Hacker Highschool

SECURITY AWARENESS FOR TEENS



LESSON 5
SYSTEM IDENTIFICATION

Hacker Highschool
SECURITY AWARENESS FOR TEENS

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

Glenn Norman, ISECOM

Marta Barceló, ISECOM

Chuck Truett, ISECOM

Kim Truett, ISECOM

Marco Ivaldi, ISECOM

Greg Playle, ISECOM

Bob Monroe, ISECOM

Simone Onofri, ISECOM

Ryan Oberto, Johannesburg South Africa

Dennis King

Mario Platt

Grigoris Chrysanthou, Cypress

ISECOM

Introduction

"I think my laptop has a virus," one of my students told me. "Can you take a look at it?"

I took the notebook computer from him, didn't open it, but tilted it every direction, looking closely. "Looks like a computer to me," I said, handing it back to him.

"But something's wrong with it," Aidan insisted. "I went to my friend's house and got on the Internet, and something got into my email and sent messages to all my friends."

"Okay, how do you get to your email? Did you install an application?" I asked.

"No, I do it on the web. I mean Internet."

"You mean in a web browser?" He nodded. "Then that means your email is online, not on your computer. So in this case I'd start with your email account. Have you changed the password?"

"Yeah. They shut down my account until I changed it." He looked down, like there was more to the story, but I didn't press him. My bet was that he'd already been yelled at. A lot.

"Have your friends gotten any more of the messages?" I asked instead.

"No." Staring firmly at his shoes.

"And did you choose a decent password? Not 12345?"

Now he smiled. "It's a really hard one. Nobody's ever gonna get it."

I had my doubts about that, but I nodded. "Okay, then, sounds like you've got it all sorted out."

"No," he insisted. "Why would somebody do that?"

Now I had the fish on the hook. "Why don't you find out. Do you have any of those emails that your friends got?"

"Yeah. A bunch of them. People sent them back to me." Ah: there it was. I'd bet his contact list numbered in the dozens. Or the hundreds. That had to have been fun.

"Then it sounds to me like you need to find out exactly where that link in the email goes."

Cameras flashed behind his eyes. "You mean we can do that?"

"Hah," I laughed. "It means YOU can do it. But I'll show you how."

Aidan stopped. "Is this what you mean by the sheep and the wolf you're always talking about?"

"Yes, exactly that. You can be one or the other. Choose now," I told him.

Suddenly he didn't look so much like a kid. "Wolf," he told me.

* * *

System identification can easily be the most important step of any computer attack or defense. Everything you do afterward depends on the data you gather at this stage. What's the operating system of the host that's attacking you, or that you're defending? Can you – or others – see what applications or services are running? How about the administrator's personal details: are they in plain sight anywhere? These are the questions

to ask at this stage. Depending on which side you're on, you might be delighted or horrified at what's easily available if you know where to look.

Knowing how an attack works is cool. Knowing how to protect against it or defeat it is even cooler. Here's where we start digging deep and learning how to identify a system and find its weaknesses – whether it's our own system or someone else's.

We'll be using tools that are publicly available and we'll even show you how to use them. It wouldn't make much sense to show you software but not teach you how to use it. As with any security program, they can be used for good or bad purposes. Our mission is to show you both uses so that you can fix your own security challenges, while protecting against similar attacks.

In this lesson, you'll be following two individuals as one teaches and the other person learns. The teacher doesn't always know what the answer will be so you as the reader will not be spoon-fed information either. Learn to break things and learn how to fix those things you broke. Repeat as necessary.

Pay close attention to attributes used in various programs. A slight change in an upper case to a lower case syntax letter may bring you entirely different data, more-so in different operating systems. These first few lessons are the foundation of networking and how the internet works. Each lesson builds on the previous knowledge so don't be in a hurry, but skipping around the paragraphs and pages is a good way to get familiar with this material before you go back and read in depth. Obviously you don't want to overlook a crucial piece of knowledge.

Identifying a Server

"Okay, Aidan, what did you find out?" I was trying not to grit my teeth with the fear that he'd gone and clicked that stupid link in the email his hacked account had sent out.

"I didn't left-click it," Aidan told me, smiling up like he'd read my mind. "I copied it and pasted it into a plain text file."

"The text you could see? Or the actual link?"

He frowned. "I'm not stupid. I right-clicked and chose 'Copy link location.' Then I pasted it here. Look, link.txt."

"Sorry. Just had to be sure. So okay. Where does it go?"

"This crazy domain. Chewmoogoo.com or something. There's a bunch of other stuff after that too," he said, opening his laptop and showing me the link.

"Oh yeah," I told him. "Now we've got 'em. Now let's see what information we can gather and the tools that can help us collect it. First let's talk about domain names and IP addresses."

Identifying the Owner of a Domain

The first step in identifying a remote system is to look at its host name, domain name or IP address. A **whois** lookup on a domain name turns up a bunch of information:

- The identity of the owner of the domain, usually a full name
- Contact information, which may include street addresses, phone numbers and email addresses
- The DNS servers where the domain is registered, which may also tell you the ISP that serves up the domain
- The IP address of the server, another potential clue to the ISP
- Domain name information, like the date it was created, when it was updated or when it will expire

Keep in mind that there are a lot of different domain name registrars, and not all whois databases contain the information for all domains. You may have to look at more than one whois database to find information about the domain that you are investigating.

Aidan soaked this up instantly. "Okay, what do I do?"

"Here's your assignment," I said.

Exercise

5.1 Get the domain name you're investigating. (If you're not Aidan, use isecom.org.) Try the following command on Linux, Windows and OSX.

```
whois isecom.org
```

Who owns the domain?
When was it created? When will it expire? (Does that expiration present an

opportunity?)
When was it last updated?
Who are the different contacts listed?
What are its primary and secondary name servers?

5.2 Now do the same lookup in a browser (for instance, http://www.whois.net -> "sample.com"). Here's the critical question: Does it match what you got from your whois command?
Check at least two whois websites. Try http://whois.domaintools.com; can you find more?).

Identifying the IP Address of a Domain

"So what have you got?" I asked Aidan.

"All this stuff. I pasted it in." He showed me his text file.

"That's good. Keep every single scrap of information. What's the domain IP?"

"This thing, isn't it?" Aidan pointed at a long number.

"Yes. You can get the domain's IP address with a whois command, or you can do a DNS lookup with a **ping** command:

```
ping isecom.org
```

"The first thing you'll see is the domain's IP address."

> If you can capture email from the target, examine the **email headers** (see Lesson 9, Email Security); that will give you the IP address of the originating mail host. You can also use resources like search engines (Lesson 20, Social Engineering) or tools like **Maltego** or **FOCA**. Search on terms like the target organization's name, the domain registration point of contact, telephone numbers and addresses. Each of those can lead you to more information.

"Once you've got an IP – or more than one – you need to find out where it is. IP numbers get assigned to service providers all over the globe in big groups. Find out which group an IP address was issued in (and who has the rights to that group, if you can). That can help you find out what server or service provider the website uses and the real gold for you - what country houses that server," I told Aidan. "Bet it's not this one. So here's what you do next."

Exercises

Now you're going to look at DNS records directly. Another way to find information about a domain and server(s) is to use information in DNS. There are three commands to get started.

5.3 Open a terminal window. Try this command:

```
dig isecom.org
```

Does this command work on your OS? Try it in Windows, Linux and OSX.

5.4 Now try this command:

```
host isecom.org
```

Does this command work on your OS? Try it in Windows, Linux and OSX again.

5.5 Finally try this command:

```
nslookup isecom.org
```

Does this command work on your OS? Once again, try it in Windows, Linux and OSX.

What's the DNS server for your target? Does the organization have a mail server? Does the mail server have the same IP address as the web server? What does this suggest? What else can you learn?

5.6 Once you have the IP address, you can access the records of the various members of the **Number Resource Organization** (http://www.arin.net/, http://www.ripe.net/, or http://www.apnic.net/), to gain insight about how IP addresses are distributed.

Game On: Slash and Burn

It was a grudge match as far as Jace was concerned. The battle of the century, as she thought of it. No matter how much sweat, blood, pain, physical and intellectual force required, the ambitious teen was prepared to win this fight. She had to win since there was no plan B. Her cocoa colored hair swayed over her eyes like a bullfighter taunting with a red cape. A one last calming deep breath and the network killer was prepared to being.

With her nimble fingers floating over the grinning keyboard, she assessed the situation and took stock of here available resources. Jace had a copy of Nmap already loaded into the computer beast. Ping and Traceroute had already been run so the combative hacker was ready to start slashing away.

Down went the first of a rapid succession of keyboard taps. A machine gun couldn't fire as quickly as Jace did when it came to working the computer commands. Ping, down! Traceroute, down! The IP commands didn't stand a chance against her massive barrage of key pecking. Time to live, down! The bloodshed was horrendous, as bits and bytes tumbled across the monitor in blurs. The CLI seemed to direct the incoming blitz of powerful switches, with attack attributes flanking the main network.

Jace maneuvered her main assault to gain a foothold inside the network. Her scouts performed an intensive reconnaissance of forward deployed firewalls, servers, and routers. This data was compared against Common Vulnerabilities and Exposures (CVE) and cross-referenced with Nmap's own Network Scanning information. Each weakness, every vulnerability, and exploit was examined for tactical advantage and damage assessments. A truce was not an option for Jace. She was winning.

It wasn't over yet, she told herself. In fact, all she had done was captured a small part of the enemies resources but the intelligence was invaluable, nonetheless. Jace suffered little causalities on her end. Fingers and knuckles were a slightly sore. She had a small bruise near her forehead where she banged it against the monitor in frustration. TTLs were killing her.

In the end, battle banners gave up details without the need for interrogation or repeated torture using the "bread-boarding" technique. Raspberry pie was kept in reserve. Jace had enough information on the enemy to perform phase two of the network attack. Next phase required loaded emails and the unintentional insider help.

This was always the scariest part of any battle, obtaining turncoats. Jace needed users on the inside who would be sympathetic to her cause. Now was the time for all good security habits to be broken. Social engineering was the weapon of mass disruption in her arsenal. She would have to craft legitimate emails loaded with Trojan soldiers to penetrate the networks inner walls.

As Jace began constructing each malicious email, she knew that she was on the right side of this confrontation. No matter what it took, no matter how long it took, Jace was determined to know which secret flavor of ice cream the local dairy was working on next.

Game continues...

Identifying Services

"So you saved all that stuff, right?" I grinned but tried not to, because I knew the answer even if my teacherly nature made me ask.

Aidan barely let a sideways glance slip: *bonehead* he was thinking, but he said, "Check it out" and handed me his notebook.

"Lots of info now, isn't there?" I scrolled down through the pages.

"Yeah. I need a better way to keep track of stuff," Aidan said, taking the computer back.

"You sure do. What's your target's IP?" This time I smiled openly.

"Well … there's about five. Maybe more than that. I'm trying to figure out why, because I can ping some and some I can't."

Good man I thought. Once you've got the IPs for a domain you can start digging into services, and that means running hosts. *Oh fun*.

Ping and Traceroute

"You're starting in the right place. You need to make sure there really are active machines. And you're right; ping is your friend. You did remember to ping the domain name, the IP addresses and the host names, didn't you?"

"Which ones are host names?" Aiden asked.

"They're the ones with letters and a dot before the domain name, like www.isecom.org," I told him.

"I don't see any of those."

"Check out your dig results. You didn't try the other ones. Did you try www.isecom.org and ftp.isecom.org and mail.isecom.org?"

"No...."

"Well, if you get a response, there's something alive at that address. And you're getting through the firewall. And they're letting ICMP through." I opened a CLI and entered a command.

```
C:\>ping isecom.org

Pinging isecom.org [216.92.116.13] with 32 bytes of data:
Reply from 216.92.116.13: bytes=32 time=186ms TTL=56
Reply from 216.92.116.13: bytes=32 time=186ms TTL=56
Reply from 216.92.116.13: bytes=32 time=186ms TTL=56
Reply from 216.92.116.13: bytes=32 time=186ms TTL=56

Ping statistics for 216.92.116.13:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
  Minimum = 186ms, Maximum = 186ms, Average = 186ms
```

"You can get an idea of how far the server is from you, both on the network and physically, by the round trip times. Divide in half, and you can get a feel for the distance to the server. I want you to try another tool, traceroute. It's spelled **tracert** in Windows and **traceroute** in Linux. It'll show you the steps packets take from your computer to your target. Like this," I said, and typed again.

```
C:\>tracert isecom.org
```

"Now, here's what I want you to do."

Exercises

5.7 Use traceroute/tracert to put together all the information you can find about the computers and routers between your computer and your target.

5.8 Computers with similar IP addresses are often part of the same network. Ping a valid website or IP address (for example, ping www.isecom.org or ping 216.92.116.13). If you get a successful response, ping the next IP address. Did you get a response? Try more nearby addresses.

5.9 Use a search engine to find out how to estimate the distance to the server.

5.10 Look for a tool that can help you map the server to a physical location.

5.11 Look for a Visual Trace Route tool online. There are quite a few sites that provide tools like this. This ought to give you a better visualization of where your traffic is going.

Nmap

"Got all that? Now let me introduce you to my little friend," I said, trying to do a Scarface voice. Aidan looked at me like I had two heads, so I cleared my throat, and, um, finished, "nmap."

"It can be really simple, or you can get really tricky. Run the nmap command with a host name or an IP address, and it'll scan that host. Or use a bunch of switches to do really tricky things. If you ask right, it'll try to tell you the OS of your target. We're going to use the 'scan TCP' option, which is -sT."

```
nmap -sT 216.92.116.13

Starting Nmap 5.51 ( http://nmap.org ) at 2012-05-28 10:58 GTB Daylight
Time
Nmap scan report for 216.92.116.13
Host is up (1.1s latency).
Not shown: 969 closed ports
PORT   STATE SERVICE
25/tcp  open smtp
80/tcp  open http
110/tcp  open pop3
119/tcp  open nntp
135/tcp  open msrpc
139/tcp  open netbios-ssn
143/tcp  open imap
445/tcp  open microsoft-ds
465/tcp  open smtps
554/tcp  open rtsp

Nmap done: 1 IP address (1 host up) scanned in 215.42 seconds
```

It's important to remember that nmap isn't the only tool for doing these scans, and that's a good thing. Different tools can give you different results, and in fact any of them can be deliberately misled.

You can tell nmap, for instance, to guess the operating system – but you should not trust its guess! Verify its theory using other tools.

Banner Grabbing

Aidan was gleeful. "Look what I've got now!" He had text documents and a spreadsheet on his laptop, drawings in a paper notebook and color printouts that had to have cost somebody a fortune in ink cartridges.

"Okay, now you know you've got some live machines, who runs it and roughly where it is. Next you want to know what kind of machine it is: what operating system is it running? What services is it running?" I asked him.

This made him less gleeful. "Um, how do I tell?"

"You don't have to. Get the machine to spill its guts: operating system, services and patch levels. When you're the attacker, that makes your job really easy; all you have to do is look up the exploits for that service, software and version. If you're the defender, you'll want to suppress that information. Or better yet, lie." This made him look thoughtful.

"So what you do next is called **banner grabbing**. Fancy word: it's an **enumeration technique** to get all kinds of information on your target active services and ports. I'm going to show you some more commands. You can use telnet, ftp or netcat to grab the banner. The banner is that old-school text message you'd get at the command line when you connected to tell you what server program is running. So check it out: when I connect to an anonymous FTP server, I get a banner." I typed into my terminal window:

```
ftp isecom.org


Connected to anon.server.
220 ProFTPD Server (Welcome . . . )
User (anon.server:(none)):
```

"That number 220 is a code that says the server's ready for a new user. And isn't this nice: ProFTPD Server is the FTP program running on that host. Now we hit the web to find out what OS ProFTPD runs on, what it can do … what's messed up, if anything." I rattled away on the keyboard. "Here: your next assignment is to use the ftp command."

Exercise

5.12 You can use FTP with either a host name or an IP address, like this:

```
ftp isecom.org
```

or

```
ftp 216.92.116.13
```

Try both to see what banner the FTP server returns. Your result may look like this:

```
Connected to isecom.org.
220 ftp316.pair.com NcFTPd Server (licensed copy) ready.
```

```
User (isecom.org:(none)):
```

5.13 You can use Telnet with either a host name or an IP address, too. With either one you can specify the port, which is 21 when we're connecting to FTP:

```
telnet isecom.org 21
```

or

```
telnet 216.92.116.13 21
```

Again, see what banner the server returns – if anything. You may get something like this:

```
220 ftp316.pair.com NcFTPd Server (licensed copy) ready.
```

5.14 Use netcat with either a host name or an IP address, too. Just like with Telnet, you can specify the port, which is 21 for FTP:

```
nc isecom.org 21
```

or

```
nc 216.92.116.13 21
```

Again, see what banner the server returns – if anything.

Misleading Banners

"Here's the trick," I told Aidan. "You can change the banner. That's one kind of **spoofing** – lying about who you are. So I can change my banner to read *NoneOfYourBusiness Server,* which is cute, but a Unix system with a banner that reads *WS_FTP Server* is going to throw people off, because that's a Windows FTP server."

"Wait a minute – how do you change the banner?" he asked.

"Glad you asked," I said.

Exercise

5.15 Get on the web and find out how to change the banners for SMTP, FTP, SSH, HTTP and HTTPS. Is it hard to do? In other words, should you just trust what banners say?

Automated Banner Grabbing

"Now check this out. We can go back to nmap and automate this; we have to use the -sTV switches to get banners." I typed the first line and got this report:

```
nmap -sTV -Pn -n --top-ports 10 --reason -oA hhs_5_06 hackerhighschool.org
```

```
Starting Nmap 6.00 ( http://nmap.org ) at 2012-06-23 05:10 CEST
Nmap scan report for hackerhighschool.org (216.92.116.13)
Host is up, received user-set (0.30s latency).
PORT    STATE  SERVICE     REASON     VERSION
21/tcp  open   ftp         syn-ack    NcFTPd
22/tcp  open   ssh         syn-ack    OpenSSH 5.9 (protocol 2.0)
23/tcp  closed telnet      conn-refused
25/tcp  filtered smtp      no-response
80/tcp  open   http        syn-ack    Apache httpd 2.2.22
110/tcp open   pop3        syn-ack    Dovecot pop3d
139/tcp closed  netbios-ssn  conn-refused
443/tcp open   ssl/http    syn-ack    Apache httpd 2.2.22
445/tcp closed  microsoft-ds conn-refused
3389/tcp closed  ms-wbt-server conn-refused
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.32 seconds
```

"Nmap found NcFTPd, OpenSSH 5.9 (protocol 2.0) and Apache httpd 2.2.22. Bingo: the OS is Unix. Sometimes the banners give you the operating system version, but we're going to need a little bit more info to get specific," I continued. "Here's what I want you to do."

Exercises

5.16 Use nmap on your target (hackerhighschool.org, if you're not Aidan).

5.17 Try it again with the option -**–version-intensity number** using numbers from 0 to 9 to get more accurate results. What differences can you see in these reports?

Identifying Services from Ports and Protocols

"Nmap did that last scan by looking for default services. But you can do it from the other direction too: look for open ports first, then see what service is actually behind them," I said.

"Wait a minute," Aidan demanded. "Aren't the ports always the same?"

"Yeah, in theory they are. But really, port numbers are sort of a gentlemen's agreement. I can put my services on different ports if I want."

"Okay, how do I do that?"

"Start by looking at your own local computer. Go to a command line and run the **netstat** command with the **-a** switch to scan all ports. Like this," I demonstrated.

```
netstat -a
```

The young hacker followed my example, then burst out, "Whoa! All of these are open?"

I looked at his screen. "Your computer is named Quasimodo?"

```
Active Connections
Proto  Local Address                  Foreign Address        State
TCP    Quasimodo:microsoft-ds         Quasimodo:0            LISTENING
TCP    Quasimodo:1025                 Quasimodo:0            LISTENING
TCP    Quasimodo:1030                 Quasimodo:0            LISTENING
TCP    Quasimodo:5000                 Quasimodo:0            LISTENING
TCP    Quasimodo:netbios-ssn          Quasimodo:0            LISTENING
TCP    Quasimodo:1110                 216.239.57.147:http    TIME_WAIT
UDP    Quasimodo:microsoft-ds         *:*
UDP    Quasimodo:isakmp               *:*
UDP    Quasimodo:1027                 *:*
UDP    Quasimodo:1034                 *:*
UDP    Quasimodo:1036                 *:*
UDP    Quasimodo:ntp                  *:*
UDP    Quasimodo:netbios-ns           *:*
UDP    Quasimodo:netbios-dgm          *:*
```

"Yeah, Quasimodo," Aidan grinned. "The Hunchback."

"Okay then, Victor. Here's what I want you to do."

Exercises

5.18 Run netstat on your local computer, using the -a switch.

```
netstat -a
```

Which ports are open?

5.19 Run netstat on your local computer, using the -o switch.

```
netstat -o
```

What services are listening behind the open ports?

5.20 Run netstat on your local computer, using the -aon switch combination.

```
netstat -aon
```

What does this combination get you?

5.21 Using a web search engine, match these ports with the services that run on them. Some of them you need for things like networking. But do you really want all the services you see running?

5.22 Run nmap, using the -sS (to do a SYN or so-called "stealth" scan) and -O (for guess operating system) switches and the IP address 127.0.0.1 as the target.The IP address 127.0.0.1 is called the **loopback** address. It always means localhost, your local computer.

```
nmap -sS -O 127.0.0.1
```

What open ports does nmap find? What services and programs are using these ports? Now try running nmap while you have a web browser or telnet client open. How does this change the results?

> The "stealth" scan uses just the first part of the TCP three-way handshake – the SYN packet – to probe a port without fully setting up a connection. While this gets you past the system's logs (which won't log your probe unless you really make a connection), it is NOT undetectable. Any intrusion detection system is going to see your big, greasy fingerprints all over the network, so don't fool yourself that you're really being stealthy.

5.23 Nmap has additional command line switches. What do -sV, -sU, -sP, -A, --top-ports and --reason do? What other possibilities are there? If you were an attacker and you wanted to remain stealthy rather than banging on the server, which switches should you *not* use, or use?

5.24 Go to www.foundstone.com, and find, download and install **fport** on your Windows box. It's similar to netstat, but it also details which programs are using the open ports and protocols. Run it. How does it compare to netstat?

System Fingerprinting

"You didn't go stumbling around and ringing bells, did you?" I asked.

Aidan replied slowly, really thinking about it, "No, I don't think so. But does it really matter? I mean their servers are way over in …."

I interrupted. "I don't know where they are, I don't care, you're going to operate ethically – and carefully – as long as you're working with me."

"Okay," sheepishly.

"It's a good policy to leave no tracks. Which is almost impossible. But you should always be trying. Because tracks are exactly what you're going to work on next. Or actually, fingerprints...."

"Hey! Those aren't the same!"

"Okay, got me. But regardless, we're going to put everything together to **fingerprint** your target, find the OS and all its services."

Scanning Remote Computers

"What did you finally get back on your stealth scans?" I asked. Aidan showed me a report he'd pasted into a text document.

```
nmap -sS -O 216.92.116.13

Starting Nmap 5.51 ( http://nmap.org ) at 2012-05-28 16:54 GTB Daylight
Time
Nmap scan report for isecom.org (216.92.116.13)
Host is up (0.19s latency).
Not shown: 965 closed ports

PORT   STATE  SERVICE
21/tcp  open   ftp
22/tcp  open   ssh
25/tcp  filtered smtp
26/tcp  open   rsftp
80/tcp  open   http
110/tcp  open   pop3
111/tcp  filtered rpcbind
113/tcp  filtered auth
135/tcp  filtered msrpc
139/tcp  filtered netbios-ssn
143/tcp  open   imap
161/tcp  filtered snmp
179/tcp  filtered bgp
306/tcp  open   unknown
```

```
443/tcp  open   https
445/tcp  filtered microsoft-ds
465/tcp  open   smtps
514/tcp  filtered shell
543/tcp  open   klogin
544/tcp  open   kshell
587/tcp  open   submission
646/tcp  filtered ldp
800/tcp  filtered mdbs_daemon
993/tcp  open   imaps
995/tcp  open   pop3s
1720/tcp filtered H.323/Q.931
2105/tcp open   eklogin
6667/tcp filtered irc
7000/tcp filtered afs3-fileserver
7001/tcp filtered afs3-callback
7007/tcp filtered afs3-bos
7777/tcp filtered cbt
9000/tcp filtered cslistener
12345/tcp filtered netbus
31337/tcp filtered Elite
Device type: general purpose|storage-misc
Running (JUST GUESSING): FreeBSD 7.X|6.X (88%)
Aggressive OS guesses: FreeBSD 7.0-BETA4 - 7.0 (88%), FreeBSD 7.0-RC1
(88%), FreeBSD 7.0-RELEASE - 8.0-STABLE (88%), FreeBSD 7.0-STABLE (88%),
FreeBSD
 7.1-RELEASE (88%), FreeBSD 6.3-RELEASE (86%), FreeNAS 0.7 (FreeBSD 7.2-
RELEASE) (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 8 hops
OS detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.09 seconds
```

"See all those ports marked **filtered**? That means they're protected by a firewall. They're well-known and vulnerable, so they should always be blocked. But look: ports 21, 22 and 80 – that's FTP, Secure Shell and HTTP – are all open." I looked over at Aidan.

"Sheep?" he asked hopefully.

"Well, rightful prey, at least. Okay. The last thing that nmap does is try to figure out the operating system on your target. Lots of the time, like now, it only makes an 'aggressive

guess,' but that's usually pretty good. Since the scan shows FTP and SSH open, the banners you grabbed would be the next piece of evidence.

"Hit the web, it tells us NcFTPd is a Unix program and that FreeBSD is a Unix-type operating system. SSH you'd usually find on Unix-like OSs. So it's likely the server's running some version of FreeBSD. You know those banners can be spoofed, but it's a reasonable guess.

"Now, depending on where your target is, your next step might be to find the ISP. The ISP itself might be famous for hosting spammers or malicious sites – do a search – but you might be able to whine to them and get your evil attacker shut down. In your case I think it's not going to be an ISP you can really deal with....

"Because it's in..." Aidan burst in, but I held up my finger.

"Stop. Your information is your information. I don't need it, as long as you are being ethical and safe. Which you are."

Aidan nodded.

"So watcha gonna do?" I asked.

"Well, they've got a web server running, right?" Aidan began, and all I could do was smile.

Feed Your Head: Going Deep with Nmap

Say you've identified the hostname, the owner, the network and verified the host is up. Now in order to identify a system you need to find some open ports. Don't forget that the host can be up but have all ports closed (or even filtered).

You can use the famous Network Mapper (aka **nmap**) tool from Fyodor to do this task. Nmap is a port scanner and is able to remotely probe computers for open ports and related network services. When you execute nmap, depending on the command line switches that you use, you'll get a list of open ports and the services or protocols that use those ports. Nmap may also be able to determine what operating system your computer is using.

Nmap has many options and scan types. We will use a few nmap options but you can always use

```
nmap --help
```

or

```
man nmap
```

to see the details.

Before we begin, have you read Lesson 3? No? Now's the time to do it! Back already? No? Then go now!

Ok, explain the differences between TCP and UDP and describe the three-way handshake. Knowing how this works is important for understanding how nmap works.

Nmap syntax is:

```
nmap scan-techniques host-discovery options target
```

- **scan-techniques** specify what kind of packets will be used and how responses from target should be interpreted. The main available techniques are:
  - **-sS** SYN scan (yes, only the first part of thee-handshake)
  - **-sT** TCP Connect scan (full three-way handshake)
  - **-sA** ACK scan (send only ACK packets)
  - **-sU** UDP Scan
  - **-O** OS Detection
  - **-A** All functionalities such as OS detection, plugins, traceroute
- **host-discovery** specifies the techniques used to define if a host is alive or not. If the host is alive it will be scanned, otherwise not.
  - **-PE** check if host responds to a ping
  - **-PS** check if host responds to a SYN
  - **-PA** check if host responds to an ACK
  - **-PU** check if host responds to a UDP datagram
  - **-PN** don't check, treat all hosts as active (we'll use this because we know that

our target is alive, since we checked earlier)

- **options** specify further details for the selected scan type, such as
  - **-p1-65535** port numbers to scan (in this example from 1 to 65535).
  - **--top-ports <number>** nmap knows which are the most frequently used ports, and can scan only for the top <number> specified
  - **-T0, -T1, -T2, -T3, -T4** for the scan speed, where 0 is slow and 4 is fast (slower means more stealthy and with less network congestion)
  - **-oA <filename>** for the output in all three main nmap formats (we will always use it to keep track of our activities)
  - **--reason** nmap writes about its interpreted results (recommended)
  - **--packet-trace** similar to –reason but you will see the traffic traces (you use these to learn about a scan technique and to troubleshoot scans)
  - **-n** do not resolve DNS (we will not use DNS because we have already analyzed it manually)

TCP Scan

Our first scan starts with the following command:

```
nmap -sT -Pn -n --top-ports 10 -oA hhs_5_tcp hackerhighschool.org
```

Which gives us this output:

```
Starting Nmap 6.00 ( http://nmap.org ) at 2012-06-23 04:10 CEST
Nmap scan report for hackerhighschool.org (216.92.116.13)
Host is up (0.23s latency).
PORT    STATE  SERVICE
21/tcp  open   ftp
22/tcp  open   ssh
23/tcp  closed  telnet
25/tcp  filtered smtp
80/tcp  open   http
110/tcp open   pop3
139/tcp closed  netbios-ssn
443/tcp open   https
445/tcp closed  microsoft-ds
3389/tcp closed  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 2.04 seconds
```

We found some ports open, some closed and one filtered. What does this mean? It depends on the scan type (in this case -sT). And we can use the --reason option to see

why nmap has inferred a particular State.

```
nmap -sT -Pn -n --top-ports 10 --reason -oA hhs_5_tcp_02
hackerhighschool.org

Starting Nmap 6.00 ( http://nmap.org ) at 2012-06-23 04:17 CEST
Nmap scan report for hackerhighschool.org (216.92.116.13)
Host is up, received user-set (0.22s latency).
PORT   STATE  SERVICE    REASON
21/tcp  open   ftp      syn-ack
22/tcp  open   ssh      syn-ack
23/tcp  closed  telnet    conn-refused
25/tcp  filtered smtp     no-response
80/tcp  open   http     syn-ack
110/tcp open   pop3     syn-ack
139/tcp closed  netbios-ssn  conn-refused
443/tcp open   https     syn-ack
445/tcp closed  microsoft-ds conn-refused
3389/tcp closed  ms-wbt-server conn-refused

Nmap done: 1 IP address (1 host up) scanned in 2.26 seconds
```

Now we know how nmap "maps" replies to states for **TCP Scan**:

- **open**: target replies with a SYN ACK packet
- **closed**: TCP connection refused
- **filtered**: no reply from target

When you find open and filtered ports use other scan techniques to find out exactly why.

SYN Scan

Another famous scanning technique is the SYN scan. When it's doing this type of scan, nmap sends only a SYN packet without completing the three-way handshake. This is also called a "half-open" or "stealth" scan because there TCP connections are not completed. (Be very clear that while a target may not log a connection, you are still making digital "noise" that can be detected.) Use the -sS scan type as follows:

```
nmap -sS -Pn -n --top-ports 10 --reason -oA hhs_5_syn
hackerhighschool.org

Starting Nmap 6.00 ( http://nmap.org ) at 2012-06-24 12:58 CEST
Nmap scan report for hackerhighschool.org (216.92.116.13)
```

```
Host is up, received user-set (0.15s latency).
PORT    STATE  SERVICE     REASON
21/tcp  open   ftp         syn-ack
22/tcp  open   ssh         syn-ack
23/tcp  closed telnet      reset
25/tcp  filtered smtp      no-response
80/tcp  open   http        syn-ack
110/tcp open   pop3        syn-ack
139/tcp filtered netbios-ssn  no-response
443/tcp open   https       syn-ack
445/tcp filtered microsoft-ds no-response
3389/tcp closed  ms-wbt-server reset

Nmap done: 1 IP address (1 host up) scanned in 1.81 seconds
```

The results are similar to the TCP Scan but notice the differences between "full" TCP Scan and "half-open" SYN scan, comparing the results (with –reason and –packet-trace) using the same target with -sT, -sS and -sA (ACK scan).

UDP scan

Another scan technique is the UDP scan (-sU): knowing the reason is fundamental to getting good results.

```
nmap -sU -Pn -n --top-ports 10 --reason -oA hhs_5_udp
hackerhighschool.org

Starting Nmap 6.00 ( http://nmap.org ) at 2012-06-23 04:28 CEST
Nmap scan report for hackerhighschool.org (216.92.116.13)
Host is up, received user-set (0.23s latency).
PORT    STATE      SERVICE    REASON
53/udp  closed     domain     port-unreach
67/udp  open|filtered dhcps     no-response
123/udp closed     ntp      port-unreach
135/udp closed     msrpc     port-unreach
137/udp closed     netbios-ns  port-unreach
138/udp closed     netbios-dgm port-unreach
161/udp closed     snmp      port-unreach
445/udp closed     microsoft-ds port-unreach
631/udp closed     ipp      port-unreach
1434/udp closed     ms-sql-m   port-unreach
```

```
Nmap done: 1 IP address (1 host up) scanned in 2.05 seconds
```

It can be a little bit confusing. What happened? We see some of the reasons: port-unreach (unreachable, i.e. closed) and no-response (open|filtered). Why? We need more details. We can use the packet trace option and limit the scan to two ports, for example ports 53 and 67 UDP:

```
nmap -sU -Pn -n -p53,67 --reason --packet-trace -oA hhs_5_udp_02
hackerhighschool.org


Starting Nmap 6.00 ( http://nmap.org ) at 2012-06-23 04:32 CEST
SENT (0.0508s) UDP 192.168.100.53:54940 > 216.92.116.13:67 ttl=46
id=54177 iplen=28
SENT (0.0509s) UDP 192.168.100.53:54940 > 216.92.116.13:53 ttl=37
id=17751 iplen=40
RCVD (0.3583s) ICMP 216.92.116.13 > 192.168.100.53 Port unreachable
(type=3/code=3) ttl=54 id=1724 iplen=56
SENT (2.5989s) UDP 192.168.100.53:54941 > 216.92.116.13:67 ttl=49
id=33695 iplen=28
Nmap scan report for hackerhighschool.org (216.92.116.13)
Host is up, received user-set (0.31s latency).
PORT STATE SERVICE REASON
53/udp closed domain port-unreach
67/udp open|filtered dhcps no-response


Nmap done: 1 IP address (1 host up) scanned in 4.15 seconds
```

We found out that 192.168.100.53 sent UDP packets to port 53 and 67 of hackerhighschool.org. What happened here? Port 67 is unresponsive and for 53 we received a Port Unreachable (T03C03).

Port Unreachable means the port is closed, and as for no-response – even if is a normal response for UDP – we don't know if the service is active or not because the UDP protocol can only reply if it receives the correct packets. Can we investigate it more? Yes, using the -sV Service Scan in which nmap tries to send well-known packets for UDP services.

Service Scan (UDP)

```
nmap -sUV -Pn -n -p53,67 --reason --packet-trace -oA hhs_5_udp_03
hackerhighschool.org


Starting Nmap 6.00 ( http://nmap.org ) at 2012-06-23 04:44 CEST
SENT (0.1730s) UDP 192.168.100.53:62664 > 216.92.116.13:53 ttl=48
```

```
id=23048 iplen=40

SENT (0.1731s) UDP 192.168.100.53:62664 > 216.92.116.13:67 ttl=48
id=53183 iplen=28

RCVD (0.4227s) ICMP 216.92.116.13 > 192.168.100.53 Port unreachable
(type=3/code=3) ttl=54 id=20172 iplen=56

SENT (2.4252s) UDP 192.168.100.53:62665 > 216.92.116.13:67 ttl=50
id=39909 iplen=28

NSOCK (3.8460s) UDP connection requested to 216.92.116.13:67 (IOD #1)
EID 8

NSOCK (3.8460s) Callback: CONNECT SUCCESS for EID 8 [216.92.116.13:67]

Service scan sending probe RPCCheck to 216.92.116.13:67 (udp)

...and 80 more packets...

Nmap scan report for hackerhighschool.org (216.92.116.13)

Host is up, received user-set (0.25s latency).

PORT  STATE     SERVICE REASON    VERSION

53/udp closed    domain port-unreach

67/udp open|filtered dhcps  no-response
```

We're not lucky this time, since we got the same results. A good hacker can also try specific UDP packets manually, or with the proper client on the standard port 67. We have already used the service scan, the next step for service identification. Learn the well known services on your local machine and do some exercises, then continue with banner grabbing.

Exercises

5.25 Go to http://nmap.org, download and install the latest version of nmap for your operating system.

5.26 Repeat all the scans in this section using more ports. Have in mind that you need the sudo command on Linux systems, or local administrator rights on Windows.

5.27 Create a table reference for all scan techniques mapping state, reason and the real response from the target (packet-trace).

OS Detection

Knowing services is important to fingerprinting the target machine. Nmap can help again using more options such as -A for all scans and -O for OS detection, using the default ports:

```
sudo nmap -A -Pn -n --reason -oA hhs_5_all hackerhighschool.org


Starting Nmap 6.00 ( http://nmap.org ) at 2012-06-23 05:38 CEST
```

```
Nmap scan report for hackerhighschool.org (216.92.116.13)
Host is up, received user-set (0.21s latency).
Not shown: 971 closed ports
Reason: 971 resets
PORT   STATE  SERVICE     REASON   VERSION
21/tcp  open   ftp       syn-ack   NcFTPd
22/tcp  open   ssh       syn-ack   OpenSSH 5.9 (protocol 2.0)
|  ssh-hostkey:  1024  cd:27:c2:bf:ad:35:e5:67:e0:1b:cf:ef:ac:2b:18:9a
(DSA)
|_1024 17:83:c5:8a:7a:ac:6c:90:48:04:0b:e5:9c:e5:4d:ab (RSA)
25/tcp  filtered smtp      no-response
26/tcp  open   tcpwrapped   syn-ack
80/tcp  open   http      syn-ack   Apache httpd 2.2.22
|_http-title: Hacker Highschool - Security Awareness for Teens
110/tcp  open   pop3      syn-ack   Dovecot pop3d
|_pop3-capabilities: USER CAPA UIDL TOP OK(K) RESP-CODES PIPELINING
STLS SASL(PLAIN LOGIN)
111/tcp  filtered rpcbind     no-response
113/tcp  open   tcpwrapped   syn-ack
143/tcp  open   imap      syn-ack   Dovecot imapd
|_imap-capabilities:  LOGIN-REFERRALS  QUOTA  AUTH=PLAIN  LIST-STATUS
CHILDREN   CONTEXT=SEARCH   THREAD=REFERENCES   UIDPLUS   SORT   IDLE
MULTIAPPEND CONDSTORE ESEARCH Capability UNSELECT AUTH=LOGINA0001
IMAP4rev1 ID WITHIN QRESYNC LIST-EXTENDED SORT=DISPLAY THREAD=REFS
STARTTLS OK completed SEARCHRES ENABLE I18NLEVEL=1 LITERAL+ ESORT
SASL-IR NAMESPACE
161/tcp  filtered snmp      no-response
179/tcp  filtered bgp       no-response
306/tcp  open   tcpwrapped   syn-ack
443/tcp  open   ssl/http    syn-ack   Apache httpd 2.2.22
| ssl-cert: Subject: commonName=www.isecom.org/organizationName=ISECOM
-      The      Institute      for      Security      and      Open
Methodologies/stateOrProvinceName=New York/countryName=US
| Not valid before: 2010-12-11 00:00:00
|_Not valid after: 2013-12-10 23:59:59
|_http-title: Site doesn't have a title (text/html).
|_sslv2: server supports SSLv2 protocol, but no SSLv2 cyphers
465/tcp  open   ssl/smtp    syn-ack   Postfix smtpd
|_smtp-commands: kunatri.pair.com, PIPELINING, SIZE 41943040, ETRN,
AUTH PLAIN LOGIN, AUTH=PLAIN LOGIN, ENHANCEDSTATUSCODES, 8BITMIME,
DSN,
```

```
|   ssl-cert:   Subject:   commonName=*.pair.com/organizationName=pair
Networks, Inc./stateOrProvinceName=Pennsylvania/countryName=US

| Not valid before: 2012-01-10 00:00:00

|_Not valid after: 2015-01-09 23:59:59

543/tcp  open   tcpwrapped   syn-ack

544/tcp  open   tcpwrapped   syn-ack

587/tcp  open   smtp      syn-ack   Postfix smtpd

|_smtp-commands: kunatri.pair.com, PIPELINING, SIZE 41943040, ETRN,
STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN,

|   ssl-cert:   Subject:   commonName=*.pair.com/organizationName=pair
Networks, Inc./stateOrProvinceName=Pennsylvania/countryName=US

| Not valid before: 2012-01-10 00:00:00

|_Not valid after: 2015-01-09 23:59:59

646/tcp  filtered ldp       no-response

800/tcp  filtered mdbs_daemon   no-response

993/tcp  open   ssl/imap    syn-ack   Dovecot imapd

|   ssl-cert:   Subject:   commonName=*.pair.com/organizationName=pair
Networks, Inc./stateOrProvinceName=Pennsylvania/countryName=US

| Not valid before: 2012-01-10 00:00:00

|_Not valid after: 2015-01-09 23:59:59

|_sslv2: server supports SSLv2 protocol, but no SSLv2 cyphers

|_imap-capabilities:   LOGIN-REFERRALS   completed   OK   SORT=DISPLAY
Capability   UNSELECT   AUTH=PLAIN   AUTH=LOGINA0001   IMAP4rev1   QUOTA
CONDSTORE LIST-STATUS ID SEARCHRES WITHIN CHILDREN LIST-EXTENDED ESORT
ESEARCH    QRESYNC    CONTEXT=SEARCH    THREAD=REFS    THREAD=REFERENCES
I18NLEVEL=1  UIDPLUS  NAMESPACE  ENABLE  SORT  LITERAL+  IDLE  SASL-IR
MULTIAPPEND

995/tcp  open   ssl/pop3    syn-ack   Dovecot pop3d

|_sslv2: server supports SSLv2 protocol, but no SSLv2 cyphers

|_pop3-capabilities: OK(K) CAPA RESP-CODES UIDL PIPELINING USER TOP
SASL(PLAIN LOGIN)

|   ssl-cert:   Subject:   commonName=*.pair.com/organizationName=pair
Networks, Inc./stateOrProvinceName=Pennsylvania/countryName=US

| Not valid before: 2012-01-10 00:00:00

|_Not valid after: 2015-01-09 23:59:59

2105/tcp open   tcpwrapped   syn-ack

6667/tcp filtered irc       no-response

7000/tcp filtered afs3-fileserver no-response

7001/tcp filtered afs3-callback  no-response

7007/tcp filtered afs3-bos    no-response

7777/tcp filtered cbt       no-response
```

```
9000/tcp filtered cslistener   no-response

31337/tcp filtered Elite      no-response

Device type: general purpose|firewall|specialized|router

Running (JUST GUESSING): FreeBSD 6.X|7.X|8.X (98%), m0n0wall FreeBSD
6.X (91%), OpenBSD 4.X (91%), VMware ESX Server 4.X (90%), AVtech
embedded (89%), Juniper JUNOS 9.X (89%)

OS    CPE:    cpe:/o:freebsd:freebsd:6.3    cpe:/o:freebsd:freebsd:7.0
cpe:/o:freebsd:freebsd:8.1                    cpe:/o:m0n0wall:freebsd
cpe:/o:openbsd:openbsd:4.0                    cpe:/o:vmware:esxi:4.1
cpe:/o:m0n0wall:freebsd:6 cpe:/o:juniper:junos:9

Aggressive OS guesses: FreeBSD 6.3-RELEASE (98%), FreeBSD 7.0-RELEASE
(95%), FreeBSD 8.1-RELEASE (94%), FreeBSD 7.1-PRERELEASE 7.2-STABLE
(94%), FreeBSD 7.0-RELEASE - 8.0-STABLE (92%), FreeBSD 7.1-RELEASE
(92%), FreeBSD 7.2-RELEASE - 8.0-RELEASE (91%), FreeBSD 7.0-RC1 (91%),
FreeBSD 7.0-STABLE (91%), m0n0wall 1.3b11 - 1.3b15 FreeBSD-based
firewall (91%)

No exact OS matches for host (test conditions non-ideal).

Network Distance: 12 hops

Service Info: Host: kunatri.pair.com; OS: Unix


TRACEROUTE (using port 1723/tcp)

HOP RTT    ADDRESS

[...]

8  94.98 ms 89.221.34.153

9  93.70 ms 89.221.34.110

10 211.60 ms 64.210.21.150

11 ...

12 209.28 ms 216.92.116.13


OS and Service detection performed. Please report any incorrect
results at http://nmap.org/submit/ .

Nmap done: 1 IP address (1 host up) scanned in 57.94 seconds
```

Using -A is possible to see more data. Specialized plugins fetch more information from a server, perform OS Guessing and use a variant of traceroute that uses different methods than regular traceroute or tracert. For OS guessing more ports are better.

Exercises

5.28 Scan your own machine with nmap. Is the OS guessing valid?

5.29 Use the traceroute option on nmap using different ports:

```
nmap -n -Pn --traceroute --version-trace -p80 hackerhighschool.org
```

5.30 Are there some differences on nmap traceroute using different ports and tracert or traceroute from your OS?

5.31 Research TCP/IP stack fingerprinting. How do you do it? Is it spoof-proof?

Using Scripts

Nmap also has a lot of useful scripts for scanning. You can use the `–script script-name` option to load scripts. One interesting script is `ipidseq`, which performs Incremental IP fingerprinting. This script can be used to find hosts for Idle Scan (-sI). This scan uses a problematic IP implementation on zombie hosts to scan other targets.

```
nmap --script ipidseq -oA hhs_5_ipidseq hackerhighschool.org


Starting Nmap 6.00 ( http://nmap.org ) at 2012-06-23 05:47 CEST
Nmap scan report for hackerhighschool.org (216.92.116.13)
Host is up (0.23s latency).
rDNS record for 216.92.116.13: isecom.org
Not shown: 971 closed ports
```

Exercises

5.32 Research Idle Scan techniques. What is it and how do you do it?

Conclusion

Knowing where to look and what to look for is only part of the security battle. Networks are constantly being surveyed, analyzed, poked and prodded. If the network you are protecting isn't being watched then you aren't using the right tools to detect that behavior. If the network you're cracking isn't being watched, you may (may) get away with scanning it. As a cyber security expert, you should know every inch of the systems you are protecting – or testing. You need to know where the weaknesses are and where the strengths are as well, regardless of which side you're on.

Simply gathering up intelligence on a server, such as the operating system and open ports, isn't enough these days. An Advanced Persistent Threat will try to learn as much about your network as it can. This information includes -

- Firewall brand, model, firmware version, and software patches that exists
- Remote connections authentication, access privileges, and processes
- Other servers that connect to the network, this includes Email, HTML, back-up, redundant, off-site, hired or out-sourced services, and even contractors that may have used your network or are using it now
- Printers, fax machines, photocopiers, wireless routers, and network connections in your company waiting room
- Portable devices such as tablets, smartphones, digital picture frames, and anything that might connect to the network.

Even though we have covered many topics in this lesson, system identification covers an even broader area. There is quite a bit of information that flows through networks that identify parts of each device. Each device on the network can be exploited and thus used as an entry point for an attacker. Approaching this daunting challenge requires more than just software. Research your own equipment and learn as much as you can. That knowledge will pay off.

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