The adaptation is exactly what is necessary to keep up with evolving spam. However, adapting to changes also means given enough persistence an adversary will get an email into your environment. In fact, an adversary can continue to send emails to individuals and have a way to verify when an email works. Verification is simple using a built-in email capability: read receipts.

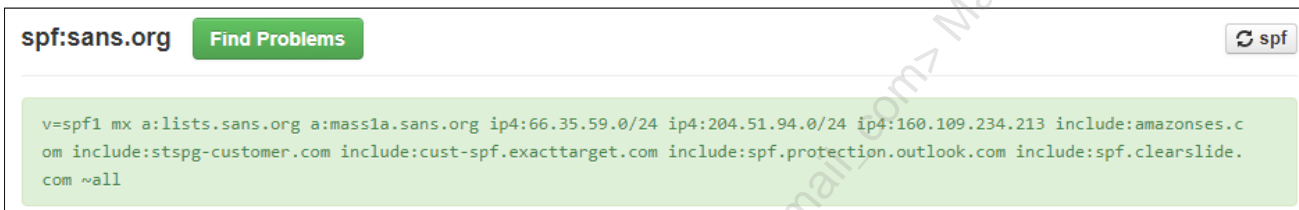
[1] <https://www.analyticsvidhya.com/blog/2016/06/bayesian-statistics-beginners-simple-english/>

[2] <https://www.lifewire.com/bayesian-spam-filtering-1164096>

## Sender Policy Framework (SPF)

DNS record validates email sent from an authorized source

- Based on authorized IP addresses
- Based on DNS domain information (A record, MX record)
- Can specify no email comes from a specific sub-domain



```
v=spf1 mx a:lists.sans.org a:mass1a.sans.org ip4:66.35.59.0/24 ip4:204.51.94.0/24 ip4:160.109.234.213 include:amazonses.com include:stspg-customer.com include:cust-spf.exacttarget.com include:spf.protection.outlook.com include:spf.clearslide.com ~all
```

### Sender Policy Framework (SPF)

Organizations own specific domains, and they are responsible for sending emails to those domains. Spammers routinely craft emails from domains they do not own. An SMTP proxy can verify emails come from an authorized source thus blocking spoofing attempts. A better way to prevent spoofed domain emails is to implement sender verification. Sender verification is the process by which a domain owner establishes rules dictating who or what can send emails from a given domain.

In order to get around spam filters, spammers routinely attempt to craft emails from domains they do not own. To catch this, an SMTP proxy can use sender verification to verify emails come from an authorized email source, thus blocking spoofing attempts. Sender verification is the process by which a domain owner establishes rules dictating who or what can send emails from a given domain.

The Sender Policy Framework is a standard to specify which systems can send emails from a certain domain. Sender Policy Framework requires two things to work. First, Sender Policy Framework requires a DNS TXT record that specifies what IP addresses or domains can send emails from a domain. Second, Sender Policy Framework requires an SMTP proxy or email system that enforces SPF checks. Many SMTP proxies enable Sender Policy Framework by default. Sender Policy Framework then prevents emails that originate outside of the systems specified in the DNS record for the domain.

+ = Pass or accept all messages

- = Hard fail which means drop the message

~ = Soft fail which means accept and tag mail

? = Neutral which means neither pass or fails (likely mail will be accepted)

all = Designates that SPF record is all inclusive and no other servers are to be allowed

a = Authorized A record address

ipv4 = Authorized IPv4 address

ipv6 = Authorized IPv6 address

mx = Authorized MX address

include = Allows 3rd party set to send mail on behalf of the domain

v = version

For a list of common Sender Policy Framework, mistakes see the OpenSPF site<sup>2</sup>.

[1] [https://blog.returnpath.com/how-to-explain-Sender Policy Framework-in-plain-english/](https://blog.returnpath.com/how-to-explain-Sender-Policy-Framework-in-plain-english/)

[2] [http://www.openSender Policy Framework.org/FAQ/Common\\_mistakes](http://www.openSenderPolicyFramework.org/FAQ/Common_mistakes)

Licensed To: Martin Brown <hermespaul56@gmail\_com> May 17, 2020



## DomainKeys Identified Mail (DKIM)

Uses digital signatures to validate email

- Means asymmetric keys (private + public) and hashing

Keys are created for each **selector** (may just need one)

- Private key goes to email system(s)
- Public key saved in DNS TXT record under **\_domainkey.domain.com**

DKIM-Signature:

```
v=1;
a=rsa-sha256;
c=relaxed/relaxed;
d=sans.org; s=selector1;
h=From:Date:Subject:Message-ID:Content-Type:MIME-Version;
bh=nRxPejqDeN2l/NoBd57Ct8luolQkhamq8m26Ts86il9E=;
b=M9Q7HknXuCN0SMYiajVJACwpaqjCoSJHc4AyA2GF+J88dpUEj5tHYHmVRCfOqR3lBhehQIEYAZWS6w5U...
```

### DomainKeys Identified Mail (DKIM)

Another approach to preventing email spoofing of a given domain is the use of DKIM. DKIM uses digital signatures to send an email that guarantees it originates from the owner of a domain. This guarantee is achieved by hashing specific fields of an email such as the from address or message body and encrypting that with a private key. The public key is in a DNS record under the \_domainkey TXT record for the domain. An email system or SMTP proxy with DKIM will hash the specified fields or message body and then download the public key of the originating domain to decrypts the encrypted hash. If the decrypted hash matches the hash calculated the email came from the domain owner as only the private key could have digitally signed or encrypted the pre-calculated hash.

Unfortunately, asymmetric encryption and digital signatures are complicated, so DKIM is not as common as SPF. Also, not all SMTP proxies fully support DKIM so plan out future purchases for DKIM support. Keep in mind that the use of asymmetric encryption is a significant step towards proving domain ownership. The breakdown of some of the key DKIM fields is below.

a = algorithm

b = signature data in base64 (whitespace is ignored)

bh = Hash of the canonicalized body (whitespace is ignored)

c = Message canonicalization (specifies how the message was canonicalized)

d = Domain of the signing entity

s = Specifies the selector being used

v = version

Setting up DKIM requires generating a public and private key. Setup also requires the creation of a selector. A selector is simply a name identifying what public/private key pair is in use. The selector is necessary as DKIM supports multiple key pairs so that an organization can handle geographical or large-scale email operations or for troubleshooting and ease of identifying what system sent an email.

[1] <https://blog.returnpath.com/how-to-explain-dkim-in-plain-english-2/>

[2] <http://dkim.org/specs/rfc4871-dkimbase.html>

Licensed To: Martin Brown <hermespaul56@gmail\_com> May 17, 2020

## Domain-Based Message Authentication, Reporting, and Compliance (DMARC)

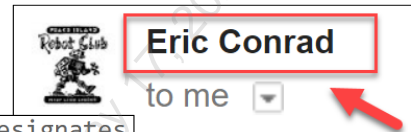
DMARC verifies domain authentication via SPF or DKIM

- Can use SPF/DKIM to force alignment of visible From
- DMARC policy dictates actions and protection level
- **Policy** – Monitor, Quarantine, Reject
- **Alignment** – Strict, Relaxed

```

spf=pass (google.com: domain of econrad@gmail.com designates
209.85.220.41 as permitted sender) smtp.mailfrom=econrad@gmail.com;
dmarc=pass (p=NONE sp=NONE dis=NONE) header.from=gmail.com
Return-Path: <econrad@gmail.com>
Received: from mail-sor-f41.google.com (mail-sor-f41.google.com.
[209.85.220.41])

```



"v=DMARC1;p=reject;pct=100;rua=mailto:dmarc@sec530.com"

### Domain-Based Message Authentication, Reporting, and Compliance (DMARC)

DMARC is different from SPF and DKIM. Both SPF and DKIM verify an email came from the owner of a domain. Neither SPF or DKIM verify that the displayed header "from" address is from the verified domain. For example, an email could have a "from" address of sec530.com, yet the "from" address displayed could be forged such as info@sec530rocks.com. DMARC is the extra step that prevents mismatched "from" and "replies to" addresses. DMARC requires another DNS TXT record to establish the policy and alignment.

A DMARC policy dictates what happens if an email fails. For an email to fail, one of two conditions must happen. If both SPF and DKIM fail then, DMARC fails a message. If either SPF or DKIM pass but the reply to the domain does not pass the DMARC alignment mode, then the email will fail. A huge benefit of DMARC is that a failed message automatically gets logged, quarantined, or rejected depending on the policy. The policy options mean DMARC is both a preventative and detective control. The alignment mode defaults to relaxed if not set. Relaxed accepts jhenderson@a.sec530.com so long as the DMARC entry exists for sec530.com. Strict mode will not. Under strict mode, you have to create a DMARC DNS TXT record per subdomain. You can find the common DMARC settings below.

adkim = Alignment mode for DKIM (either r for restricted or s for strict)

aspf = Alignment mode for SPF (either r for restricted or s for strict)

p = Policy action (either monitor, quarantine, or reject)

pct = Percentage of messages that are filtered

rua = Report address for aggregate reports

ruf = Report address for forensic reports

sp = Policy action for subdomains

v = version

[1] <https://dmarc.org/overview/>

[2] <http://www.zytrax.com/books/dns/ch9/dmarc.html>

## Sender Authentication

Not all email systems support SPF, DKIM, or DMARC

- Even when supported enforcement varies

Protection only applies to company-owned domains

- No protection against cousin domain attacks

**sec530.com** is a cousin domain of **sec530.com**

- Blocking example.com from outside is great
- Verifying all email sources of example.com is too

But how does one protect against **sec530.com**?

### Sender Authentication

Destroying spoofed emails before they cause damage is why sender authentication exists. However, sender authentication only helps with emails spoofing domains an organization owns. What about cousin domains? A cousin domain is a domain that looks similar to a domain. An example of a cousin domain to sec530.com is the domain sec530.com. The latter domain looks like sec530.com except the zero is replaced by a capital o. The use of a cousin domain flies past sender authenticated domains and also attacks an organization's layer 8: the humans.

The human mind is fast and intelligent. When reading something your brain is jumping ahead and filling in gaps according to what it expects. This fill in the blank functionality helps cousin domains be effective. Many individuals, both security professionals and otherwise, cannot see the illusion and deception a cousin domain presents. And yet cousin domains are easy to handle.

## dnstwist<sup>1</sup>

### SMTP proxy can protect against cousin domains

- You can add all possible domains into a proxy
- Should configure to **block** or **quarantine and alert**
- Requires identifying all possible domain permutations

### dnstwist<sup>1</sup> calculates permutations against a given domain

- Also checks to see if any domains have been registered
- And provides additional information about the domain

### Use dnstwist with scripting to handle deal with evil cousins

#### dnstwist<sup>1</sup>

The domain sec530.com has a cousin domain of sec53O.com. The sec53O.com domain is easy to handle. Simply block the domain or better yet accept emails for the domain but auto route them to a special mailbox. Any emails to a cousin domain represent a targeted phishing attack. By allowing the email in the adversary is unaware that you are on to them. This level of prevention plus detection acts as an early warning indicator.

However, sec53O.com is not the only cousin domain for sec530.com. An organization needs to identify all possible cousin domain permutations. Fortunately, dnstwist exists<sup>1</sup>. dnstwist calculates all the possible permutations of a given domain and then outputs to the screen or a CSV file. On top of this, dnstwist verifies a cousin domain is registered. A great way to deal with cousin domains is to use dnstwist to identify all possible permutations and then set up an SMTP proxy to auto quarantine and alert on emails from the cousin domains. Also, script dnstwist to run once a day or once a week to identify if a cousin domain is registered for early phishing detection.

[1] <https://github.com/elceef/dnstwist>

## Intentional Email Modification

SMTP proxies and email systems can add to a message

- Disclaimer messages
- Custom headers or footer banners
  - "This message came from an external source"
  - "This message may be a phishing email acting as an executive"

Requires setting up rules to do **X** when **Y** is true

- If display name matches executive add **phishing message**
- If external source add **external source message**

### Intentional Email Modification

Often automation handles prevention and detection. However, a different approach to applying automation is the use of automation to help people make informed decisions. An example of this is conditionally modifying email messages. A simple example is applying a message that states "This message came from an external source. Be cautious of phishing.". Apply this message by looking for external email sources.

Another, more targeted example is using an organization's staff hierarchy to look for phishing. The hierarchy is read to find key display names such as those belonging to a CEO or CFO. Then if an external email comes into the SMTP proxy, the email is modified notifying the receiving individual to be careful. A warning is displayed such as "This is an external email and it looks like it may be acting as an internal employee. Please verify this email is not a phishing email. If you believe this to be a phishing email notify security."

## Combating Open-Source Intelligence

A good architect protects an organization

- A great architecture raises the bar against attackers

Try creating email accounts for users that do not exist

- Possibly register these accounts for online services

Any attempt to access or email these accounts = BAD

- Scripting can automatically ban email senders
- Or at a minimum spam appliance can generate an alert

**jsans@organization.com**



### Combating Open-Source Intelligence

What if there was a way with certainty you could identify foul play? Adversaries and automated spam systems routinely scavenge the Internet to identify candidates to send emails. This scavenging opens up an opportunity for detection. As a defender, you can and should create email addresses never find use. You can also register these email addresses to various sites to give them more of an online presence.

Now, if a phishing or spam campaign emails one of these email addresses, you know with certainty that something is occurring that should not take place. For a spam campaign, a nuisance is found. For a phishing campaign, early detection is achieved. An automation script can take these further by automatically reacting to the situation.

## Rate-Limiting

Too much of something is a bad thing

- Rate limiting protects by slowing down mass email
- Both inbound and outbound

Rate limiting thresholds can be adjusted per sender

- SMTP proxy has a default threshold based on solution
- The mass email usually from business partners
- Inbound rate limiting could be early detection or noise
- Outbound rate limiting usually is a compromise

### Rate-Limiting

Email servers and SMTP proxies limit how many emails they send to or from a source within a given time frame. This rate-limiting capability is not a security feature. However, you can apply rate-limiting as a detection mechanism.

If an organization partners with another business or marketing firm they will often disable rate-limiting to or from the partner domains or increase the threshold. What this means is rate-limiting, when set up properly, is not a regular occurrence and should be investigated. To investigate rate-limiting events make sure your SMTP proxy is generating alerts, reports, or logs any time rate-limiting occurs. A more advanced implementation is using logs to monitor which sources are emailing a large group of employees within a longer period. By expanding the range, you will find phishing or spam campaigns that are sending emails in batches to avoid hitting a rate-limit.



## SMTP Proxy Review

An SMTP proxy is a mature technology to deal with spam

- But it can do so much more if you tune it

Consider configuring:

- Sender authentication
- Message header/footers on external or phishing emails
- Setup cousin domains for quarantine and detection
- Consider using and looking at rate limiting differently

### SMTP Proxy Review

SMTP proxies and spam appliances are old technologies. Old does not mean ineffective. An SMTP proxy is a mature, highly effective prevention and detection system for securing mail. An SMTP proxy should verify sender information, apply custom headers or footers, and identify ways to prevent and detect both spam and phishing attempts.

## Course Roadmap

- Day 1: Defensible Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- Day 5: Zero Trust Architecture
- Day 6: Capstone: Hands-On Defend the Flag Challenge

### NETWORK SECURITY ARCHITECTURE

1. Layer 3 Attacks and Mitigation
2. Switch and Router Benchmarks
3. **EXERCISE: Auditing Router Security**
4. Securing SNMP
5. Securing NTP
6. Bogon Filtering
7. Blackholes and Darknets
8. **EXERCISE: Router SNMP Security**
9. IPv6
10. IPv6 Misconceptions
11. Securing IPv6
12. **EXERCISE: IPv6**
13. Layer 3/4 Stateful Firewall
14. Web Proxy
15. SMTP Proxy
16. **EXERCISE: Proxy Power**
17. 530.2 Summary

### Course Roadmap

We will now have a lab on implementing and tuning a proxy.

## Exercise 2.4: Proxy Power

- Exercise 2.4 is in the digital wiki found in your student VM (recommended)
- Alternatively, you may use your Workbook

This page intentionally left blank.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

## Course Roadmap

- Day 1: Defensible Security Architecture
- **Day 2: Network Security Architecture**
- Day 3: Network-Centric Application Security Architecture
- Day 4: Data-Centric Application Security Architecture
- Day 5: Zero Trust Architecture
- Day 6: Capstone: Hands-On Defend the Flag Challenge

### NETWORK SECURITY ARCHITECTURE

1. Layer 3 Attacks and Mitigation
2. Switch and Router Benchmarks
3. EXERCISE: Auditing Router Security
4. Securing SNMP
5. Securing NTP
6. Bogon Filtering
7. Blackholes and Darknets
8. EXERCISE: Router SNMP Security
9. IPv6
10. IPv6 Misconceptions
11. Securing IPv6
12. EXERCISE: IPv6
13. Layer 3/4 Stateful Firewall
14. Web Proxy
15. SMTP Proxy
16. EXERCISE: Proxy Power
17. 530.2 Summary

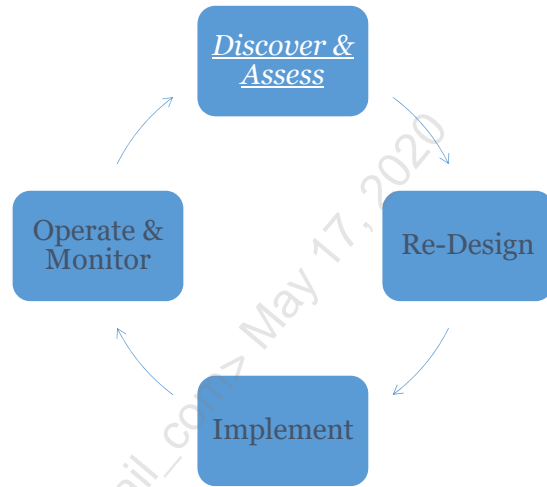
### Course Roadmap

That wraps up 530.2!

Discover & Assess on 530.2



- IPv6 traffic on your network
- Common basic issues to layer 3/4 devices:
  - Secure administration
  - Services offered
  - Vulnerabilities
  - ACLs
  - Banners
  - Logging
  - Authentication, Authorization and Accounting



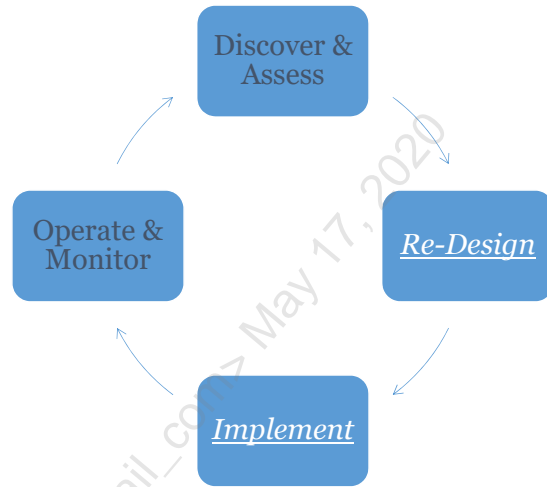
This page intentionally left blank.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

## Re-Design & Implement on 530.2



- Harden layer 3/4 according to best practices
- Disable unnecessary services and harden existing ones
- Set up ACLs to filter inbound / outbound traffic
- Configure privilege levels and role-based CLI access
- Know IPv6 capabilities of your prevention and detection tools and blind spots. Mitigate those accordingly
- Provide routing update authentication
- Enable centralized logging

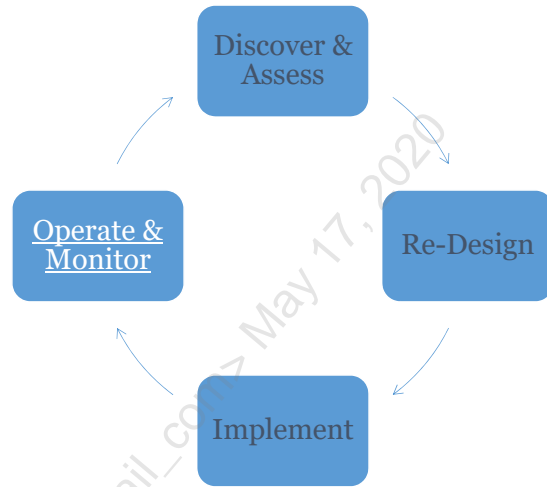


This page intentionally left blank.

## Operate & Monitor on 530.2



- Manage routers over secure protocols (SSHv2)
- Monitor layer 3/4 attacks
- Monitor IPv6 traffic and alert on potential IPv6 tunneling
- Monitor changes to layer 3 /4 devices via built-in mechanisms or differential configuration snapshots



This page intentionally left blank.

Licensed To: Martin Brown <hermespaul56@gmail.com> May 17, 2020

## 530.2 Summary

- That wraps up 530.2
- We will continue with network centric security in 530.3
- See you then!

That wraps up 530.2.

We will continue our journey with network centric security during 530.3, covering the following topics:

- Next-generation firewalls
- SMTP (Simple Mail Transfer Protocol) proxies
- Network Security Monitoring (NSM)
- Malware Detonation Devices
- Network Encryption
- Jump Boxes
- DDoS protection
- And more...

See you then!