SCUOLA
NORMALE
SUPERIORE

SELEZIONE PUBBLICA, PER TITOLI ED ESAMI, PER LA COPERTURA DI N. 1 POSTO DI CATEGORIA D – POSIZIONE ECONOMICA D1 – AREA TECNICA, TECNICO-SCIENTIFICA ED ELABORAZIONE DATI, A TEMPO INDETERMINATO E PIENO, PRESSO IL SERVIZIO INFRASTRUTTURE INFORMATICHE.

Criteri di valutazione delle prove

La prova consisterà in tre quesiti a risposta aperta, un quesito a risposta multipla con necessità di un commento da parte del candidato, un quesito a risposta multipla non esclusiva (una o più risposte corrette) e un quesito a risposta multipla esclusiva (una sola risposta esatta)

Per la valutazione la Commissione terrà conto dei seguenti criteri:
- correttezza
- capacità di argomentazione (ove previsto)
- ampiezza dell'informazione (ove previsto)
- chiarezza di esposizione (ove previsto)
- capacità di sintesi (ove previsto)

Nello specifico i punteggi saranno così attribuiti:
- massimo 6 punti alle domande n.1), 2) e 3) di ciascuna prova;
- massimo 5 punti alla domanda n.4) di ciascuna prova;
- massimo 4 punti per la domanda n.5) di ciascuna prova;
- 3 punti per risposta esatta alla domanda n.6) di ciascuna prova.

Criteri di valutazione dei titoli

A. **LAUREA: max 5 punti su 15**

- Laurea di I livello rilasciata secondo l'ordinamento successivo ai DM 509/99 e DM 270/2004 max 4 punti su 15, secondo la seguente suddivisione:

Voto laurea	Punti assegnati
da 67/110 a 75/110	punti 0,40
da 76/110 a 84/110	punti 0,80
da 85/110 a 89/110	punti 1,20
da 90/110 a 94/110	punti 1,60
da 95/110 a 99/110	punti 2,00
da 100/110 a 103/110	punti 2,40
da 104/110 a 106/110	punti 2,80
da 107/110 a 109/110	punti 3,20
110/110	punti 3,60
110/110 con lode	punti 4,00

Il punteggio viene attribuito al voto di laurea relativo al titolo di studio conseguito con miglior profitto nell'ambito di quelli dichiarati per l'ammissione alla selezione previsti dall'art.2, comma 1 lett. a) del bando di selezione.
Il punteggio viene calcolato unicamente sulla base di quanto dichiarato dai candidati nella domanda di partecipazione ai fini dell'ammissione alla selezione. In caso di mancata indicazione della votazione conseguita, sarà considerata la votazione minima e pertanto non sarà attribuito alcun punteggio.

- Diploma di Laurea rilasciato secondo l'ordinamento previgente ai DM 509/99 e DM 270/2004 o della Laurea Specialistica/Magistrale rilasciata secondo l'ordinamento di cui ai DM 509/99 e DM 270/2004:
  max 1 punto su 15, secondo i seguenti criteri:
  1 punto per il possesso di diploma di laurea o di laurea specialistica (LS) o magistrale (LM) prevista dall'art.2, comma 1 lett. a) del bando di selezione. In caso di laurea specialistica o magistrale tale punteggio è attribuibile solo nel caso in cui essa sia il proseguimento di una laurea triennale conseguita dal

candidato tra quelle previste come requisito minimo di partecipazione all'art.2, comma 1 lett. a) del bando di selezione.

B) TITOLI ACCADEMICI POST-LAUREA : solo se giudicati attinenti alla professionalità oggetto di selezione, max 2 punti su 15 assegnati come di seguito indicato:

Tipologia titolo	Punti assegnati
Titolo di Dottore di Ricerca (PhD)	2
Master universitario di secondo livello	2
Laurea ulteriore rispetto al titolo di studio per l'ammissione al concorso	2
Diploma di specializzazione	2
Master universitario di primo livello	1

C. ATTIVITA' LAVORATIVA prestata con rapporto di lavoro subordinato o autonomo presso università, soggetti pubblici o privati, giudicata attinente alla professionalità oggetto di selezione max 5 punti su 15 assegnati come di seguito indicato:

a. **Attività di lavoro subordinato max 2,5 punti**

c/o amministrazioni non universitarie e c/o privati: 0,5 p. per ogni semestre
c/o amministrazioni universitarie: 0,5 p. per ogni semestre
Per periodi inferiori a sei mesi verranno attribuiti punteggi in proporzione.

b. **Attività di lavoro autonomo max 2,5 punti**

c/o amministrazioni non universitarie e c/o privati: 0,5 p. per ogni semestre
c/o amministrazioni universitarie: 0,5 p. per ogni semestre
Per periodi inferiori a sei mesi verranno attribuiti punteggi in proporzione.

D. TITOLI PERTINENTI: max 3 punti su 15 assegnati, come di seguito indicato:

Abilitazione professionale 3 punti
Iscrizione all'Ordine professionale 3 punti
Incarichi professionali 1 punto per ciascun incarico
Pubblicazioni 1 punti per ciascuna pubblicazione
Saranno valutate pubblicazioni in cui il candidato è autore/coautore che dimostrino la sua conoscenza e/o competenza in uno degli ambiti indicati nell'art. 1 del bando

Prove scritte

PROVA 1

DOMANDA 1
Descrivere il comportamento del seguente script? Quale è il suo output? Ci sono dei casi in cui produce un errore?

```
#!/bin/bash

i=1
m=0
echo "Inserire 5 numeri"
while [ $i -le 5 ]
do
  read j
  m=$((m + j))
  i=$((i + 1))
done
g=$(echo $m / 5 | bc -l)
echo $g
```

DOMANDA 2
Dare almeno 3 motivazioni per cui è utile adottare un sistema di virtualizzazione

DOMANDA 3
Descrivere il funzionamento delle vulnerabilità di cross site scripting e quali sono le differenze con l'SQL Injection.

DOMANDA 4 (scegliere una risposta)
Quanti host si possono inserire in una sottorete con netmask /28 ?

A. 14
B. 16
C. 18
D. 30

Motivare brevemente la risposta

DOMANDA 5
Dove è possibile ottenere il mapping tra ip e dominio e di dominio (scegliere anche più di una risposta valida)

a. Local client cache
b. Local DNS server cache
c. Authoritative DNS servers
d. Non-authoritative DNS servers

DOMANDA 6
In sistemi Windows, Active Directory rappresenta (scegliere una risposta):

a. Il programma di gestione delle partizioni di un disco
b. Uno strumento di gestione globale delle risorse della rete
c. Un software di backup dell'intero File System
d. Le directory attive sul file system corrente

PROVA 2

DOMANDA 1
Descrivere il comportamento del seguente script? Quale è il suo output? Ci sono dei casi in cui produce un errore?

```
#!/bin/bash

declare -a a
a=(1 2 3 4 5 6 7)
i=1
m=0
while [ $i -le 5 ]
do
  j=${a[i]}
  m=$((m + j))
  i=$((i + 1))
done
g=$(echo $m / 5 | bc -l)
echo $g
```

DOMANDA 2
Cosa si intende per sideloads delle app su dispositivi mobile e perchè può concretamente rappresentare un problema di privacy e sicurezza

DOMANDA 3
Descrivere cosa si intende per container applicativo e in che modo questi si differenziano dalla virtualizzazione classica

DOMANDA 4
Avendo a disposizione inizialmente una netmask a /22, quale tra le seguenti maschere permette di creare 5 nuove sottoreti con il maggior numero possibile di host? (scegliere una risposta)

A. /25
B. /24
C. /26
D. /27

Descrivere brevemente il perché della scelta fatta

DOMANDA 5
Relativamente al protocollo NTP (Network Time Protocol),quale delle seguenti affermazioni è vera? (possibile più di una risposta)

a. Usa TCP
b. Ci sono router che possono funzionare sia da clients che da servers
c. Usa UDP
d. Usa la porta 123

DOMANDA 6
Un router packet filtering di un firewall (scegliere una risposta):

a. filtra a livello applicativo
b. filtra a livello di indirizzi IP
c. filtra a livello di porta
d. nessuna delle precedenti

PROVA 3

DOMANDA 1
Descrivere il comportamento del seguente script? Quale è il suo output? Ci sono dei casi in cui produce un errore?

```
#!/bin/bash

echo "Inserire un numero"
read m
u=0
while [ $m -gt 0 ]
do
   g=$((m % 10))
   z=$((z + g))
   m=$((m / 10))
done

echo $z
```

DOMANDA 2
Descrivere in sintesi come funzionano i gestori di macchine virtuali in un'architettura cloud

DOMANDA 3
Descrivere le problematiche di sicurezza relative alle reti wireless e come eventualmente sia possibile rendere sicura una connessione wifi da un accesso wifi pubblico

DOMANDA 4
Quale dei seguenti indirizzi si trova nella stessa sottorete di un pc che sta usando 10.0.35.90 /23 ? (Scegliere una risposta)

a. 10.0.32.1
b. 10.0.35.254
c. 10.0.36.15
d. 10.0.33.92

Spiegare brevemente perché

DOMANDA 5
Un client non-DHCP non può accedere alla rete se sulla porta è abilitata la funzionalità di ARP Inspection. Quale delle seguenti risolverebbe il problema (scegliere più di una risposta)?

a. Configurare una ACL per ARP sulla porta dello switch
b. Abilitare DHCP relay sulla porta
c. Abilitare la funzionalità di ip helper sulla porta
d. Rendere il client un client DHCP compatibile

DOMANDA 6
Seguendo le linee guida attuali di Microsoft, è possibile connettere la directory locale ad Azure AD (scegliere una risposta)?

a. si
b. si ma occorre attivare uno specifico servizio su Azure
c. si ma occorre attivare uno specifico servizio sul Domain Controller dell'AD locale
d. no

Quesiti dei colloqui e testo per l’accertamento della lingua inglese

Orale 1

1- Descrivere brevemente le funzionalità di uno switch di rete

2- Parlare delle problematiche relative alla disposizione fisica dei server in una sala CED. Come funzionano gli armadi rack? Esistono delle misure standard? Di quali tipologie di rack si è a conoscenza? Descrivere una o più modalità di disposizione degli armadi in funzione delle soluzioni di continuità elettrica e di raffreddamento

3-Descrivere come si potrebbe implementare un sistema di autenticazione mediante directory LDAP. Descrivere in relazione a quanto detto quali vantaggi è possibile ottenere mediante l'utilizzo di una VPN e un Firewall centralizzato.

4-Il candidato, con riferimento alla disciplina privacy vigente, esponga il concetto di data-breach. Effettui alcune esemplificazioni pratiche. Approfondisca le modalità con cui deve essere notificato, a chi ed in quali tempi.

5-Il candidato esponga le linee generali di sicurezza richieste nella Pubblica Amministrazione dalle Misure Minime AGID con riferimento a: "la valutazione e correzione continua delle vulnerabilità"

Orale 2

1- Descrivere brevemente le funzionalità di un router

2- Parlare delle problematiche relative alla disposizione di gruppi di continuità in una sala CED. Che tipo di continuità operativa possono garantire? Esistono soluzioni che assicurano una maggiore continuità operativa? Quali soluzioni si possono individuare per garantire la copertura di lunghi periodi di mancanza di corrente e quali problematiche ne conseguono?

3- Descrivere come si potrebbe implementare un sistema di autenticazione mediante Single Sign ON. Descrivere in relazione a quanto detto quali vantaggi è possibile ottenere mediante l'utilizzo di una VPN e un Firewall centralizzato.

4-Il candidato approfondisca almeno 4 dei principi fondamentali richiamati dal GDPR nel campo della Protezione dei Dati Personali.

5-Il candidato esponga le linee generali di sicurezza richieste nella Pubblica Amministrazione dalle Misure Minime AGID con riferimento a: "l'uso appropriato dei privilegi di amministratore"

Orale 3

1- Descrivere brevemente le funzionalità di un firewall

2- Descrivere i sistemi di condizionamento in una sala CED. Come si misura la capacità di condizionamento? Conosci varie alternative nel progettare un sistema di raffreddamento? Cosa si potrebbe fare per garantire la continuità operativa del condizionamento?

3- Descrivere come si potrebbe implementare un sistema di autenticazione mediante Active Directory o tecnologie simili. Descrivere in relazione a quanto detto quali vantaggi è possibile ottenere mediante l'utilizzo di una VPN e un Firewall centralizzato.

4- A norma GDPR e sulla base delle recenti indicazioni del Garante per la protezione dei Dati personali, quali sono le misure privacy da implementare obbligatoriamente su un sito WEB?

5-Il candidato esponga le linee generali di sicurezza richieste nella Pubblica Amministrazione dalle Misure Minime AGID con riferimento a: "le copie di sicurezza"

Orale 4

1- Descrivere brevemente la le funzionalità di un access point

2- Parlare delle problematiche relative al monitoraggio remoto dello stato di salute di tutte le apparecchiature presenti in sala server, con riferimento alla temperatura e alla continuità elettrica. Eventualmente si è a conoscenza di meccanismi che permettono di controllare lo stato del server da remoto limitando gli accessi fisici nelle sale server?

3- Descrivere come si potrebbe implementare un meccanismo di autenticazione basato su un database utenti. Quale precauzioni prenderesti per la memorizzazione delle password? Descrivere in relazione a quanto detto quali vantaggi è possibile ottenere mediante l'utilizzo di una VPN e un Firewall centralizzato.

4- Il candidato, con riferimento al GDPR, descriva in maniera dettagliata i ruoli delle figure chiave coinvolte nella Protezione dei Dati Personali all'interno di una organizzazione.

5- Il candidato esponga le linee generali di sicurezza richieste nella Pubblica Amministrazione dalle Misure Minime AGID con riferimento a: "le difese contro il malware"

Orale 5

1- Descrivere brevemente le funzionalità di un captive portal

2- Parlare delle problematiche relative alla sicurezza per garantire l’accesso fisico a diversi operatori. Come organizzeresti gli accessi e che tipo di precauzioni prenderesti per monitorare il sorgere di problemi e per esempio garantire che ogni operatore agisca solo sugli ambiti per cui è autorizzato?

3- Descrivere come si potrebbe implementare un meccanismo di autenticazione basato sull'autenticazione a due fattori. Descrivere in relazione a quanto detto quali vantaggi è possibile ottenere mediante l’utilizzo di una VPN e un Firewall centralizzato.

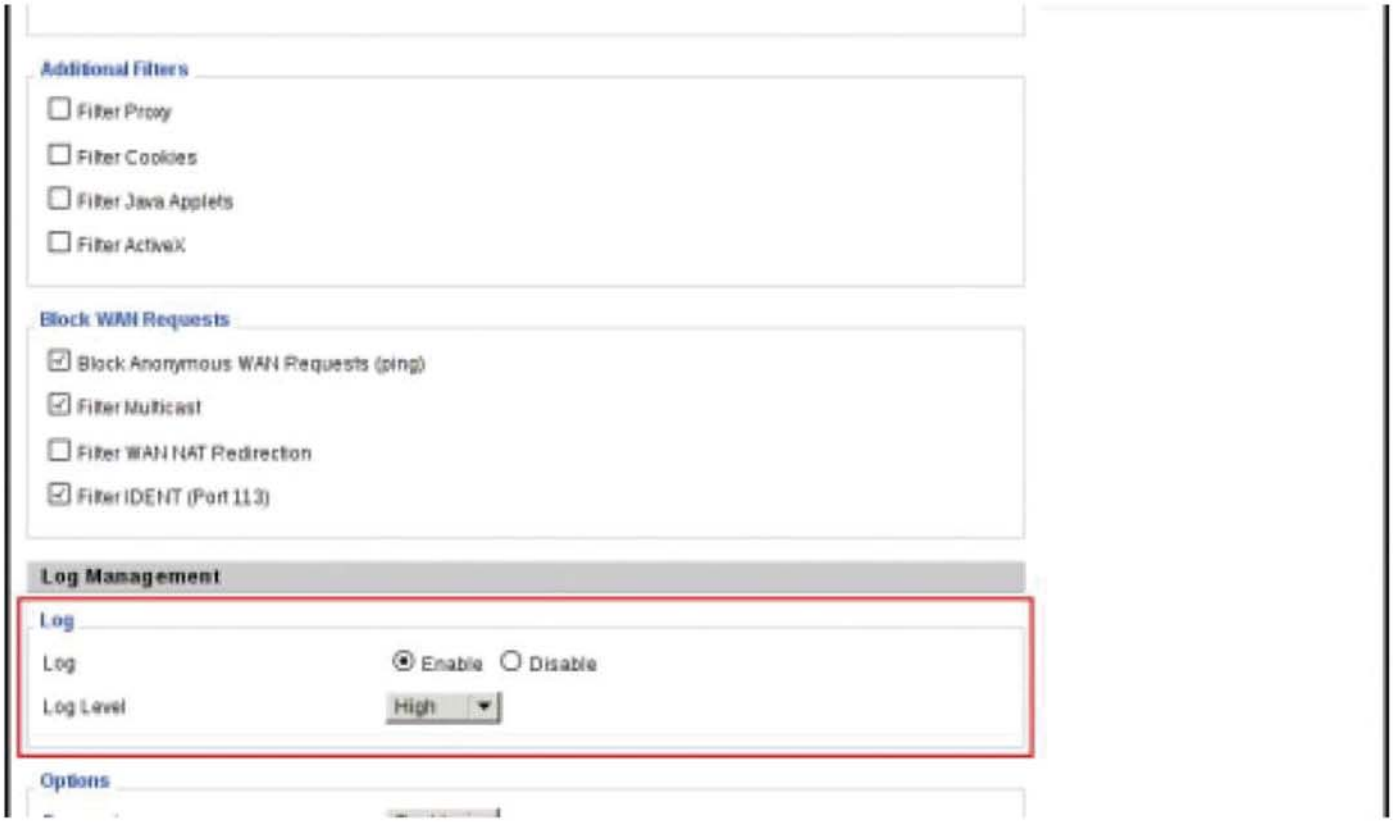
4- Il candidato, con riferimento alla disciplina GDPR inquadri, anche grazie ad esemplificazioni, lo stretto legame che esiste tra privacy e security

5- Il candidato esponga le linee generali di sicurezza richieste nella Pubblica Amministrazione dalle Misure Minime AGID con riferimento a: “Inventario dei software autorizzati e non autorizzati”

Networking (/tag/networking) Security (/tag/security) Wi-Fi (/tag/wi-fi)

Wi-Fi Mini Honeypot

by Marcin Teodorczyk (/users/marcin-teodorczyk) on March 29, 2013



Do you have an old, unused wireless router collecting dust? Have some fun and make a Wi-Fi honeypot with it!

Recently, I've been playing with some new wireless gear. It's nothing special: 200mW Atheros-based transceiver and 18dBi yagi antenna. I'm living in an apartment in a city of about 640,000 people. I've pointed the antenna to a window and passively received about 30 wireless ESSIDs, three of which were unsecured (open) and six secured with WEP (easily crackable). I haven't connected to any of them, of course, but that gave me some ideas.

What if I deployed a wireless access point deliberately open? Some people eventually will connect and try to use it for Internet access—some might be malicious, and some might think that it's a hotspot. And, what if I deployed a similar access point, but secured with easily crackable WEP this time? Well, in my humble opinion, it's not possible to unconsciously crack WEP. If somebody that I don't know connects to this AP, I've just been attacked. All I need to do is to monitor.

That's exactly a wireless honeypot: fake access point, deliberately unsecured or poorly secured and monitored, so you can get as much information about attackers as you want. Such honeypots are especially useful in large networks as early threat indicators, but you also can play with them on your home network, just for fun and research.

You can build a wireless honeypot with old hardware, some spare time and, of course, a Linux-based solution. OpenWrt (https://openwrt.org) and DD-WRT (http://www.dd-wrt.com/site/index) are the two most popular Linux-based firmware projects for routers. I use them and some old spare routers in this article to show you how to build three kinds of honeypots: a very basic one that logs only information about packets sent by users into its memory, a little more sophisticated one with USB storage that logs a few more details about malicious clients to the storage, and finally, a solution that redirects HTTP traffic through a proxy that not only can log, but also interfere with communication.

Basic Honeypot with DD-WRT

Building a very basic wireless honeypot shouldn't take you more than an hour or two. Just grab your old router and pick up the firmware. Be sure to look at supported routers for both DD-WRT and OpenWrt. In my case, it came up that the router is supported only by DD-WRT, as it has 32MB of RAM and 4MB of Flash memory. OpenWrt's hardware requirements are a little bigger.

Next, flash your router (that's the risky part). Basically, you need to download the firmware for your machine and upload it to the memory. On some routers, it's as easy as clicking a button on the Web interface. On others, you have to connect through a serial cable, for example. Remember, this step can be dangerous. Make a backup first and be sure to read the instructions carefully on the DD-WRT/OpenWrt sites.

After successfully flashing your router, you should see an enhanced (as compared to the original one) Web interface. Now, set up SSH access and wireless network parameters. If you don't know how, you can find detailed instructions on the DD-WRT home page. As it is

going to be a honeypot, I would suggest WEP, which should attract potential attackers. At the same time, it won't be so vulnerable to false positives—people with devices automatically connecting to an open network.

If you can log in as root and see the prompt, you're ready for the next step: enabling system logging. You can do this using the Web interface: Services→Services→System Log and Security→Log Enable (Figure 1).

Figure 1. Enabling System Logging

You also can set a few ESSIDs instead of just one: Wireless→Basic Settings→Virtual Interfaces. After that, your honeypot will be seen as a few networks—at least at first glance. This increases the probability of attacks, especially when there are many other networks in your neighborhood.

Remember, you don't have to connect your honeypot to the Internet. In fact, you shouldn't, as you have no control of what potential users might do with the Internet access. After configuring it as described above, test whether it logs your connections. DD-WRT writes the log in /var/log/messages by default. You can check it using SSH. Here's an example fragment of such a log:

SysAdmin (/tag/sysadmin) Security (/tag/security) Networking (/tag/networking) HOWTOs (/tag/howtos) Firewalls (/tag/firewalls)

Understanding Firewalld in Multi-Zone Configurations

by Nathan Vance (/users/nathan-vance-0) on February 2, 2017

Stories of compromised servers and data theft fill today's news. It isn't difficult for someone who has read an informative blog post to access a system via a misconfigured service, take advantage of a recently exposed vulnerability or gain control using a stolen password. Any of the many internet services found on a typical Linux server could harbor a vulnerability that grants unauthorized access to the system.

Since it's an impossible task to harden a system at the application level against every possible threat, firewalls provide security by limiting access to a system. Firewalls filter incoming packets based on their IP of origin, their destination port and their protocol. This way, only a few IP/port/protocol combinations interact with the system, and the rest do not.

Linux firewalls are handled by netfilter, which is a kernel-level framework. For more than a decade, iptables has provided the userland abstraction layer for netfilter. iptables subjects packets to a gauntlet of rules, and if the IP/port/protocol combination of the rule matches the packet, the rule is applied causing the packet to be accepted, rejected or dropped.

Firewalld is a newer userland abstraction layer for netfilter. Unfortunately, its power and flexibility are underappreciated due to a lack of documentation describing multi-zoned configurations. This article provides examples to remedy this situation.

Firewalld Design Goals

The designers of firewalld realized that most iptables usage cases involve only a few unique IP sources, for each of which a whitelist of services is allowed and the rest are denied. To take advantage of this pattern, firewalld categorizes incoming traffic into zones defined by the source IP and/or network interface. Each zone has its own configuration to accept or deny packets based on specified criteria.

Another improvement over iptables is a simplified syntax. Firewalld makes it easier to specify services by using the name of the service rather than its port(s) and protocol(s)—for example, samba rather than UDP ports 137 and 138 and TCP ports 139 and 445. It further simplifies syntax by removing the dependence on the order of statements as was the case for iptables.

Finally, firewalld enables the interactive modification of netfilter, allowing a change in the firewall to occur independently of the permanent configuration stored in XML. Thus, the following is a temporary modification that will be overwritten by the next reload:

```
# firewall-cmd <some modification>
```

And, the following is a permanent change that persists across reboots:

```
# firewall-cmd --permanent <some modification>
# firewall-cmd --reload
```

Zones

The top layer of organization in firewalld is zones. A packet is part of a zone if it matches that zone's associated network interface or IP/mask source. Several predefined zones are available:

```
# firewall-cmd --get-zones
block dmz drop external home internal public trusted work
```

An active zone is any zone that is configured with an interface and/or a source. To list active zones:

```
# firewall-cmd --get-active-zones
public
  interfaces: eno1 eno2
```

Interfaces are the system's names for hardware and virtual network adapters, as you can see in the above example. All active interfaces will be assigned to zones, either to the default zone or to a user-specified one. However, an interface cannot be assigned to more than one zone.

In its default configuration, firewalld pairs all interfaces with the public zone and doesn't set up sources for any zones. As a result, public is the only active zone.

Sources are incoming IP address ranges, which also can be assigned to zones. A source (or overlapping sources) cannot be assigned to multiple zones. Doing so results in undefined behavior, as it would not be clear which rules should be applied to that source.

Since specifying a source is not required, for every packet there will be a zone with a matching interface, but there won't necessarily be a zone with a matching source. This indicates some form of precedence with priority going to the more specific source zones, but more on that later. First, let's inspect how the public zone is configured:

PoE (/tag/poe) Networking (/tag/networking)

PoE, PoE+ and Passive POE

by Shawn Powers (/users/shawn-powers) on November 6, 2017

I've been installing a lot of POE devices recently, and the different methods for providing power over Ethernet cables can be very confusing. There are a few standards in place, and then there's a method that isn't a standard, but is widely used.

802.3af or Active PoE:

This is the oldest standard for providing power over Ethernet cables. It allows a maximum of 15.4 watts of power to be transmitted, and the devices (switch and peripheral) negotiate the amount of power and the wires on which the power is transmitted. If a device says it is PoE-compliant, that compliance is usually referring to 802.3.af.

802.3at or PoE+:

The main difference between PoE and PoE+ is the amount of power that can be transmitted. There is still negotiation to determine the amount of power and what wires it's transmitted on, but PoE+ supports up to 25.5 watts of power. Often, access points with multiple radios or higher-powered antennas require more power than 802.3af can supply.

Passive PoE:

This provides power over the Ethernet lines, but it doesn't negotiate the amount of power or the wires on which the power is sent. Many devices use Passive PoE (notably, the Ubiquiti line of network hardware often uses 24v Passive PoE) to provide power to remote devices. With Passive PoE, the proprietary nature of the power specifics means that it's often wise to use only power injectors or switches specifically designed for the devices that require Passive PoE. The power is "always on", so it's possible to burn out devices if they're not prepared for electrified Ethernet wires, or if the CAT5 cabling is wired incorrectly.

Figure 1. This AP requires a Passive PoE 24v supply. It can be confusing, because even though it says it's PoE, it won't power on using a standard 802.3af switch.

The best practice for using power over Ethernet is either to use equipment that adheres to the 802.3af/at standards or to use the power injectors or switches specifically designed for the hardware. Usually, the standard-based PoE devices are more expensive, but the ability to use any brand PoE switch and device often makes the extra expense worthwhile. That said, there's nothing wrong with Passive PoE, as long as the correct power is given to the correct devices.

Shawn is Associate Editor here at *Linux Journal*, and has been around Linux since the beginning. He has a passion for open source, and he loves to teach. He also drinks too much coffee, which often shows in his writing.

Recent Articles

Configuring TACACS+ Server With A Simple GUI (/content/configuring-tacacs-server-simple-gui)

Dmitriy Kuptsov (/users/dmitriy-kuptsov)

(/content/configuring-tacacs-server-simple-gui)

How Can You Install Google Chrome Browser on Debian? (/content/how-can-you-install-google-browser-debian)

Supama Ganguly (/users/supama-ganguly)

(/content/how-can-you-install-google-browser-debian)

Sending Emails? Send them from Linux Terminal (/content/sending-emailssend-them-linux-terminal)

Supama Ganguly (/users/supama-ganguly)

(/content/sending-emailssend-them-linux-terminal)

7 Important Linux Commands for Every Linux User (/content/7-important-linux-commands-every-linux-user)

Supama Ganguly (/users/supama-ganguly)

(/content/7-important-linux-commands-every-linux-user)

In PuTTY, Scripted Passwords are Exposed Passwords (/content/putty-scripted-passwords-are-exposed-passwords)

Charles Fisher (/users/charles-fisher)

(/content/putty-scripted-passwords-are-exposed-passwords)

How To Pick a Linux Distribution for Non-Techies (/content/how-pick-linux-distribution)

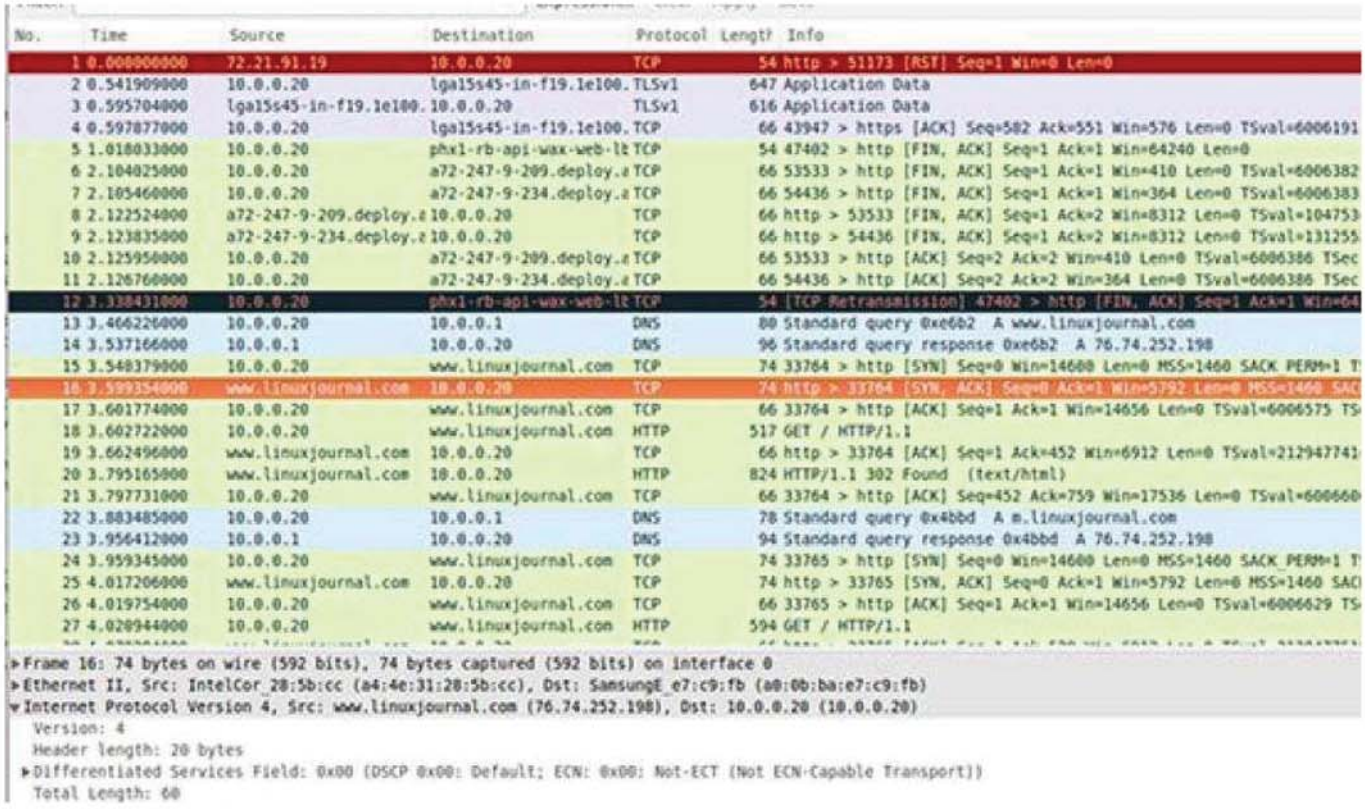
Ujjwal Anand (/users/ujjwal-anand)

(/content/how-pick-linux-distribution)

Mobile (/tag/mobile) Networking (/tag/networking) HOW-TOs (/tag/how-tos)

Monitoring Android Traffic with Wireshark

by Brian Trapp (/users/brian-trapp) on August 14, 2014



The ubiquity and convenience of smartphones has been a real boon for getting information on the go. I love being able to jump on a Wi-Fi hotspot, catch up on my mail, check my banking balance or read the latest tech news—all without having to bring along or boot up a laptop. Now that mobile development is mainstream, most of this access is done via specialized apps, instead of via a Web browser.

This migration away from direct Web access in favor of dedicated smartphone apps has made for a richer user experience, but it also has made knowing exactly what is going on "under the hood" a lot harder. On our Linux boxes, there are many tools to help user peer into the internals of what's going to and from the machine. Our browsers have simple HTTP versus HTTPS checks to see if there's encryption, and there are simple but easy-to-use browser plugins like Firebug that let us view exactly what's being sent and retrieved over the Web. At the operating system level, powerful tools like Wireshark let us drill down even further, capturing *all* traffic flowing through a network interface. Smartphones usually are locked up to a point where it's almost impossible for a regular user to run any network monitoring or tracing software directly on the phone—so how can a curious user get access to that phone traffic?

Fortunately, with just a little bit of work, you can use Linux to transform almost any laptop into a secret-sharing wireless access point (WAP), connect your phone and view the data flowing to and from the phone with relative ease. All you really need is a laptop running Linux with one wireless and one Ethernet connection.

Intercepting Traffic

The first step is to set up your own "naughty" WAP where you can capture and log all the Internet traffic passing through it—simulating the kind of information that a rogue employee could be obtaining from a coffee-shop Wi-Fi hotspot. Let's do this in a distribution-independent way that doesn't mess around with your existing router (no need to change security settings) and doesn't require rooting or installing anything unseemly on your phone.

False Starts

It may be tempting to try a shortcut for capturing this traffic. Here are a few techniques I tried and discarded before sticking with a hostapd/dnsmasq/iptables solution.

Ubuntu's Built-in Hotspots:

Ubuntu has a handy "Use as Hotspot" feature tucked away in its networking settings. Unfortunately, it creates hotspots in ad hoc mode, which isn't compatible with most versions of Android. I didn't try Fedora's implementation, but the method I recommend instead will work on any distribution.

Monitor Mode:

It's tempting just to put the wireless card in monitor mode and capture *all* wireless traffic, independent of SSID. This is pretty cool, but there are quite a few "gotchas":

- The drivers for your wireless card must support monitor mode. Many, but not all cards support this mode.
- Your capture needs to include the four WPA "handshake" packets.
- You'll probably have to compile and use airmon-ng to start monitor mode and then capture on the mon0 pseudo-device airmon creates.
- If the WAP is using encryption, the packets you capture also will be encrypted. Wireshark does have a facility to help decode the packets, but you'll need to enter information about the security scheme used by the WAP and toggle a few sets of options until the decoded packets look right. For a first-time user, it's hard enough making sense out of Wireshark dumps without having to worry about toggling security options on and off.

Capturing with the Android Emulator:

Another approach would be to use an Android emulator on your capture device, install and then run the target application, and capture the traffic from the emulator. It's much harder than it sounds actually to get a banking app on the emulator though:

- Due to recent Android licensing changes, the major Android VMs no longer include the Google Play store. (I tried both the Android SDK and the free product from Genymotion.)

- If your phone isn't rooted, it's not easy to get the application's .apk off your phone and onto the VM.

To turn a laptop into a WAP, you'll first use hostapd to use the wireless card as an access point mode (broadcasting an SSID, authenticating with security and so on). Next, you'll use dnsmasq to provide DNS and DHCP services for clients connecting on the wireless connection. Finally, iptables' masquerading features will be used to direct IP traffic from clients on the wireless connection to the Internet (via your Ethernet connection), and then rout responses back to the correct client on the wireless side.

hostapd

hostapd is a small utility that lets you create your own wireless access point. Installation is straightforward, and configuration is just as easy. Most wireless cards and modern kernels will be using the mac80211 driver. Check yours via `lsmod|grep mac80211`. If that's your driver, find your wireless device via `ifconfig`, and set up the SSID of your choice as shown below for an unsecured, totally open access point:

```
===[/etc/hostapd/hostapd.conf]======
interface=wlan0
driver=nl80211
ssid=WatchingU
channel=1
===[/etc/hostapd/hostapd.conf]======
```

I recommend not using Wi-Fi security for this test; it would be overkill, as your access point will only be temporary. Should you desire a more permanent solution, hostapd supports many different authentication options.

dnsmasq

Now that hostapd is ready to start letting clients connect to your wireless connection, you need dnsmasq to serve DCHP and provide DNS for your access point. Fortunately, dnsmasq is also very easy to install and configure. The example below is the minimum required. Make sure the `dhcp-range` you specify will not conflict with anything already on your network. By default, dnsmasq will read your existing /etc/resolv.conf and propagate the DNS settings listed there to its clients. That's a pretty sane default configuration, but if you need something else, use the `no-resolv` option and specify the DNS servers manually:

```
========[/etc/dnsmasq.conf]===============
interface=wlan0
dhcp-range=10.0.0.3,10.0.0.20,12h
========[/etc/dnsmasq.conf]===============
```

iptables

The final piece of your wireless access point is iptables, which will use IP Masquerading to get the traffic from the wireless connection, send it over the wired connection and route any responses to back to the correct source on the wireless side. There are many distribution-specific ways to save and script iptables rules, but it's simpler to create a distribution-independent shell script to enable iptables and network address translation (NAT). A script for iptables that ties in hostapd and dnsmasq would look like the following (modify the wlan0 and eth0 entries to match your system):

LINUX JOURNAL

SysAdmin (/tag/sysadmin) Storage (/tag/storage) SNMP (/tag/snmp) Networking (/tag/networking)

SNMP

by Andrew Kirch (/users/andrew-kirch-0) on January 24, 2017

How would you find out how much RAM is free on your Linux desktop? That's a really easy question with a lot of answers— `free` , any of the implementations of `top` and Glances all are valid responses. How would you find out how much RAM is free on 200 Linux instances, which are running on a mixture of real and virtual hardware, in dozens of physical locations spread out around the globe? That's a much bigger problem, and there is a tool to make the job easier. However, the lack of upkeep on the standards and lack of development support for the Linux implementation are resulting in proprietary standards creeping in where there once was a more open standard.

SNMP (Simple Network Management Protocol) was designed in 1990 to read and write structured data on devices attached to a network, such as how much free RAM there is. Yes, and this is important, the M in SNMP really does stand for "Management", not "Monitoring". Although SNMP is usually used to request operational status information, the SNMP "write" functionality can be used to change the configuration on remote devices. Given the lack of security and authentication in the SNMP protocol, SNMP "write" functionality almost always is disabled on the modern internet, and I will not be discussing it here.

History of SNMP

The original IETF (Internet Engineering Task Force) RFC (Request for Comments) standard for SNMP v1 was published by the IETF in 1990. SNMP v2 was published in 1994–1996 as a series of RFCs and included the first effort to secure SNMP. This effort proved unpopular due to the load it placed on network hardware, which, at the time, had very low performance CPUs. This performance issue exists today and still can cause problems for administrators attempting to secure SNMP. Due to the performance problems, SNMP v2c (SNMP v2 with SNMP v1 communities) became the standard. Concurrently with the release of SNMPv2c, the public began to access the internet, and during the next decade, security would become a serious problem with SNMP since SNMP v2c was entirely unencrypted. SNMPv3 came along in 2003 and added TLS to the previous implementation of SNMP v2c. If all of this seems a bit complicated and unnecessary, it's important to know that many implementations of SNMP still ship with support for SNMP v1, v2c and SNMP v3. This means you're likely to see all of them in the wild.

How Is SNMP Used?

One of the challenges on a modern network is scale, and achieving scale requires managing resources. SNMP provides an agent, which listens for incoming SNMP requests on each host, and a standard communications protocol allowing a central collection system called a Network Management System (NMS) to collect data. NMS is outside the scope of this article, but there are many good open-source NMSes, including Zabbix, OpenNMS, Nagios and Zenoss. The data collected by each NMS is pretty standard, and it includes basic systems information like CPU, memory, network and storage utilization.

SNMP Data Structure

SNMP isn't just an agent, it's also a data structure. Each object in the data structure has an Object IDentifier, or OID. Each OID belongs to an MIB, or Management Information Base. These object identifiers and the hierarchical structure function as a tree. Each sequential number is a branch and has a meaning, and each branch is separated by periods (.), somewhat like an IPv4 address. This means that the meaning of an OID can be decoded very simply.

Given an example OID, `1.3.6.1.2.1.1.1.0`, each number has the following meaning:

- 1 = iso
- 3 = org
- 6 = dod
- 1 = internet
- 2 = IETF Management
- 1.1 = SNMP MIB-2 System
- 0 = sysDescr

From the decoded values, it can be determined that this OID is from the IETF standard MIB (more on MIBs later in the article), and it provides a system description of some sort. Let's look at a real-world example from a CentOS 6 box:

```
1.3.6.1.2.1.1.1.0 = STRING: "Linux foo.example.lan
 ↪2.6.32-573.1.1.v6.i686 #1 SMP Fri Aug 21 14:37:07 MDT 2015 i686"
```

From this description, you can determine that the system this agent is running on is running Linux, 2.6.32, and is 32-bit.

Nearly every OID starts with "1.3.6.1", and the reason for this should be obvious. The modern public internet originally was created by the United States Department of Defense, and at one time, TCP/IP was called the "DOD Model". Since these values are in every OID, they aren't all that useful for identifying what that OID does, and they generally can be ignored.

After 1.3.6.1, there are more types of OID. If the MIB continues with 1.2, as with the example above, the description of the OID can be found in the standard IETF MIB. If it continues with 1.4, the MIB is "private", and you will need to get the MIB from your hardware vendor. Despite being called "private", these MIBs are almost always available.

What Types of OIDs Are There and How Is Each Used?

There are many different types of OIDs so that SNMP can provide an extensive and extensible variety of information. The example from the previous section, 1.3.6.1.2.1.1.1.0, is a `STRING`. You can tell because SNMP tells you the type of OID when you retrieve it: