Hacker Highschool

SECURITY AWARENESS FOR TEENS

LESSON 9: HACKING EMAIL

HACKING IS LEARNING
www.hackerhighschool.org

ISECOM

WARNING

The Hacker Highschool Project is a learning tool and as with any learning tool there are dangers. Some lessons if abused may result in physical injury. Some additional dangers may also exist where there is not enough research on possible effects of emanations from particular technologies. Students using these lessons should be supervised yet encouraged to learn, try, and do. However ISECOM cannot accept responsibility for how any information herein is abused.

The following lessons and workbooks are open and publicly available under the following terms and conditions of ISECOM:

All works in the Hacker Highschool Project are provided for non-commercial use with elementary school students, junior high school students, and high school students whether in a public institution, private institution, or a part of home-schooling. These materials may not be reproduced for sale in any form. The provision of any class, course, training, or camp with these materials for which a fee is charged is expressly forbidden without a license including college classes, university classes, trade-school classes, summer or computer camps, and similar. To purchase a license, visit the LICENSE section of the HHS web page at http://www.hackerhighschool.org/licensing.html.

The HHS Project is an open community effort and if you find value in this project we ask that you support us through the purchase of a license, a donation, or sponsorship.

Table of Contents

Contributors

Pete Herzog, ISECOM

Bob Monroe, ISECOM

Greg Playle, ISECOM

Marco Ivaldi, ISECOM

Simone Onofri, ISECOM

Peter Houppermans

Andrea Zwirner

Introduction

Email has been around for a long time; like longer than those socks stuffed under your bed. It predates the Internet (not your dirty socks), and is one of the first forms of electronic information exchange. Before email, we had smoke signals, half-naked guys running as messengers, bricks with notes attached, Morse code, large rocks slung over castle walls with curse words written on them, and a variety of other analog communication methods like the telephone and paper “snail mail” (not really delivered by snails). Many of these original message transmission required special tools, training, or lots of rocks. Luckily, enterprising authors created text that could be written on stone tablets or bound in books and thrown at people or read by them. One of the first books was *Smoke Signals for Dummies*.

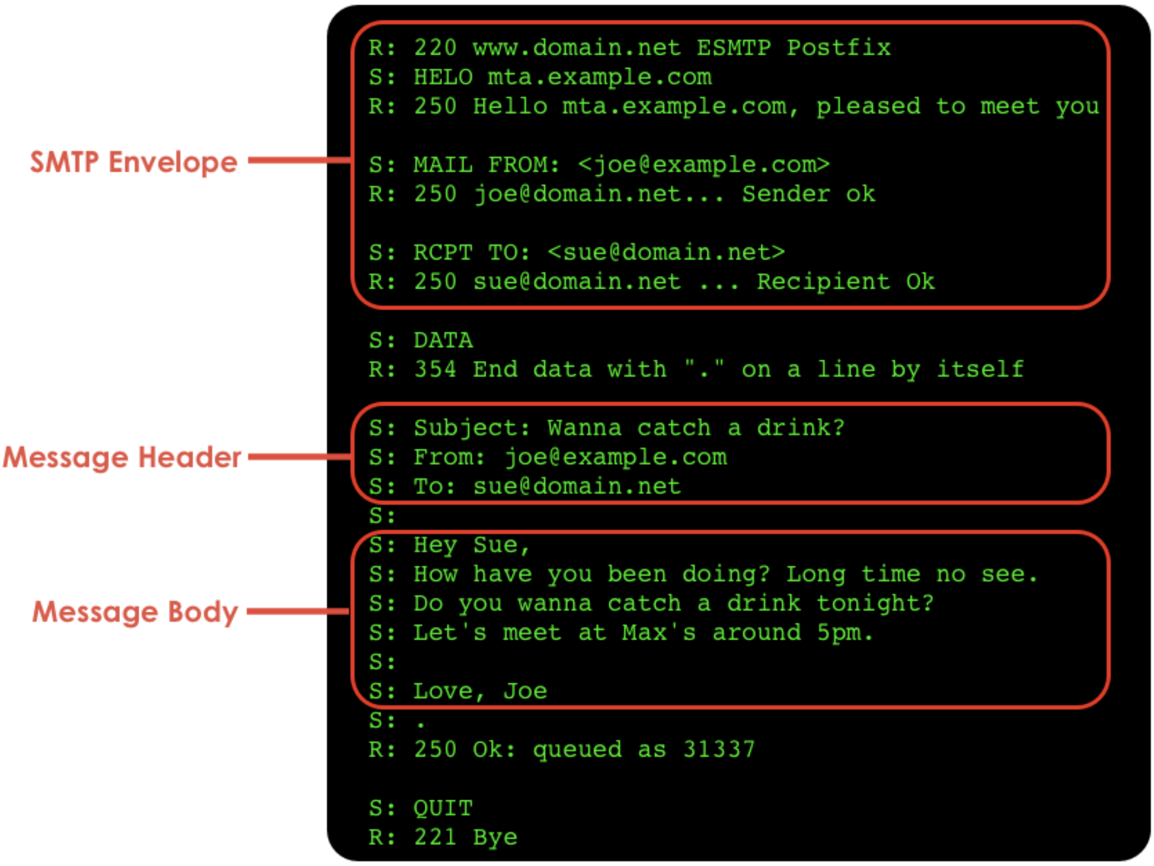
Email is based on simple store and forward principles. It can be relatively easy to use (unless you are in a huge hurry), very robust and so cheap that it is often abused for commercial and criminal purposes. Its asynchronous design allows communication to take place without the need for sender and receiver to both be online at the same time. Kind of like when your mother is talking to you and you're not paying attention until she asks you a question. You are not there for the transmission but you better be a quick deceiver. Um, receiver. A quick receiver.

In this lesson, we will focus on modern Internet email and hacking or security issues you can use for fun and profit.

Overall: How Email Works

First, we are going to pretend that you are an email. You will follow the transmission and receipt of yourself as an email, and we will identify the various components that move you along.

1. Email (you) is (are) created either using an email **client** such as Outlook, Mail, Eudora, Pegasus or Thunderbird, or on a web service like Yahoo Mail, using a web interface. It's almost funny how much email mimics "snail mail," because your message is enclosed in an envelope, like in Figure 9.1.



```
R: 220 www.domain.net ESMTP Postfix
S: HELO mta.example.com
R: 250 Hello mta.example.com, pleased to meet you

S: MAIL FROM: <joe@example.com>
R: 250 joe@domain.net... Sender ok

S: RCPT TO: <sue@domain.net>
R: 250 sue@domain.net ... Recipient Ok

S: DATA
R: 354 End data with "." on a line by itself

S: Subject: Wanna catch a drink?
S: From: joe@example.com
S: To: sue@domain.net
S:
S: Hey Sue,
S: How have you been doing? Long time no see.
S: Do you wanna catch a drink tonight?
S: Let's meet at Max's around 5pm.
S:
S: Love, Joe
S: .
R: 250 Ok: queued as 31337

S: QUIT
R: 221 Bye
```

Figure 9.1: Email message, headers and envelope

2. You are sent to a mail server called a **Mail Transmission Agent (MTA)**, which queues you for transmission. Modern mail systems do this typically via encrypted **SMTP (Simple Mail Transport Protocol)** since they require authentication to prevent abuse, and encryption protects credentials from disclosure, along with the email contents. MTAs accepting email (you) without some sort of authentication are called "open relays" and tend to be abused by senders of junk mail, also known as UCE (Unsolicited Commercial Email) or **spam**.

3. For each address ("recipient") in the message, the MTA first checks if a recipient is local (right on the same computer). If not, the MTA uses a so-called MX record (explained below) to find the server for the relevant domain. If there is no valid receiving host found, a failure message for that specific address is sent back to the sender.

4. The MTA attempts to deliver you to each address. If this fails, the MTA re-queues the message to try again later until timeout occurs and a delivery failure message is returned, usually in 48 hours. So you have to hang around for about two days. This delivery may initially be deliberately delayed by the receiving MTA as an anti-spam technique: spam software is typically less intelligent and will not queue and retry delivery (the technique is called **greylisting**). By default, this delivery takes place via **unencrypted** SMTP. Encrypted connections are the exception rather than the rule.

5. Optionally, a mail relay picks you up and routes you to your final destination. This typically happens in environments with spam and virus filtering and where security dictates a layered model, such as enterprise or government networks.

Did you catch that reference to a **layered security model**? Those heavy-duty government security guys can't create M&Ms, hard on the outside but soft in the middle. They put in lots of layers of armor: router controls and firewalls, intrusion detection systems (IDS), anti-virus, anti-malware, spam control and a whole lot more.

Which sounds pretty tough to hack. But never forget: every program you install adds more code, with more vulnerabilities, and the same goes for hardware. That cool VPN device, for instance, might give you "secure" VPN – or it might offer backdoors of its own. It depends a lot on whether you're Red Team or Blue Team how much you like this.

6. The receiving MTA expands the address if it is an alias or a mailing list. These do not need to be in the same domain: an alias can expand into a whole new email address on another server. After expansion, you are re-queued for further delivery.

7. When an email address refers to a local mailbox, you are now moved into that mailbox (unless the mailbox has exceeded its storage quota). You might be too big. You gotta stop eating so much junk food.

8. You are then picked up via the POP3 or IMAP protocol by webmail or a mail client. Here too, the connection is generally encrypted (with SSL) to prevent leaking login credentials; the protocols are POP3S and SSL IMAP. POP3 is a "pick up" process: it downloads messages, then deletes them from the server (this can be date driven). IMAP is a synchronization process that seeks to keep clients' mailboxes identical to what is on the server account (for mobile devices this is typically within a date range to preserve device storage), which makes IMAP perfect to maintain email on multiple devices at the same time.

9. Finally, most email clients now have junk mail detection built in, usually based on Bayesian, pattern scoring principles. Try sending your friend an email with "Viagra" in the Subject to see how this works.

The Three Stages of Spam Filtering

a. Receiving servers first check on origin: an SMTP connection is refused from blacklisted servers (various companies exist to provide these lists).

b. When a connection is accepted, email is then scanned for content. Some organizations are concerned they may have a message falsely marked as junk mail; they may require suspect email to be marked as junk, but still delivered.

c. Finally, most email clients now have junk mail detection built in, usually based on Bayesian, pattern-scoring principles.

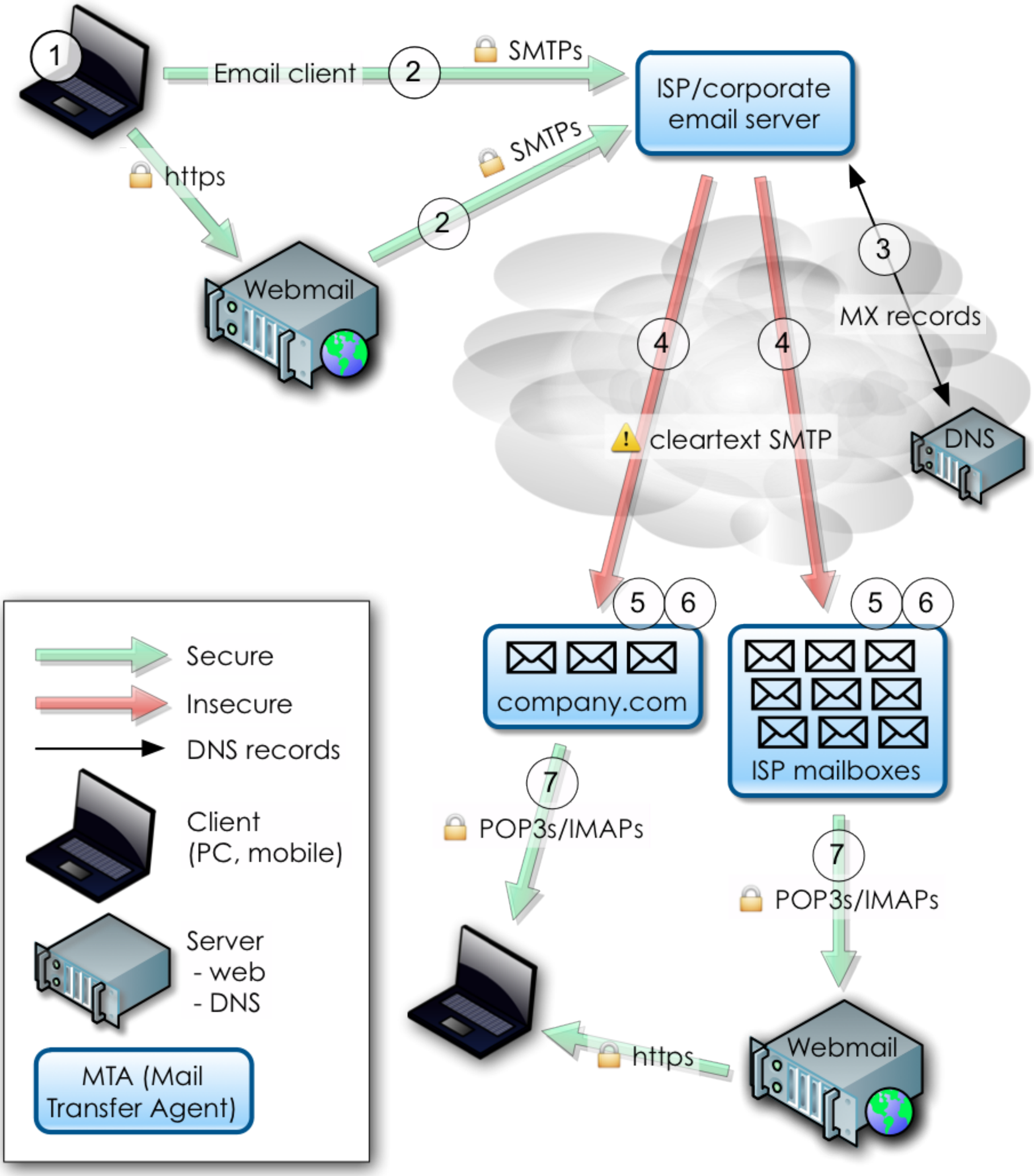


Figure 9.2: Email process flow

So, there ya go. That was easy, wasn't it? You start at one place and may or may not end up at another place, depending on whether:

- You have the correct address

- You are spam
- You are too big
- The receiving mail box is too small
- or you are too old.

With all this great information you now know how emails move through the digital world. Everything you need to know about life can come from email traffic.

- Know where you are going.
- Don't eat spam.
- Get big mail boxes (communicate a lot).
- Don't get big (Eat right and exercise to maintain a healthy lifestyle).
- Lastly, don't get old.

See how easy that is?

Feed Your Head: Email Headers

A **message**, from the SMTP point of view, consists of **headers** and a **body**. Headers are machine-parseable statements containing information of all kinds, the most basic ones being headers like 'To:' for the mail recipient or 'Subject:'. The sender address might be quickly dismissed as a basic piece of information easy to describe but we'll see that it's a more complex concept.

The body of the message contains everything else (everything other than headers) and it's not normally supposed to be parsed by MTAs (although, as we'll see, it might happen for filtering purposes). Usually the body of the message contains simple text but it can also be HTML (which often annoys really technical people), and in multi-part messages (i.e. messages with attachments) MIME is used. MIME stands for Multipurpose Internet Mail Extensions and it's a standard that is used for sending character encodings other than plain ASCII and binary content. MIME is automatically used by the email client when needed.

Some headers can be removed, some can be modified and some will be added by different components in the mail flow process. Every MTA should always add a "Received" header for tracking its role the email path during transmission. In theory, by looking at the headers you should always be able to track the original sender. We'll soon see why this is not always the case.

There's a set of headers that every email should have in order to be parseable by the SMTP standard, some headers that most SMTP implementations consider standard but that really are not, and some custom headers (X-*) that are customizable and can contain any sort of message. Think of it as a way to shift user-definable content from the body to the headers. Some of the most widely used examples are filtering applications information (X-Spam) and MUA (X-Mailer). (It's not uncommon to spot very interesting customized headers in the wild; email from security consultants may have weird ones!)

Consider this example.

```
[Sample message]


From root@isecom.org Sat Sep 30 13:50:39 2006

Return-Path: <root@isecom.org>

Received: from isecom.org (localhost.localdomain [127.0.0.1])

      by isecom.org (8.13.8/8.13.7) with ESMTP id k8UBodHB001194

      for <test@isecom.org>; Sat, 30 Sep 2006 13:50:39 +0200

Received: (from root@localhost)

      by isecom.org (8.13.8/8.13.5/Submit) id k8UBoNcZ001193

      for root; Sat, 30 Sep 2006 13:50:23 +0200

Date: Sat, 30 Sep 2006 13:50:23 +0200

Message-Id: <200609301150.k8UBoNcZ001193@isecom.org>

From: root@isecom.org

To: test@isecom.org

Subject: foobar


test
```

If you look at your raw mailbox, you can sometimes see an additional "From" followed by a space and then a sender address, without the colon seen in the usual "From:" header. That's an internal separator for messages defined by the mbox storage format and it's not really an SMTP header.

The **Mail Delivery Agent (MDA)**, which is the component responsible for storing the message in final delivery, also has the task of protecting any existing line that begins with "From" in the body of the message, a process that's prone to misinterpretation.

The sample message shown above was transmitted with the following SMTP transaction:

```
CONNECT [127.0.0.1]

220 isecom.org ESMTP Sendmail 8.13.8/8.13.7; Sat, 30 Sep 2006 14:08:38 +0200

EHLO isecom.org

250-isecom.org Hello localhost.localdomain [127.0.0.1], pleased to meet you

250-ENHANCEDSTATUSCODES

250-PIPELINING

250-8BITMIME

250-SIZE 5000000

250-DSN
```

```
250-ETRN
250-DELIVERBY
250 HELP
MAIL From:<root@isecom.org> SIZE=57
250 2.1.0 <root@isecom.org>... Sender ok
RCPT To:<test@isecom.org>
DATA
250 2.1.5 <test@isecom.org>... Recipient ok
Received: (from root@localhost)
      by isecom.org (8.13.8/8.13.5/Submit) id k8UC8EMj001346
      for root; Sat, 30 Sep 2006 14:08:14 +0200
Date: Sat, 30 Sep 2006 14:08:14 +0200
Message-Id: <200609301208.k8UC8EMj001346@isecom.org>
From: root@isecom.org
To: test@isecom.org
Subject: foobar

test
.
250 2.0.0 k8UC8c3M001347 Message accepted for delivery
QUIT
221 2.0.0 isecom.org closing connection
```

The path of an email message is traced with the “Received” headers:

```
Delivered-To: <spoofer@isecom.org>
Return-Path: test@isecom.org
Received: from smtp.isecom.org (smtp.isecom.org [140.211.166.183])
        by azzurra.isecom.org (8.13.6/8.13.6) with ESMTP id
k4KL5UOq014773
        (version=TLSv1/SSLv3 cipher=DHE-RSA-AES256-SHA bits=256
verify=NO)
        for <spoofer@isecom.org>; Sat, 20 May 2006 21:05:30 GMT
Received: by smtp.isecom.org (Postfix)
        id D138A64413; Sat, 20 May 2006 21:05:29 +0000 (UTC)
Delivered-To: spoofer@isecom.org
Received: from localhost (localhost [127.0.0.1])
```

```
        by smtp.isecom.org (Postfix) with ESMTP id B87EF64409
        for <spoofer@isecom.org>; Sat, 20 May 2006 21:05:29 +0000
(UTC)
Received: from smtp.isecom.org ([127.0.0.1])
 by localhost (smtp.isecom.org [127.0.0.1]) (amavisd-new, port 10024)
 with ESMTP id 24780-13 for <spoofer@isecom.org>;
 Sat, 20 May 2006 21:05:23 +0000 (UTC)
Received: from mail2.isecom.org (bsiC.pl [83.18.69.210])
        (using TLSv1 with cipher DHE-RSA-AES256-SHA (256/256 bits))
        (No client certificate requested)
        by smtp.isecom.org (Postfix) with ESMTP id 6B37E64405
        for <spoofer@isecom.org>; Sat, 20 May 2006 21:05:23 +0000
(UTC)
Received: from localhost (localhost.isecom.org [127.0.0.1])
        by mail2.isecom.org (Postfix) with ESMTP id BDF11B02DE
        for <spoofer@isecom.org>; Sat, 20 May 2006 23:12:55 +0200
(CEST)
Received: from mail2.isecom.org ([127.0.0.1])
 by localhost ([127.0.0.1]) (amavisd-new, port 10024) with ESMTP
 id 11508-04 for <spoofer@isecom.org>; Sat, 20 May 2006 23:12:42
+0200 (CEST)
Received: from localhost (unknown [192.168.0.5])
        by mail2.isecom.org (Postfix) with ESMTP id 54666B02DC
        for <spoofer@isecom.org>; Sat, 20 May 2006 23:12:41 +0200
(CEST)
Date: Sat, 20 May 2006 23:05:04 +0200
From: John Doe <test@isecom.org>
To: spoofer@isecom.org
```

Dig Me

When you're working in Linux and UNIX in general, **dig** is your best friend for testing DNS settings. MX records are very important to email delivery, so let's have a look at them briefly. MX records are for email, and have no relationship to websites for the same domain. The web server of "domain.com" may be a completely different system from the mail server, which is why those DNS records are identified differently.

The way to get MX records is by using the **dig** command from a UNIX, Linux, or OSX command line. **dig** is a DNS information tool, and as any UNIX program it has a gazillion options. We will just use one format. Using

```
dig <domain name> MX
```

tells `dig` to extract only mail exchange records from the relevant domain. Another easy example is

```
dig <servername> <type>
```

For example the public DNS server 213.133.105.2 ns.second-ns.de can be used for testing. See which server the client receives the answer.

```
dig sleepyowl.net
sleepyowl.net.          600     IN      A       78.31.70.238
;; SERVER: 192.168.51.254#53(192.168.51.254)
```

The local router 192.168.51.254 answered and the response is the A entry. Any entry can be queried and the DNS server can be selected with @:

```
dig MX google.com                   # Get the mail MX entry
dig @127.0.0.1 NS sun.com           # To test the local server
dig @204.97.212.10 NS MX heise.de   # Query an external server
dig AXFR @ns1.xname.org cb.vu       # Get the full zone (zone transfer)
```

The command **`host`** is also powerful.

```
host -t MX cb.vu                    # Get the mail MX entry
host -t NS -T sun.com               # Get the NS record
host -a sleepyowl.net               # Get everything
```

As a larger example, here are the MX records for Google's *gmail.com* domain:

```
;; ANSWER SECTION:
gmail.com.        893   IN    MX    10 alt1.gmail-smtp-in.l.google.com.
gmail.com.        893   IN    MX    40 alt4.gmail-smtp-in.l.google.com.
gmail.com.        893   IN    MX    30 alt3.gmail-smtp-in.l.google.com.
gmail.com.        893   IN    MX    20 alt2.gmail-smtp-in.l.google.com.
gmail.com.        893   IN    MX    5 gmail-smtp-in-v4v6.l.google.com.
```

There are three values in each line that are of interest. The “893” is a **time to live** value (how many seconds, or how many routers to hop, depending) you will find in every DNS record – it indicates how long a DNS is allowed to cache the record before the information is considered stale and has to be retrieved again.

The “10” in the top line, and “40”, “30”, “20” and “5” in subsequent lines are “preference” values, followed by a **Fully Qualified Domain Name (FQDN)** of a system prepared to handle email. The preference values are used by the MTA to decide which of the machines in the MX records list to try first, and which to contact next, should the first one fail or refuse email. If no server is found to accept the email, a failure message is sent back to the email originator (using the “reply-to” or “from” information). Lower values indicate preferred MTAs. Thus, the last entry in the list above will be tried first, with the rest as fallback if the first system fails or is overloaded.

A service can also offer records with identical preferences; here is the response from yahoo.com where the preferences value is set to “1” on all records:

```
;; ANSWER SECTION:

yahoo.com.          48    IN    MX    1 mta6.am0.yahoodns.net.

yahoo.com.          48    IN    MX    1 mta5.am0.yahoodns.net.

yahoo.com.          48    IN    MX    1 mta7.am0.yahoodns.net.
```

Doing this means the email load will be distributed over the 3 systems equally. The very low TTL value of “48” suggests this DNS entry is dynamically controlled, a sign of an active load balancer. Load balancers do pretty much what their name says they do; they make sure traffic (inbound, outbound, high priority, low priority) gets the level of attention it deserves.

Last but not least, you can also identify whether the receiving domain uses mail filtering. The famous domain no10.gsi.gov.uk (the domain of Britain's Prime Minister) shows that a company called MessageLabs is presently responsible for mail filtering:

```
;; ANSWER SECTION:

no10.gsi.gov.uk.   3600  IN    MX    20 cluster.gsi2.messagelabs.com.

no10.gsi.gov.uk.   3600  IN    MX    10 cluster.gsi.messagelabs.com.
```

You do not need to fear black helicopters when you look this up: this information is public, as email would otherwise not work. Besides, the UK military only has *green* helicopters!

Exercises

9.1 Does your email platform support a “Delivery Receipt” or any kind of delivery flag that lets you know (at least) that your mail reached some destination? If it does exchange messages with a friend and examine the headers from that traffic.

9.2 Choose a domain name. Find out which system handles email for that domain by looking up MX records.

Game On: The Bug Trap

The cafeteria floor was slightly damp, almost like fly paper, with a stickiness pulling at the bottom of her rubber soled shoes. Jace glanced at the reflective sheen of the gunk floor wondering how a place that serves food could smell so bad but keep a mirror shine. The odor reminded her of when her Grandpa used to place the cockroach traps behind the apartment couch. As Grandpa pulled out the old trap, Jace could see the encrusted remains. It seemed like the entire inside of the roach trap was filled with dead bugs.

She was never shy about asking questions. “Why don’t the roaches just leave the box? Can’t they see all their friends in there dead?” she asked Grandpa more than once just to be sure she got the right answer each time. Jace loved to watch

Grandpa work, often getting in the way by putting her head right over his shoulder. He never complained. He always loved having his granddaughter right by his side as much as possible.

“Jace, the roaches are attracted to this trap by the smell inside the box. As soon as they go inside the trap, the floor is really sticky and they get stuck in the box. It’s like they are glued to the floor. They don’t seem to notice all the other dead insects inside and they die too,” her grandfather would explain in roughly the same version each time he was asked.

“Roaches aren’t very smart,” little Jace would reply with a smug smile.

“Yes, my dear, you are much smarter than a cockroach.”

“Thank you, I guess.”

Back at the school cafeteria, coffee brown hair fell into her face as she continued to look at the mirror floor; she needed to get back her classwork.

Just out the corner of her right eye, Jace caught a sudden commotion at the cafeteria double doors. They swung open as several people burst into the large open room. Instantly going into clandestine mode, she leaned forward and let her thin shoulder-length hair cover her face. She heard, “There, she’s over there trying to hide herself! Grab Jace. Don’t let her move!” Several older voices shrilled with the excitement of a witch hunt.

The teen hacker held her position at the table, clenching her knapsack and pretending to be unaware of the coming attack. Knives, pitchforks, torches, angry mobs and all those monster movie images compressed themselves into her calculating mind. Curiosity got the best of her and she looked up to see the school principal, his secretary, Mr. Tri, three freshmen table tennis players and several other oddballs approaching. The roar of their voices was tremendous as it bounced against the polished floor towards her.

“Hold on! I said hold on,” a familiar voice commanded somewhere behind the angry freaks. The mob lost momentum. The freshmen parted the crowd so the Chief of Police could step through the cluster of confusion. “Alright folks, thank you for your overzealous help locating Ms. Jace for me. Now, I would like to have a moment alone with her,” the chief said in a calming voice. He'd used that same voice to talk down a jumper off a 14 story building years ago. It worked then, it sort of worked now. The crowd became a loose net of individuals trying to look busy, tying their shoelaces, too obviously stretching to listen to the chief’s private conversation.

“Hi Jace,” Chief couldn’t think of anything else to open with.

“Yes, Police Chief. How may I help you at MY school, while I am enjoying MY lunch. With MY peers all staring at ME,” she almost broke her jaw clenching her teeth.

“I apologize. I’m sorry to interrupt your massive gathering of friends here but I need your help now,” the chief said, trying to keep his cool, but also letting Jace know that this wasn’t the time to be difficult. Jace unclenched her hands from her pack and looked the Chief in the face. He tilted his head towards the double cafeteria doors, motioning her to follow him.

Jace looked down at her unfinished sandwich, reluctant. The Chief didn’t take his eyes off of her. He raised his right hand and slapped his fingers in the air. Jace flinched. The entire lunchroom flinched. Principal Mantral realized what the signal meant and came rushing forward with a clear plastic bag.

“Cookies, eh,” Jace asked.

“Chocolate chip pecan, made by Officer Hank’s wife,” Chief replied. “Deal?”

“Deal,” she replied with one cookie already in her mouth.

As the two walked out of the school, the chief asked if Jace had ever ridden in a SWAT van before. “That was the only vehicle I could get at the last minute. Sorry,” he said. The two of them left the school looking like rock stars in the SWAT van. Jace laughed looking in the rearview mirror at the stunned students and school staff.

“Here’s what is going on. Someone is looking at my email. I don’t know how or who or why but I do know that my emails are being hacked. I need you to help me stop this. It's causing major problems with our law enforcement capabilities,” the Chief didn’t give Jace a chance to interrupt. “When you set up our network last summer, you did a bunch of extra security stuff. It hasn't been enough. I can tell you that one email three weeks ago concerned a technicality on a particular suspect we had. Only me and the District Attorney knew about this problem.”

The police chief reached across the van console to grab a cookie from the open bag. Jace jokingly slapped his hand away. He reached down towards his baton that he didn’t wear since he was a wasn’t a street cop anymore. Jace relented and handed him a large pecan piece to keep him talking, which he did, cookie crumbs littering his uniform.

“Two hours after the DA and I emailed each other, I get a call from the front desk telling me that the suspect just posted bail. The suspect’s lawyer found out about the technicality and got a judge to sign off on release for the suspect. There were only two people who knew about this technicality and that was through my email,” the Chief said.

He continued, “Last week I got a call about the possibility of some missing evidence at a crime scene. That was just an anonymous phone call. There wasn’t any specific item mentioned in that call. I wrote a quick email to our evidence clerk asking for the inventory log for the entire weeks’ worth of cases, especially the log from that day. The clerk emails me the log and I compare it to the police report from the crime. Being the thorough investigator I am, I delete all the information in the log file that isn’t relevant and forwarded the log file to our Internal Investigation team.”

Jace is trying to understand what he is saying in-between all his police jargon. “So?” she blurted out, feeling much better after her sandwich and five cookies.

The chief looked a bit annoyed but answered anyway, “So! So there isn’t any missing evidence that we can tell. Later that day, I get another call from the DA asking me where the murder weapon is from that same case. It didn’t occur to me that the gun was missing from the evidence locker. Again, two hours later another suspect is out on bail because the police and forensic team didn’t document or turn in the pistol used in the crime.”

Jace, wishing she had a large cold glass of milk, chimed in, “So the mysterious caller was checking to see if the gun has in police custody. The email you sent to your Internal Thugs, verified that the weapon was never entered in as police evidence.”

The police chief had an amazing smile of satisfaction when she finished her conclusion. “Ya know, Jace, you’d make a terrific police detective when you get older."

Jace shot back, "Yeah, well, I have too much self-respect to be a cop. I'd rather be a lawyer or a politician or some other lower form of life." Luckily she started to giggle as she said the last part because the chief was getting angry with her insult. "Just joking, Chief."

Game continues...

The Risky Business of Email Composition

- **Disclosure**. Think about whom you are emailing, why and how. Not only is email transmission by default insecure when it leaves the local MTA, you are also releasing information. The use of encryption such as PGP, GPG and S/MIME requires both sides to be similarly equipped, and is generally perceived as very complicated to use (translated: users avoid it with enthusiasm). An alternative way to protect the transmission would be the use of the *same* email provider: that way, the message never needs to travel across the Internet in an unencrypted form. This is where the *how* question appears: are you sure your provider or that of the recipient (or one or the other of your governments) is not listening in? Take that into consideration when you handle something confidential.
- **Rerouting**. An email address does not need to remain in the domain it is sent to, but could be redirected elsewhere. As an example, the US company *pobox.com* does not sell mailbox services, only aliases. The main risk is that your email may thus travel over various, different legal jurisdictions before it arrives at its destination. In our example, a *pobox.com* alias will always go via MTAs in the US first, and is thus at risk of interception under the ongoing abuse of the US PATRIOT Act.
- **Privacy violations**. A recipient using services such as Facebook or Google exposes his email to automated scanning of content, even though the *sender* never gave that permission!
- **Distribution lists**. If you use an email distribution list, use the BCC (blind carbon copy) field for it. Email addresses in the TO: and CC: field are visible to every recipient of the email, and could end up giving away the contents of your email list to an uncontrolled third party, and expose your recipients to spam and other junk mail.
- **Conflict**. An email is like a letter, but is written and sent much more quickly, which leaves you less time to consider its contents. Writing email is like driving: it's best not done in anger. In case of emotional involvement, write a draft and leave it for an hour, then reconsider if you really should send it. It could save a friendship or a career.
- **Misaddressing**. One of the main causes of email going astray is misaddressing. This is a consequence of mail clients trying to autofill an address from the characters typed by the user. Always check if the recipient is indeed the intended one.
- **Multiple recipients**. When you send email to more than one person, make sure the content is appropriate for all recipients. Also, it is good practice and ethically correct to visibly copy someone in if you use their information or talk about them.
- **Legal issues**. Disclaimers under your email may look impressive, but have no legal value other than a copyright notice. You sent the email, so you cannot disavow its content (to a degree, of course you could always claim its sender was spoofed) and

you cannot prescribe what an incorrect recipient should do with an email because you probably don't have a contractual relation with them. (See the Ultimate Disclaimer at the end of this lesson.)

- **Top posting.** When you reply to an email, does your email program automatically put your reply on top of the original message? This seems to be the default these days, but unfortunately it's … rude. Recipients who have to start from your reply and work their way down to some context aren't likely to love you. On the other hand, weren't they in on the original message to begin with? Consider, at least, whether you want to engage in "top posting."
- **Autoresponders.** "You sent me an email so I'm sending you this automatic email response to let you know I won't get your email until I return, so heaven help us both if you've got an autoresponder too, because this is going to go in circles until the end of the universe." This stuff can drive you crazy, but it's also an awfully convenient notice to evildoers that you're probably not at home. What was your street address again?
- **Signatures.** Do you use a signature, an automatic "Yours Truly, Stardonk Cluck, Program Manager for Automated Actions" that sticks itself to the bottom of every message you send? They're not necessarily bad – until they get really long. And ten of them stack up at the bottom of a long back-and-forth conversation. And they are all in HTML, not polite plain text, so that your graphic of a gorilla climbing a skyscraper appears over and over and over. Be kind about using a signature, and don't subject your recipient to the dangers of HTML email at all if you can help it.

Exercises

9.3 Head over to http://www.gaijin.at/en/olsmailheader.php and insert an email header you've taken from any email. This program is an analyzer that will provide you information about that email header. From the information given, what can you do with these results?

Receiving Email

Mail clients contact the servers on which mailboxes are stored, and check if the top message count has not changed. Some clients do this periodically (for instance, every 30 minutes), some do it manually (usually to preserve bandwidth) and some maintain a simple permanent connection with the mail server so that they receive an update as soon as new email arrives (called **push notification**).

When an email is inbound, the mail client or webmail environment will pick this up via POP3 or IMAP. Mobile clients usually download only the header and a small portion of the message to save bandwidth, leaving it up to the user to decide if the whole message should be picked up, left for later or deleted.

The early days of email communication took place over unreliable, slow connections, and the handling of attachment such as documents, spreadsheets or images still shows this. An attachment must always be downloaded in full before it can be displayed. Users using webmail on a third party computer such as in cyber cafes must be careful: **viewing an attachment means leaving a copy behind on the system's hard disk**. By default, those are **not** erased after use.

Webmail on an untrusted computer naturally carries another risk: unless you use one-time passwords, you may leave behind your email access credentials because there is no guarantee the third party machine is not infected or monitored.

Systems with active email should have up-to-date antivirus protection, but you should understand that antivirus only protects against **known** malware. Especially with targeted attacks, it can take several days before malware is added to virus scanners; some malware is never added at all.

Incoming email contains a travel history in the headers. Every system mail passes through adds a line in the hidden part of the header, with the latest one on top. However, be aware this is also easy to forge: keep in mind that not all entries may be real.

Exercise

9.4 Consider the "temporary" files people leave behind when they use email. You can see a lot by looking into temp directories (there are usually more than one). Windows, for instance, makes it easy to see what's in temp directories even if you don't know where they are: the **%temp%** variable knows all of them.

Open a command-line interface in Windows and type

```
dir %temp%
```

What do you see?
For an even handier view, use Windows Explorer by typing this command:

```
explorer %temp%
```

9.5 Open up the email header on any email. See if you can locate any additional receivers besides you. They might be located in the carbon copy (cc) section of the email header.

- Select several emails. Check the mail path and origin via mail headers. Check what other information is available in the headers (hint: email client and antivirus software versions; encryption algorithms; etc.).
- Compare the sender and reply-to addresses
- Have a look at some junk mail. What do you see in those headers compared to normal emails? Check where all the links go (just as text, not by going there). Do the links URLs go where the text says they are going?

Responding to Email

Responding to email needs to be done with some care. How many times have you said something or done something that you didn't mean or wish you could take back?

First of all, NEVER react to what is clearly junk mail, even if it is to unsubscribe. All you do is confirm that (a) the email account is live and (b) someone at that email address actually reads junk mail. The result of the cancellation attempt is thus, ironically, *more* junk mail.

Check for address disclosure. Do all recipients need to be visible? If you use a mailing list, are all recipients still valid? Does every recipient really need to see your answer?

Be hygienic. Does the whole email need to be repeated or can you just use the parts that are relevant? If you re-use parts of a previous email you can show that by "quoting" - a way to make it visible you are repeating part of an email, and then respond to that specific part.

Be careful with quoting: is all that you repeat actually meant for the recipients you are selecting now, or are you including (parts of) a discussion that was confidential and not meant for the new recipient(s)? Avoid quoting the whole original message including signature and (usually extensive) disclaimer. Be aware that whatever you send can be forwarded on to anyone without your permission or knowledge too. It is a good and polite habit to copy a person in when you talk about them or refer to something they have done, so they know what has being said about them, and it prompts you to avoid stating something you may regret later. Delivery flags are helpful in keeping an eye on email as it travels to its destination, and by looking up MX records you can work out where the email will go. You could use geolocation to give you a physical location as well. Delivery flags also chew up bandwidth. Because a delivery flag was set, your email server must send a response. Not everyone appreciates emails flagged as “urgent” or “important.” Flags such as these are usually an indicator that the message is spam, if they weren't sent by a co-worker.

Exercises

9.6 Forward an email to another account and compare headers.

- How can headers be used against you, and how you could prevent this from happening?
- Can you forward an email that was sent to you as a blind copy (BCC)?

9.7 Write yourself an email and send it to yourself. During delivery, quickly retract that email (unsend). If the email was successfully pulled back, take a look at the header of that draft email. Copy that header to a text editor and see if you can locate which email server stopped that email from going through. Cool huh?

Cryptography Protecting Contents From Disclosure

The simplicity of email makes it also vulnerable. The sender cannot be sure that an email is not altered on its way to the recipient, there is no way to make sure that only the receiver can read it and a receiver cannot be sure it is actually sent by the person listed in the email as the sender.

One way to ensure confidentiality is to encrypt a document before attaching it to an email. For example, it is possible to encrypt text documents and spreadsheets like those produced by OpenOffice, and PDF files support encryption too. However, applying cryptography to email itself is easier, and also allows for the email contents to be secured.

Email headers still need to be left in cleartext so that the mail servers can process and deliver the email.

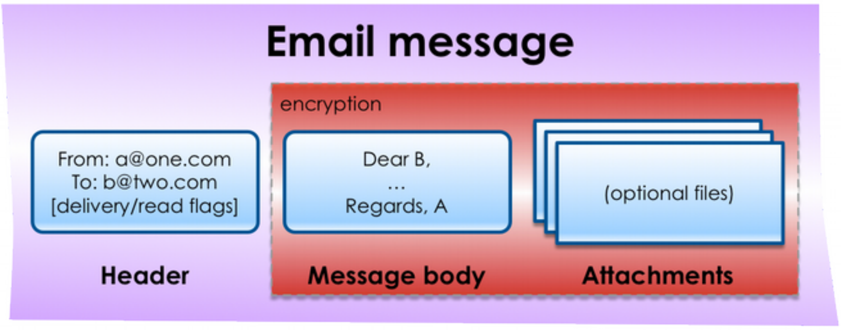


Figure 9.3: Email encryption

Email security can be provided in two different ways: using PGP (or GPG) and S/MIME. Both use encryption to assure:

– **C**onfidentiality: can only the intended recipient(s) read this email?

– **I**ntegrity: is the email content unchanged?

– **A**uthenticity: did the email really come from a particular sender?

(An easy way to remember this list is the acronym they form: **CIA**.)

In general, authenticity and integrity are combined in electronically **signing** an email: the email gets checksummed, and the result is encrypted and embedded in an electronic signature that could only have been made by the person who holds the right private key (see "PGP and GPG" below).

Confidentiality is assured by using someone's public key to encrypt the message body, so that only the holder of the correct private key can decrypt and read it (see "PGP and GPG" below). For extra assurance, such a message can be signed too.

You should keep in mind that encrypted email is rather uncommon, especially in an era where people voluntarily let their email be scanned by companies such as Google and Facebook. In some countries, you must make sure that you have the means to access your email when authorities demand this of you, for instance in the US by the TSA when you cross the border, and in the UK when served with a warrant under the Regulation of Investigative Powers Act.

PGP and GPG

PGP stands for Pretty Good Privacy and was developed by Phil Zimmermann. The history of PGP is interesting and certainly worth reading, but for the purposes of this chapter we will only focus on its use.

You are more likely to come across the Open Source version called GPG (GNU Privacy Guard). GPG is available for free for many platforms, and only uses open, publicly evaluated algorithms.

GPG works on the principle of **public/private key management**, which means that keys have a PUBLIC part you can give to anyone who wants to send you encrypted email, and a PRIVATE part you have to keep secret, which is the only way to decipher the message you have received. The combination of private and public key is called a **key pair**, and it is generally the first thing you generate when you install GPG on a machine. The key pair is protected by a password so that it cannot be altered by anyone but the owner. Alterations may be necessary because you want to change the email addresses the key supports, or want to make use of other functions.

Because you need someone's public key to encrypt a message to them, servers such as pgp.mit.edu exist where you can download the key or keys associated with a specific email address and upload your own. It is possible that keys have expired or passwords have been lost, so always use the latest key or even better, ask your recipient to send theirs and confirm the key fingerprint (a short version of the checksum).

MIME Your Manners

MIME (Multi-Purpose Internet Mail Extensions) is an email extension of the Simple Mail Transfer Protocol (SMTP). MIME gives you the ability to transfer different types of media and data like audio, video, images, compressed files, and applications as attachments to email. The MIME header is inserted at the beginning of the email and the receiving email client uses this information to determine which program is associated with the attached file. MIME in itself does not provide any security to emails or attachments.

S/MIME (Secure/Multipurpose Internet Mail Extensions) is a protocol that adds digital signatures and encryption to Internet MIME message attachments. Using digital signatures, S/MIME allows for authentication, message integrity and **non-repudiation** of origin ("non-repudiation" means you can't deny you sent it). S/MIME provides privacy and data security (using encryption) to emails that use this protocol.

S/MIME is both a security tool and a security issue since users can send sensitive data or secrets as attachments to outbound emails in order to avoid detection. Therefore, the use of S/MIME in a corporate setting should be carefully monitored on the email servers.

Key Trust

How do you ensure that a key for an email recipient is really theirs, and not uploaded by someone else? The solution to this is that keys can be signed by others. Imagine you already have the key of someone else who you trust, and who knows the person you want to email with. That other person can **sign** the public key, which means you invest a bit more trust in the key, provided you know this other person. This is known as **inherited trust**. You can also find another way to get in touch with the person and either receive their full public key, or receive the key "fingerprint" - a checksum of the key which is quick to verify. On a key server, a key can also have an ID – yet another checksum serving the same goal.

Sending An Encrypted Email Using GPG

Most email clients support plugins that make the handling of keys and encryption easier. The best thing to do is to check beforehand if your recipient has a public key and get it from a key server, or from the recipient themself.

Then, compose your email as normal (once again we strongly recommend plain text email over HTML), add any attachments and tell your email client to encrypt and send the email. If you decided to sign the email, the email client will use your private key to sign the message first, then use the public key of your recipient to encrypt the email and any attachments. If you protect your key pair with a passphrase (you'd better!), your email client will ask for that passphrase.

Receiving An Encrypted Email Using GPG

Email encrypted with GPG contains either an attachment flagged as GPG, or has a block of text with a header that tells a GPG-capable email client it has just received an encrypted message. The email client will now access your private key (possibly via a password) and decrypt the message and any attachments. If the message was not encrypted with your public key this decryption will simply fail. If the message was signed by the sender, the GPG plugin will use the corresponding public key to verify that signature too.

GPG plugins will alert you to problems with signatures or attachments, but in general you will find that once installed, the use of GPG is quite easy.

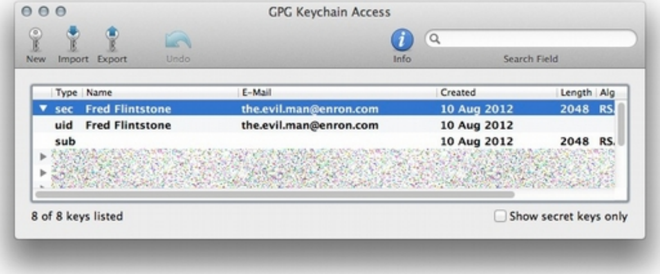
GPG Implications

Be aware that the majority of email is not encrypted, and that probably includes your own. Some people think that using encryption is suspicious and draws attention all by itself. It's your right to have privacy when communicating, so don't worry too much about the opinions of others.

GPG is not easy to use in a webmail environment (aside from the obvious question whether you can trust a third party to encrypt properly and not mount a man in the middle attack on your secrecy) and also doesn't work well on mobile clients. Be wary of mobile apps that claim to solve this issue: some have been found to send your data elsewhere for processing!

> There are online mail services that sell encrypted, "enhanced security" email accounts. But be careful to read the fine print on your user agreement. One provider's reads like this:
>
> "I understand that this service is not suitable for illegal activity and that the providers of this service will cooperate fully with authorities pursuing evidence via valid legal channels."
>
> Of course, "legal channels" include programs like Echelon, Carnivore, PRISM, the Patriot Act and XKeyscore. Look 'em up and ask yourself just how "enhanced" you think this paid "security" really is.

Some countries mandate that you must be able to decrypt any information when ordered to do so by court. For example, in the UK you can be served under the Regulation of Power Act 2000, and non-compliance is termed as contempt of court, automatically resulting in jail time. This has unpleasant implications: if you have been experimenting with encryption and have forgotten the keys or passwords, you will effectively face jail for being forgetful (yes, you will be guilty until you can prove your innocence). It is thus good practice to erase any encrypted material and email that you can no longer access. In a corporate setting you must manage and document key and passphrase changes and disposal of encrypted information very carefully.



Type	Name	E-Mail	Created	Length	Alg
▼ sec	Fred Flintstone	the.evil.man@enron.com	10 Aug 2012	2048	RS.
uid	Fred Flintstone	the.evil.man@enron.com	10 Aug 2012		
sub			10 Aug 2012	2048	RS.
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

8 of 8 keys listed

□ Show secret keys only

Figure 9.4: The GPG Keychain

Last but not least, it is interesting to note that the use of email addresses to identify a key is a **convention**, not a mandated standard. It is entirely possible to generate and use keys for email addresses that do not exist. Such keys are still accepted at public servers. This is known as "hiding in plain sight," and it means that there is no relation between the email addresses and the key used to encrypt/decrypt traffic. In the above image, for instance, both person and email address are fictitious.

The disadvantage of doing this is that it breaks an established way of working, and plugins such as **Enigmail** may need some convincing before they support this more creative approach. A further area for experimentation is expired keys: expiration doesn't stop the keys from working.

Exercises

9.8 Download the GPG support for your email client program and install it.

9.9 Find out how to generate your own key. Do so. Keep it local; don't accept any offers to upload your public key to a public key management server.

9.10 Add other email addresses to your key, then change the passphrase.

9.11 Now publish your public key to a key management server.

9.12 Compose an email to someone using GPG. How would you go about getting their key? Do it.

9.13 What can you do with messages when you only have your own key?

9.14 Create a new key for a fake email address. How easy is it to do on your machine?

Email Server-Side Vulnerabilities and Threats

Both small and large organizations use email servers to send and receive these electronic messages, unless they outsource the task or use a cloud service. Email serves multiple purposes to its users: some are good purposes and some are evil (hear madman laughing in the distance). Email servers are the first line of attack/defense on a network perimeter.

Emails have been used to send family vacation pictures, piano recital songs, birthday cards, have a bad day cards, homework assignments, excuses for not turning homework, company communications, marketing, newsletters and an assortment of other media. Besides the "Have a Bad Day" email, all the emails mentioned above have a friendly use. Email serves a valuable purpose for daily communications.

On the other hand, emails have been exploited to send out flame letters, porn, pirated MP3s, classified information, corporate research secrets, taunts, cyber bullying, malware, phishing, and spam. In 2012, email attachments moved to second place behind rogue web sites as the primary delivery tool for malware. A vital part of our communal society has become perverted for criminal use.

Bandwidth Eating

Email servers should be configured to block the bad stuff and allow the good stuff to pass to you. This sounds easy enough and it will be easy for us to tell you about it. You'll have all the hard work of making it happen (again, mad scientist laughing in the distance). All email traffic coming and going through a network eats up vital bandwidth. You will never hear someone complain, "My connection is too fast." The sooner you can detect and inspect email traffic (outbound and especially inbound) on your email server, the less bandwidth is wasted. Besides preserving bandwidth, the ability to filter bad emails early on will save work on CPU server processing.

Some studies estimate that 80% of all inbound email is spam. Do you really want to wait until that junk gets into your email inbox before this stuff is detected and deleted? The sooner spam is intercepted by your email servers, the better. One technique used is when spam is detected, the server will eliminate it after a certain amount of time. This prevents deletion of email traffic that a user might be expecting. Your organization's marketing department might really want that one email with the subject line "How to improve your performance." Turn off automatic email confirmations and receipts to save bandwidth on the email server, too. Your users won't mind, trust us.

Since your email servers are access points exposed to attacks coming from the Internet, you should take extra precautions for anyone with admin rights. Those who have admin rights should never send or receive email while they are logged in with admin privileges. In fact, those with admin rights should only use those rights for internal network maintenance. Over the years, many networks have been compromised when an admin logged in and surfed the Internet or sent emails while working with escalated privileges.

Email Server Vulnerabilities

As the name might suggest, an email server is just like any other server. The server will have vulnerabilities that can be exploited. The Common Vulnerabilities and Enumeration database at http://cve.mitre.org listed a total of 1043 email server vulnerabilities in 2012. Many of these issues can be resolved through proper server configuration and user privileges. Other issues can only be solved by the software manufacture or by being vigilant when shopping for server software.

For a complete list of all known email server vulnerabilities go to http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=email+server.

Email Server Threats

Large web email clients like Gmail, Yahoo, and Microsoft migrated to a new cryptographic email signature program called **DomainKeys Identified Mail (DKIM)**. DKIM wraps a cryptographic signature around an email that verifies the domain name that the message was sent through. DKIM helps filter out spoofed messages from legitimate ones. The specifications for DKIM can be found at http://www.dkim.org/.

The problem involves DKIM test messages. According to **US-CERT (United States Computer Emergency Readiness Team)**, an evil hacker can send a flag that it is testing DKIM in messages. Some recipients will "accept DKIM messages in testing mode when the messages should be treated as if they were not DKIM signed.”

This isn't the first DKIM problem that has attracted CERN's attention. The signature key length used for encryption was vulnerable to cracking if the key size was too small. DKIM standards set the minimum key size to 1024, with any email using a smaller key being rejected by the program. But DKIM operations didn't reject smaller key sized emails. Instead, the emails were sent along their merry way, fully vulnerable to factor cracking. Once the key was cracked, a hacker could spoof emails or send out malware using that user's email key and address.

DKIM is designed to act as a “trust” verification tool for email. The system uses public-key cryptography, just like PGP does. With proper use, an email can be traced back to its original sender through a domain verification process. Basically, you are identifying yourself as the sender by your domain origin. This should drastically reduce spoofed emails, filter spam, and prove that you sent that message. Security folks call this non-repudiation.

In non-repudiation, the information provider data cannot be changed. The information is not refutable. If you said, “I want to wear a dress,” that statement cannot be contested. You said it, that is a fact and you will not be able to retract that statement. This is important when dealing with contracts, legal matter and excuses to your father for not taking out the trash.

Email for Fun and Profit

Thanks to the profitable market for corporate espionage, email is a simple method to find client contact lists, customer information, meeting notes, new product developments, answers to the next math test and all kinds of valuable data. We are not even going to get into government espionage, simply because we all know this has been taking place since the dawn of man. There are several primitive cave paintings depicting one caveman spying on another caveman's woolly mammoth with envy. One can only imagine the spy caveman returning to his tribe to describe the newest version of Mammoth 2.0.

A simple but often missed email security method is scanning of all email attachments. Scanning needs to be performed on all data packets, compressed files, unknown file types, split files, files that can do the splits, files that spit, meta data, files with URLs, and pretty much everything that can be done to a file. This scanning should be focused on inbound traffic but don't forget to be suspicious when large attachments are leaving your network. Sensitive company information needs to be encrypted, especially if sent by email, to anyone inside or outside the network. Oh, by the way, sensitive information really should never leave the network. If a user is sending information outside the network, you might want to keep an eye on their activities.

Large organizations like the Veterans Affairs Hospitals in the US use **data loss prevention (DLP)** software to do all this stuff. Never attempt to playfully email your ex's medical records from the VA Hospital, for instance, since the people who come play with you will make you wish you had just cut your throat and gone *straight* to … well, you know where.

The Key to Success

Keyword filtering is a type of application layer filtering (layer 7) that lets you block all messages containing particular keywords or phrases (text strings) that commonly appear in spam (for instance, “Viagra” or “hot sexy babes”). Other forms of email filtering include:

- **Address blocking**: a filtering method that blocks mail from particular IP addresses, email addresses or domains of known spammers.
- **Bayesian filtering**: “intelligent” software that can analyze spam messages and learn to recognize other messages as spam using **heuristics** (patterns of behavior).
- **Blacklisting**: lists of known spammers’ addresses can be shared, so each user doesn’t have to develop a list from scratch. These lists are available from several providers, and are highly valuable for address blocking.
- **Whitelisting**: a filtering method that, instead of specifying which senders should be blocked, specifies which senders should be allowed. Again, these lists are used as part of address blocking.
- **Greylisting**: this temporarily blocks email from unknown sources. Legitimate email will be re-transmitted, but spam usually won't.
- **Challenge/Response filtering**: replies to email from senders not on a “trusted senders” list with a challenge, usually involving solving a task that is easy for humans but difficult for automated bots or scripts.

There are many open-source and for-pay applications that can do these kinds of filtering, some better and some worse. If you've ever had to deal with these PITAs (we'll let you figure out on your own what a PITA is) you'll see them as the challenges to clever script writers that they are.

Email Client-Side Vulnerabilities and Threats

Incoming email may contain malware, usually in the form or an attachment or a web link. When you see these clues think "Scam!"

- An unexpected origin: who sent you this mail, and would they have sent it? A favorite trick of spammers is to use other people's valid email address as the origin so spam passes filters and users are more likely to open the email.
- A "too good to be true" event such as a lottery win, inheritance or bank "mistakes" in your favor. Look up the "Nigerian scam." Do any of the sample messages you'll find look familiar? Practically everyone has gotten one.
- A domain mismatch between "from" and "reply-to" addresses (compare them).
- Weird, strangely incorrect or over-complex use of language.
- Unexplained or illogical urgency. (Why would this email be so urgent?)
- Embedded web links which go to different domains than the human readable text suggests (e.g. a link that appears to go to *www.bank.com* in reality goes to *www.l33thacker.org* with a fake banking site). Most mail clients now show the real website address when a mouse cursor is hovered over the relevant text.
- Attachments with active content, such as .exe or .html. These are especially risky on platforms that auto-execute content.

Exercise

9.15 Go to http://www.419eater.com/. What is scam baiting? Can you find instructions? Can you find *precautions*? This is dangerous stuff. Knowing about it doesn't mean you should do it. But you shouldn't be defenseless either.

Turn On The Lights

Email content is a wonderful way to get users to click on malicious links. One common tool is the **Blackhole Exploit Kit**. Sounds scary, doesn't it! Can you say "Blackhole Exploit Kit" five times really quick without making any mistakes? Blackhole is a web application exploitation program that takes advantage of known vulnerabilities in Java and Adobe applications. It's used to send a phishing email to users, trying to get the user to click on a compromised web link.

Phishing is an attempt to gather important information from a victim by using **social engineering**, persuasive emails sent out to thousands of users. Typical phishing emails appear to come from a well-known and trusted organization. The attackers will use the exact same logo, similar reply email address, and as much professional wording as they can to fool as many people as they can. The email will ask the reader to "verify" or "update" credit card data, personal bank account information and other stuff that you would only give to a trusted source.

When the victim clicks on the official-looking link, the link sends them to a compromised page where the malware payload is installed on their machine. The user is unaware they're being **pwned**, and not all anti-virus programs detect the installation. Once the payload is on the user's computer, the phishing spam senders now control that computer as well as any content they want to collect from it.

Blackhole spam pretends to come from legitimate companies like Amazon, Visa, Twitter, UPS and many organizations that wouldn't raise a user's suspicion. This program is rented by paying for server time on the Blackhole server. Recent fees ranged from $50 for a day to $150 a month.

Malware, Trojans, And Rootkits, Oh My

The media loves to play up email cracking, because fear sells. (Remember that when anyone who's selling you something tries to scare you.) But the truth is, malware has been around a long, long time.

Dr. Fred Cohen wrote his PhD dissertation on the idea of a computer virus in 1984, published it in 1985, and it was yanked from public viewing a few weeks later. 1985 was a long time ago and the media still acts like malware is a massive new threat to the entire world.

We are not going to tell you every transmittable threat; you can look them up yourself in Lesson 6, Malware. We are going to show you how email security works from both the inside and the outside.

This Email Looks Legitimate, Let's Open It Up

STOP!!!!! Don't open that email just yet. In fact, don't even preview the email. There are several ways email has been and is still used as an attack tool. Social engineering is the preeminent technique to get people to open email, open attachments or click on malicious web links inside an email or message. Social engineering preys on several human emotions we all have including curiosity, wanting to be helpful, trust in our friends, greed and many types of financial or medical concerns. See Lesson 20 for much more about social engineering.

Our curiosity with new or unknown information can be used to persuade us to do stupid things. When you're sent an email that has “Subject: Re: Re: Thanks!” if you're a typical user, you'll want to know why you're being thanked. In this case, you are being thanked ahead of time for opening up an email with a malicious payload.

These types of emails ask you to call a phone number, click on a URL in the message or do anything that can give away your goodies.

Exercises

Consider an email with this at the top:

from: Mr Norman Chan <naveen.kumar@iitg.ac.in>

reply-to: 2259575299@qq.com

to: (your email address)

date: Mon, Nov 19, 2012 at 7:40 AM

mailed-by: iitg.ac.in

9.16 Would you respond to an email that has this in its subject line: “Hello,I'm Norman Chan,i have a bussiness worth 47.1M USD for you to handle with me?” The sender is “naveen.kumar@iitg.ac.in.”

9.17 Investigate this email address to see if Norman Chan owns a business worth 47.1M. Also check the reply-to address, “2259575299@qq.com.” DO NOT GO TO QQ.COM.

9.18 Max out your browser's security settings before attempting this. Do a little research into qq.com but do not go to this URL. DO NOT OPEN THIS URL. Based on your research of qq.com, would this site raise any alarms for you?

Exciting Tricks With Email Systems (Hacking the Postman)

Email seems to always play some part when it comes to a security breach or a huge network attack. Every virus, every bit of malware, every phishing event seems to involve email as either a major transport mechanism or a way to enter a system to begin an attack. Email may not be as popular as other communication forms like SMS or IM, but it's what's used the most in the corporate and government world. Now we're going to look closely at the very idea of email and how it can be used as a weapon or a shield.

When mapping a network, we need to know several entry points. If we rely on just one entry point, what we do we do when that vulnerability gets patched or fixed? Multiple entry points to a network allow us more freedom to move around that network and give us more avenues of approach. Avenues of approach are a good thing, trust us.

Knowing the user name scheme used by an organization for email or network access gives us a major advantage. Once we know a user name, we can then focus on obtaining that user's password. Most, but not all, organizations use “firstname.last name@companyname.com. Some use variations of first initial and lastname@ companyname.com. Others use the last name followed by the first initial @companyname.com. Pretty lame, right? How much more lame is it when the email address is also the user's login name? This mistake is far too common.

Organizations have a directory within their network that allows users to know who's who and where they work or what they do. That internal directory is a gold mine of information for an attacker. You can look up people in Facebook or other social media to learn more about each user. You can find out when they're going on vacation, what they do, what their hobbies are and other clues to the types of passwords they would use. This information is also valuable if you want to pursue social engineering against that person (for fun and profit).

SEAK And Ye Shall Find

Let's look at gathering email addresses and using email as a hacking tool. Email hacking is closely linked with social engineering, just in case we haven't pointed that out enough already. The **Social Engineering Automation Kit (SEAK)** at http://www.seak.com.ar/ is designed to use search engines to locate email addresses in a network or on a web site. SEAK is basically a set of Perl scripts that allow search engines to look deep into web pages and networks then report back all the email addresses it finds. SEAK can also be used to locate people in the same way.

SEAK has a brother program at https://github.com/FreedomCoder/ESearchy-ng, called **Esearchy**. We didn't name it so don't blame us for the strange name. Esearchy does the same things as SEAK but does it in a Windows environment and searches documents too. Esearchy looks for passwords hidden in metadata, along with any other useful information like email addresses that are available to the public.

Another tool, **Maltego**, is an open-source intelligence and forensics analyzer. It provides tools for discovering data from open sources, and showing that information as a graph, which is handy for link analysis and data mining. The whole point is analyzing real-world relationships between people, groups, websites, domains, networks and online services like your favorite social networks.

An easy tool that you may have heard of is Google search. If you want to see all the employee profile information for a company you can use this command:

```
site:www.google.com intitle:"Google Profile" "Companies I've worked for"
"at company_name"
```

If you want to search all email addresses in a domain or URL, you can use Esearchy. You'd type the following all on one line, substituting a real domain for "company."

```
esearchy -q"@company" -y
AwgiZ8rV34Ejo9hDAsmE925sNwU0iwXoFxBSEky8wu1viJqXjwyPP7No9DYdCaUW28y0.i8p
yTh4 -b 220E2E31383CA320FF7E022ABBB8B9959F3C0CFE --enable-bing --enable-
google --enable-yahoo --enable-pgp -m 500
```

Gpscan is a Ruby application that can automate this search and produce even more results. When paired with the command above, Gpscan becomes a powerful script tool for reconnaissance and social engineering. You can find Gpscan at http://www.digininja.org/projects/gpscan.php.

While you're using any of these, take the time to learn how these tools work. Pay particular attention to the syntax available for each tool and what each one does. You can learn quite a lot about how search engines can be used to hunt for and return email addresses, and possibly a few passwords. Also know which engines are used for the job. Some of the best search engines on the Internet are difficult to find.

Exercises

9.19 Now it's time for you to research a security tool on your own. Find **FOCA** (the metadata tool). What does it do? Would you want it as an arrow In your quiver?

Spoofing Versus Malware

In 2007, the CEO of a Fortune 500 company received an email that appeared to be from one of his senior staff. The email's **From:** line showed that the email was sent by a close associate. The **Subject:** line said something like, "How to reduce energy costs." When the CEO opened the email he saw an attachment and a link that also appeared to be legitimate. The CEO opened the attachment and didn't see anything on his screen so he closed the email.

Several months later, the FBI met with that same CEO to tell him that several terabytes of data had been stolen from his company due to a malware infection from the CEO's machine. The FBI confirmed which email had the malware attachment and that "How to reduce energy costs" was the culprit. The message had been spoofed.

This situation happens every day. Your uncle will call you and ask, "Why are you sending me so many email ads?" At school, your buddy keeps getting spam from you for useless products. Why are you sending out all these spam messages?!

You're not.

Your email address has either been spoofed or your computer email program has been hacked. To find out whether an email address was spoofed, you will need to look at the email header of the sent email. We talked about this already. Now let's put your knowledge to work.

Ask anyone who received a spam email from you to FWD it to you, whole, not just by quoting the message. The header will tell you if your email address was spoofed or not. Look at the **Reply** and **Sent** portion of the email. As we already saw in previous exercises, the header will show that the email was either sent by you - or by someone else.

Stupid Email Tricks

When it comes to privacy, web based email is anything but private. The web service can be asked to provide all of your emails, contacts, calendars and such, if presented with legal documents showing a cause to release that data. One old but still useful trick is to create a web email account using some name other than your own. Whoever you are sending secret emails to has the same access to the account as you. You create your email and save it as a draft. You never send the email, just create a draft of the contents. The email stays in your account but is not traceable since it was never sent. Your partner logs into the same account and read the draft email. Once read, the draft can be deleted or altered to create a new draft email for you to read. It's like Ping-Pong without a ball. The same can be done with a shared Google Doc, by the way.

> **NOTE:** You don't get to be the director of the United States Central Intelligence Agency without knowing this little trick.

The email meta data is actually outlined in RFC 2822. Someone thought that it was a good idea to include meta data in email! What medication were they on when they thought this one up? The email meta data can include the following information:

- To
- From
- CC
- BCC
- Date
- Subject
- Sender
- Received
- Message-ID
- References
- Resent
- Return-Path
- Time/Date
- Encrypted
- In-Reply-To

Okay, don't worry too much about RFC 2822 on email meta data. The reason is, the RFC only covers electronic text traffic. Your SMS texts, all your Instant messages, and those photos you shared contain this wonderful invasion of your privacy, as well. However, this hidden data is covered under a different RFC. Listen to the sound of the evil scientist laughing once again in the background. “Wha ha, ha, ha, snort, ha, ha, ha, snort, ha, ha ha - Igor get me a tissue.”

Outsmarting The Email Bots (Email Obfuscation)

This is so simple that you will laugh at yourself for not thinking of this earlier.

When you need to send your email address to someone else or to some account or some other crazy reason; are you going to send it in plain text? If you do, you're opening yourself up to spambots. These spambots are vicious creatures that have mother issues and they slime their way through the Internet looking for email addresses.

It's like connecting a brand new computer to the Internet without enabling any security features: An email address sent in plain text is just asking for trouble. You might as well connect a J-2 crossover to the Viper 115 ROJ input module inside a .002 latent relay link with a Blast Ion K-0a801 inside a quad reverberator! How messed up is that! Bad idea, very bad idea.

To outsmart these, you might want to try altering your email address when you send it. As with anything else, there is a trade-off between ease of use and security. There are several techniques, just use your imagination.

```
somebodyatsome.whereelse
```

```
Somebody@somedotwhereelse
```

```
somebody2some.whereelse
```

These have been used successfully to pass on an email address and bypass those weak bots. We'll see how long that lasts.

Exercises

9.20 Visit this URL to take a look at Etherios EasyDescribe:
http://appexchange.salesforce.com/listingDetail?listingId=a0N300000018leZEAQ

This is a free metadata viewer/extractor. Grab some of those emails you have stashed away and run them through Etherios. What metadata is in those emails that is not in the header?

9.21 If some data is not in the header, where could that metadata be hidden in the email?

9.22 Does RFC 2822 require metadata to be embedded in electronic text or is it just a standardized method for Internet email traffic?

9.23 Using whatever means you think is best, try and locate the correct business email address of the three company CEOs listed below. Hint, first find out who they are.

Coca Cola
Kia
British Aerospace Engineering (BAE)

Conclusion

Now that you are thoroughly educated (or confused) about email, you can quickly see that this simple communication tool is not very simple at all. The way email works through systems can become quite complex and requires you to meet certain criteria. Remember how we sent you as an email through the typical send, route and receive process. You did pretty well, too. You might want to consider a career as an email. Just a thought.

Email etiquette is important since writing and sending off an email written when you were angry or upset could cause you trouble later on. Stop sending everyone CC'd replies. If you are going to respond to a bunch of folks, use BCC to protect others' email address privacy.

Along the same lines of protecting privacy, we discussed using encryption tools like PGP or GPG to send and receive emails away from prying eyes. The cool part of that section was actually creating a key to use for real. It wasn't all that painful was it? If you answered "yes," sorry. We are certain that the local fast food restaurant is still hiring, since security just isn't your thing. We hope that you didn't answer "yes," because we need as many security experts as we can get.

After that section, we went into heavy security topics. Ok, maybe they weren't all that hard core but we thought you would enjoy some of it. You have to admit that email server and client-side vulnerabilities and threats were fun! Well, we had fun writing it. We got to look at spam, spam and more spam. We know that it eats up valuable bandwidth so you need to filter it as soon as possible in the routing chain. We also know that the state of Hawaii consumes more real Spam annually than any other state or country. They like their Spam.

Dig is an important email tool for Linux and Unix users for tracking down specific information. If you see large chunks of data leaving your network as email attachments, you know to take a closer look to ensure company secrets aren't part of those emails. One interesting point we discussed was the use of Blackhole server to exploit vulnerabilities within networks. This tool was been widely used for sending malware in email, knowing that someone in the domain is likely to open that email or click on the infected link within that electronic document. This kind of threat can be stopped by filtering email traffic and educating users on this issue. User education is a key element in security.

At this point in a conclusion you would get some advice that may or may not interest you. Not here. Just know that email security is a challenge to cyber security. How you view that fact depends on which side of the fence you're on.

The Ultimate Disclaimer

Contributor Peter Houppermans, author of *The Evil Guide to Privacy*, writes:

If you ever feel the need to highlight the folly of email disclaimers, please find here one Peter Houppermans cobbled together from very old USENET posts together with some of his own. A critical part of being a successful hacker is having a functional sense of humor: not only does it make you unpredictable, it also provides armor against those times when IT gets the better of you. Laugh, dust yourself off and rejoin the fight. Computers cannot win; we'll unplug them.

The opinions expressed herein are those of the author, do not necessarily represent those of his employer or someone else, are probably silly, and have nothing to do with the recent consumption of liquor to the value of the GDP of a small country.

The information in this email and any attachments is purely nonsensical and unlikely to be politically correct. It is intended solely for the attention of world + dog and its mother. I sincerely believe it's totally pointless to tell you what to do with this email if I managed to send it to the wrong place. This email does not contain nudity (yet), and no cuddly animals or whales were hurt during its production as there weren't any available. In true consulting style this email has been composed entirely from recycled keystrokes and cut-n-paste from many, many other leaving emails and other versions to other audiences (minus the embarrassing parts - I take PayPal, hint). May harm the digestive system if swallowed (especially when printed on cardboard), batteries not included. Author may sue, contents may settle. For more culture, add yoghurt. Unsuitable for people under 18 or those lacking an operable sense of humour. Do not hold upside down, open other end. If you received this email in error, well done.

Any attachments should not be trusted or relied upon, but may prove highly entertaining.

This disclaimer is meant for educational purposes only. Send no money now. Ask your doctor or pharmacist. To prevent electric shock, do not open back panel. No user serviceable parts inside. You may or may not have additional rights which may vary from country to country. Not recommended for children under twelve years of age. Batteries not included. Limit 1 per customer. Does not come with any other figures. Any resemblance to real persons, living or dead, is purely coincidental. Keep away from open flame or spark. Void where prohibited. Some assembly required. All rights reserved. List each check separately by bank number. Contents may settle during shipment. Use only as directed. Parental discretion advised. No other warranty expressed or implied. Unauthorized copying of this signature strictly prohibited. Do not read while operating a motor vehicle or heavy equipment. Postage will be paid by addressee. In case of eye contact, flush with water. Subject to approval. This is not an offer to sell securities. Apply only to affected area. May be too intense for some viewers. Do not fold, spindle, or mutilate. Use other side for additional listings. For recreational use only. Shipping and handling extra. No animals were harmed in the production of this signature. Do not disturb. All models over 18 years of age. If condition persists, consult your physician. Freshest if consumed before date on carton. Prices subject to change without notice. Times approximate. No postage necessary if mailed in Singapore. If swallowed, do not induce vomiting. Breaking seal constitutes acceptance of agreement. For off-road use only. As seen on TV. We reserve the right to limit quantities. One size fits all. Do not leave funds without

> *collecting a receipt. Many suitcases look alike. Contains a substantial amount of non-active ingredients. Colours may, in time, fade. We have sent the forms which seem to be right for you. Slippery when wet. This product is only warranted to the original retail purchaser or gift recipient. For office use only. Net weight before cooking. Not affiliated with the Red Cross. Surfaces should be clean of paint, grease, dirt, etc. Drop in any mailbox. Edited for television. Keep cool; process promptly. $2.98/min AE/V/MC. Post office will not deliver without postage. Simulated picture. List was current at time of printing. Penalty for private use. Return to sender, no forwarding order on file, unable to forward. Do not expose to direct sunlight. Not responsible for direct, indirect, incidental, or consequential damages resulting from any defect, error, or failure to perform. No Canadian coins. Do not puncture or incinerate empty container. See label for sequence. Prices subject to change without notice. Do not write below this line. Time lock safe, clerk cannot open. At participating locations only. Serial numbers must be visible. Align parts carefully, then bond. Falling rock zone. Keep out of reach of children. Lost ticket pays maximum rate. Your cancelled check is your receipt. Check paper path. Place stamp here. Avoid contact with skin. Sanitized for your protection. Be sure each item is properly endorsed. Penalty for early withdrawal. Sign here without admitting guilt. No solicitors. Slightly higher west of the Mississippi. Storage temperature: -30 C (-22 F) to 40 C (104 F). Employees and their families are not eligible. Beware of dog. Contestants have been briefed on some questions before the show. No purchase necessary. Limited time offer, call now to ensure prompt delivery. You must be present to win. No passes accepted for this engagement. Extinguish all pilot lights. Processed at location stamped in code at top of carton. Shading within a signature may occur. Use only in well-ventilated areas. Replace with same type. Accessories sold separately. Booths for two or more. Check here if tax deductible. Keep away from fire or flame. Some equipment shown is optional. Price does not include taxes. Hard hat area. Pre-recorded for this time zone. Reproduction strictly prohibited. Adults 18 and over only. Detach and keep for your reference. No alcohol, dogs, or horses. Demo package, not for resale. List at least two alternate dates. First pull up, then pull down. Call toll free before deciding. Driver does not carry cash. Some of the trademarks mentioned in this product appear for identification purposes only. Record additional transactions on back of previous stub. This supersedes all previous notices. Tag not to be removed under penalty of law.*

If you've read all the way to here you're either exceptionally curious. Or a lawyer.

Today's teens are in a world with major communication and productivity channels open to them and they don't have the knowledge to defend themselves against the fraud, identity theft, privacy leaks and other attacks made against them just for using the Internet. This is the reason for Hacker Highschool.

The Hacker Highschool project is the development of security and privacy awareness learning materials for junior high and high school students.

Hacker Highschool is a set of lessons and a practical means of making hackers. Beyond just providing cybersecurity awareness and critical Internet skills, we need to teach the young people of today how to be resourceful, creative, and logical, traits synonymous with hackers. The program contains free security and privacy awareness teaching materials and back-end support for teachers of accredited junior high, high schools, and home schooling. There are multiple workbooks available in multiple languages. These are lessons that challenge teens to be as resourceful as hackers, including safe Internet use, web privacy, researching on the internet, avoiding viruses and Trojans, legalities and ethics, and more.

The HHS program is developed by ISECOM, a non-profit, open-source research group focused on security awareness and professional security development and accreditation.

Hacker Highschool
SECURITY AWARENESS FOR TEENS

ISECOM