



v4

Penetration Testing Student

Preliminary Skills

Section 01 | Module 01

© Caendra Inc. 2019
All Rights Reserved

Table of Contents

Module 01 | Preliminary Skills

- 1.1 Welcome
- 1.2 The Information Security Field
- 1.3 Cryptography and VPNs
- 1.4 Wireshark Introduction
- 1.5 Binary Arithmetic Basics



Learning Objectives

By the end of this module, you should have a better understanding of:

- The Infosec Culture
- Basics of Cryptography
- Wireshark Usage
- Numeric Systems



Welcome



1.1 Welcome



Welcome to this eLearnSecurity course,
Penetration Testing Student version 4.

With this course, you are starting your
journey in IT security and penetration
testing!



1.1 Welcome

This module will give you the basic skills to get started, as well as:

- Some information about the IT security field: culture, career opportunities, and jargon.
- Basic understanding of the difference between a clear-text protocol and an encrypted protocol.
- **Your first laboratory:** intercepting network traffic!

1.1.1 Course Structure

Let's start by introducing the course material. The *Penetration Testing Student version 4* course comes in different plans. According to your plan you get different training material:

- The **Barebone** plan includes training slides only.
- The **Full** and the **Elite** plans come with training slides, video lessons, and practical hands-on labs in the Hera Lab environment plus the opportunity to become certified!

Now, let's see how to best use the training material!

1.1.2 Slides

The course is divided into three main sections:



Every section is made up of several **modules**.

The content of every module is made up of slides, like the ones you are reading right now.

1.1.2 Slides

In the **slides**, you will also find directions on how to access the videos, as well as how to access the labs where you will put into **practice** the theoretical skills acquired.

Every module contains an index at the beginning, in addition to external references at the end along with a full list of videos and labs available for the module you are reading.



1.1.2 Slides

You can immediately access different places in the slides by using the right navigation menu. For instance, clicking on the **Map** icon (🌐) will take you to the Table of Contents where you can select to review a topic of your choosing, while the **References** folder (📁), **Video** icon (🎬), or **Lab** icon (🧪) buttons will take you to the end of the slides to show you the full list of references, videos or labs respectively; this feature comes in very handy throughout your learning journey when you need to rapidly find a link, a video, or practice a topic a little more.



1.1.3 Videos

Videos are a great way to see information presented in the course put into action; this allows you to deepen your knowledge on specific topics.

Throughout the course, you will encounter some special slides containing a link to a video.


You can start the video by clicking on the video image.

1.4.1 Video

HTTPS and HTTPS Traffic Sniffing

In this video, you will see how to use *Wireshark* to intercept communication between your computer and a web server placed inside a dedicated network within Hera Lab.

This will help you understand the difference between a clear-text protocol and a cryptographic one from an attacker point of view.



*Videos are only available in Full or Elite Editions of the course. To upgrade, click [HERE](#). To access, click the image above or go to the course in your members area and click the resources drop-down in the appropriate module line.

PTSv4 - Caendra Inc. © 2019 | p.65

1.1.3 Videos

Additionally, you can access videos from your member's area by going to the appropriate module line and clicking the resources drop-down menu.

Course content Expand All | Collapse All

Preliminary Skills - Prerequisites

Progress	Module Name	HTML5	PDF	Resources	Labs	Completed
In Progress	1 - Introduction	Study		Resources ▾	Labs ▾	<input type="checkbox"/>
In Progress	2 - Networking	Study		Resources ▾	Labs ▾	<input type="checkbox"/>
In Progress	3 - Web Applications	Study		Resources ▾	Labs ▾	<input type="checkbox"/>
In Progress	4 - Penetration Testing	Study		Resources ▾	Labs ▾	<input type="checkbox"/>

Preliminary Skills - Programming

Progress	Module Name	HTML5	PDF	Resources	Labs	Completed
In Progress	1 - C++	Study		Resources ▾	No labs	<input type="checkbox"/>
In Progress	2 - Python	Study		Resources ▾	No labs	<input type="checkbox"/>

Resources dropdown menu (for Module 3):

- HTTP(s) Protocol Basics
- HTTP(s) Cookies and Sessions
- Burpsuite

1.1.3 Videos

Practice makes perfect! Try to apply what you see in the videos as much as you can.

As a reminder, videos are available in the **Full** and **Elite** plans only.



1.1.4 Virtual Labs

Hera Virtual labs, the most sophisticated labs on IT Security, differentiate eLearnSecurity courses from all the others.

Each virtual lab scenario included in this course consists of **isolated** computer network environments, which means that every lab is **dedicated** to the student and other students **will not be able to interfere with your environment**.

It would be a shame if you were successfully attacking a machine while another student made it crash! Right?



1.1.4 Virtual Labs

Throughout the course, you will find special slides telling you that you are ready to put into practice what you have just learned.

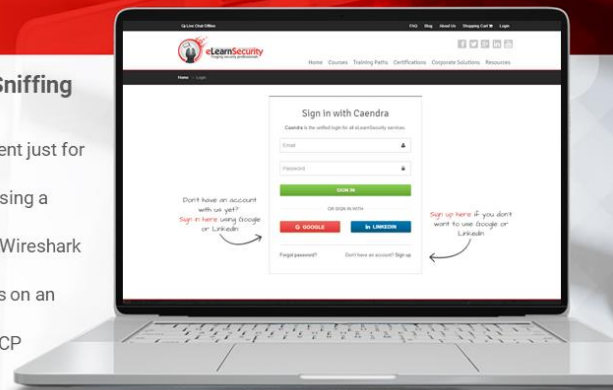
Those slides contain instructions regarding the lab learning objectives.

1.4.2 Hera Lab

HTTP and HTTPS Traffic Sniffing

In this lab you will:

- Start a dedicated lab environment just for you
- Login to a web application by using a username and a password
- Sniff the login credentials with Wireshark on a clear-text communication
- Try to sniff the login credentials on an encrypted communication
- Learn how to use the "Follow TCP stream" command



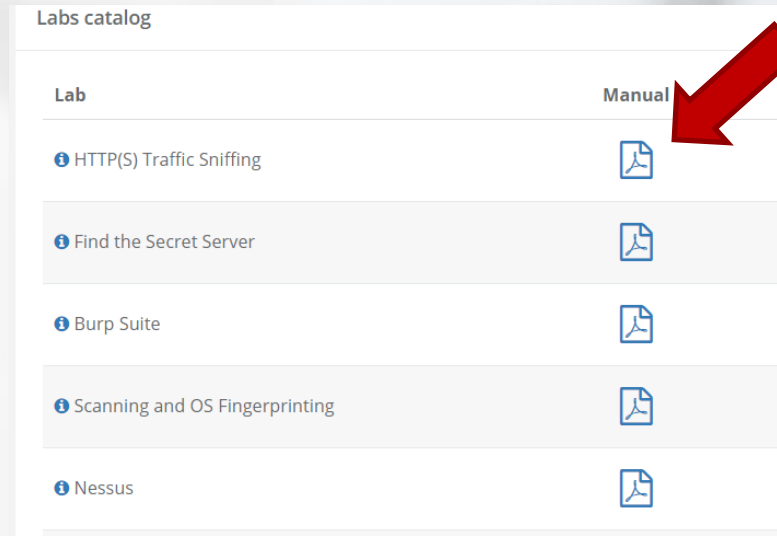
**Labs are only available in Full or Elite Editions of the course. To upgrade, click [HERE](#). To access, go to the course in your members area and click the labs drop-down in the appropriate module line or to the virtual labs tabs on the left navigation.*

PTsv4 - Caendra Inc. © 2019 | p.65

1.1.4 Virtual Labs

If you get stuck while doing the labs, don't worry!

You can find the solutions to each lab in the related PDF Lab Manual.



Lab	Manual
HTTP(S) Traffic Sniffing	PDF
Find the Secret Server	PDF
Burp Suite	PDF
Scanning and OS Fingerprinting	PDF
Nessus	PDF

1.1.4 Virtual Labs

To best use a lab, a great idea is to first focus on achieving the lab goals and then use it for the extra-mile: to **test other tools and techniques** to reach the same goals or even attempt **other attacks**.

Since every lab in Hera is made up of networks of real computers and servers, there are many (unexpected) things you can do!

Please note that the labs are available in the **Full** and **Elite** plans only!



The Information Security Field



1.2 The Information Security Field

How does this support my pentesting career?

- Knowing the information security field
- Career opportunities
- Talking with colleagues

1.2.1 Infosec Culture

Information Security has deep roots in the **underground hacking scene**, which still looks at computer systems with curiosity, trying to figure out new ways to use and break them!



1.2.1 Infosec Culture

The term *hacker* was born in the sixties in the MIT community.

It refers to people who prefer to understand how a system works rather than just using it. These people are **curious, highly intelligent and strongly motivated to pursue knowledge!**



1.2.1 Infosec Culture

Approaching systems with curiosity lets hackers and infosec professionals find new ways to use computer systems, bypassing the restrictions imposed by software vendors or programmers and deeply understanding any security pitfall of any kind of implementation.

Being able to perform an attack also means being able to deeply understand the technology and the functioning of the target system.



1.2.1 Infosec Culture

Being a hacker means being pushed by curiosity and having a hunger for knowledge. Hackers explore and improve their skills daily.

This aspect is still valid in the modern information security field; **there is always something new** to learn, something interesting to try or something exciting to study!



1.2.1 Infosec Culture

The history of hacking could easily be an entire book or a course by itself. If you do a quick Internet search on hacking, you will find that it is not necessarily related to computers only.

Hacking is more of an approach, or a lifestyle, applied to telephone lines, people and software development.

https://en.wikipedia.org/wiki/John_Draper

<https://www.mitnicksecurity.com/about/kevin-mitnick-worlds-most-famous-hacker-biography>

<https://stallman.org/>

1.2.1 Infosec Culture

The Conscience of a Hacker, also known as *The Hacker's Manifesto* written by *The Mentor*, is a document that gives an idea about the ideals of the underground hacking community.

Being an information security professional means pursuing knowledge, being honest with yourself and never stop challenging yourself and your colleagues.

1.2.2 Career Opportunities

Nowadays, companies of all sizes, as well as government bodies are using advanced technologies to store and process a great deal of confidential data on computers and mobile devices.

Data is not only stored but also transmitted across private and public networks to other computers. Therefore, it is a **must** to protect sensitive information. Companies pay a premium to safeguard their data and ensure that their systems are protected. Or, at least they should.



1.2.2 Career Opportunities

An even more important sector is national security. Recently, governments have to face a broad range of cyber-threats: global cyber syndicates, hackers for hire, hacktivists, terrorists and state-sponsored hackers.

With critical infrastructure like power plants, trains or dams being controlled by computers, using hacking skills for good has become critical for the safety of nations.



1.2.2 Career Opportunities

Companies and governments need to implement hardware and software defensive systems to protect their digital assets.

Additionally, they also need to train their entire organization to make sure:

- Secure applications are developed,
- Proper defensive measures are taken, and
- That proper use of the company's data is in place.

IT Security is a very difficult game! A way to ensure that a system is secure from cyber-attacks is by **hiring a penetration tester**.

1.2.2 Career Opportunities

Penetration testers (also called pentesters) are professionals who are hired to simulate a hacking attack against a network, a computer system, a web application or the entire organization.

They master the same tools and techniques that malicious hackers use to discover any (and all) vulnerability in the systems they test.



1.2.2 Career Opportunities

These highly skilled professionals often work:

- As freelancers
- In an IT Security services company
- As in-house employees

1.2.2 Career Opportunities

Moreover, as IT is a broad knowledge domain, they can specialize in specific infosec sectors such as:

- Systems attacks
- Web applications
- Malware analysis
- Reverse engineering
- Mobile applications
- Other



1.2.2 Career Opportunities

The demand for penetration testers is on a steady growth.

Being passionate, skilled and hungry for knowledge are fundamental characteristics for a successful pentesting career.

By starting this course, you have made a big step in the right direction! We'll now introduce some of the jargon used by information security professionals.



1.2.3 Information Security Terms

Speaking the **domain language** is **fundamental** in any field. It helps you to better understand the industry and better communicate with your colleagues.

We will now review a list of important terms to know. Keep this chapter as a reference while studying.



1.2.3.1 White Hat Hacker

A **white hat hacker** is a professional penetration tester or ethical hacker who performs authorized attacks against a system helping the client solve their security issues.

White hat hackers do not perform illegal actions.



1.2.3.2 Black Hat Hacker

A **black hat hacker** is a hacker who performs unauthorized attacks against a system with the purpose of causing damage or gaining profit.

There is also a category of black hat hackers called **crackers**.



1.2.3.3 Users and Malicious Users

A **user** is a computer system user. It can be an employee of your client or an external user.

A **malicious user** is a user who misuses or attacks computer systems and applications.

```
21 def initialize_experiment
22   # Create the experiment
23   # @experiment - the experiment will result in
24   # @observations - an array of Observations, in which
25   # @control - the control observation
26
27   # Initialize the experiment
28   def initialize(experiment, observations = [], control = nil)
29     @experiment = experiment
30     @observations = observations
31     @control = control
32     @candidates = observations - [control]
33     evaluate_candidates
34   end
35
36   # Public: Create a new experiment
37   # @experiment - the experiment will result in
38   # @observations - an array of Observations, in which
39   # @control - the control observation
40
41   # Initialize the experiment's context
42   def context
43     @experiment.context
44   end
45
46   # Public: Get the name of the experiment
47   def experiment_name
48     @experiment.name
49   end
50
51   # Public: Get the result a match between all
52   def match
53     @experiment.result
54   end
55 end
```



1.2.3.4 Root or Administrator

The **root** or **administrator** users are the users who manage IT networks or single systems.

They have the maximum privileges over a system.



1.2.3.6 Security Through Obscurity

Security through obscurity is the use of secrecy of design, implementation or configuration in order to provide security.

In this course, you will learn that security through obscurity cannot stop a skilled and motivated attacker.



1.2.3.7 Attack

An **attack** is any kind of action aimed at misusing or taking control over a computer system or application. Some examples of attacks are:

- Getting unauthorized access to an administration area
- Stealing a user's password
- Causing denial of service
- Eavesdropping on communications

```
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```


1.2.3.9 Denial of Service

With a **denial of service (DoS)** attack, a malicious user makes a system or a service unavailable.

The attack could be carried out by making the service crash or by saturating the service resources, thus making it unresponsive for legitimate users.



1.2.3.10 Remote Code Execution

During a **remote code execution** attack, a malicious user manages to execute some attacker-controlled code on a victim remote machine.

Remote code execution vulnerabilities are very dangerous because they can be exploited over the network by a remote attacker.

```
25 def @control: Control = ...
26
27 def @experiment: Experiment = ...
28   // The Experiment will result in ...
29   @observations = an array of Observations, ...
30   @control = the control observation
31
32 def initialize(experiment, observations = [], control = null)
33   @experiment = experiment
34   @observations = observations
35   @control = control
36   @candidates = observations - {control}
37   evaluate_candidates
38
39 def @control: Control = ...
40   // The Control will result in ...
41   @observations = an array of Observations, ...
42   @experiment = the name of the experiment
43
44 def @experiment_name
45   experiment.name
46
47
48 // Publish the result a match between ...
49 def @match
50   // ...
51   @candidates/resultLab 11
```



1.2.3.11 Shell Code

A **shellcode** is a piece of custom code which provides the attacker a shell on the victim machine.

Shellcodes are generally used during remote code execution attacks.

```
21 # @param: context
22 # @param: candidates
23 # @param: result - the Experiment will result in
24 # @param: observations - an array of Observations, in
25 # @param: control - the control observation
26
27 def initialize(experiment, observations = [], control = nil)
28   @experiment = experiment
29   @observations = observations
30   @control = control
31   @candidates = observations - [control]
32   evaluate_candidates
33
34   # Publish the experiment's context
35   def context
36     @experiment.context
37   end
38
39   # Publish the name of the experiment
40   def experiment_name
41     @experiment.name
42   end
43
44   # Publish the result a match between all
45   def match?
46     @candidates == []
47   end
48
49   # Publish the result a match between all
50   def result
51     @candidates[0]
52   end
53 end
```



1.2.3.11 Shell Code

Now that you know a little more about the information security field, it is time to start learning some technical skills!



Cryptography and VPNs



1.3 Cryptography and VPNs

How does this support my pentesting career?

- Understanding how information is transmitted over computer networks
- Choosing the right protocol for the job
- Knowing how to protect your traffic

1.3 Cryptography and VPNs

Why do we introduce Cryptography here?

The main goal of this chapter is to introduce you to concepts that will be useful throughout the course; for instance, accessing our virtual labs.



1.3 Cryptography and VPNs

We will now explain the main difference between clear-text and cryptographic protocols.

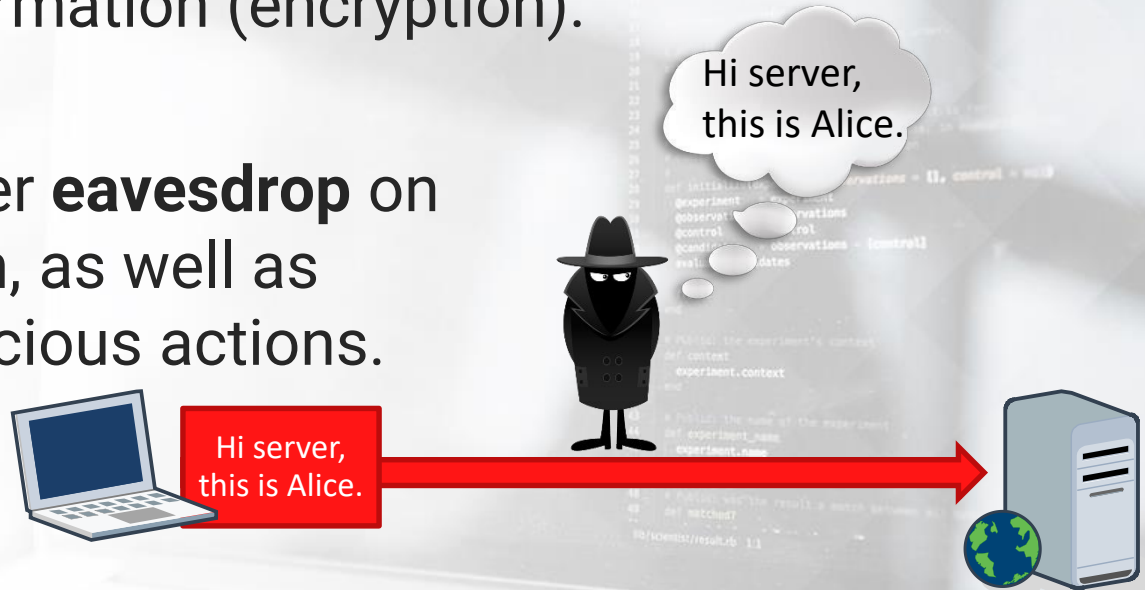
Additionally, you will learn what a VPN (Virtual Private Network) is and how it works. All our virtual labs use VPN so knowing what it is will help you get most of out this course!



1.3.1 Clear-text Protocols

Clear-text protocols transmit data over the network without any kind of transformation (encryption).

This lets an attacker **eavesdrop** on the communication, as well as perform other malicious actions.



1.3.1 Clear-text Protocols

Because of their nature, clear-text protocols are **easy to intercept, eavesdrop and mangle**. They should not be used to transmit critical or private information.

If there is **absolutely no alternative** to a clear-text protocol, you should use it **only on trusted networks**.



1.3.2 Cryptographic Protocols

On the other hand, **cryptographic** protocols transform (encrypt) the information transmitted to protect the communication.

Cryptographic protocols have many different goals. One of them is to **prevent eavesdropping**.



1.3.2 Cryptographic Protocols

If an attacker intercepts the traffic, they will not be able to understand it.



1.3.2 Cryptographic Protocols

If you need to transmit private information, for example - a username and a password, you should always **use a cryptographic protocol** to protect the communication over the network.

What if you need to run a clear-text protocol on an untrusted network?



1.3.2 Cryptographic Protocols

You can wrap (**tunnel**) a clear-text protocol into a cryptographic one.



1.3.2 Cryptographic Protocols

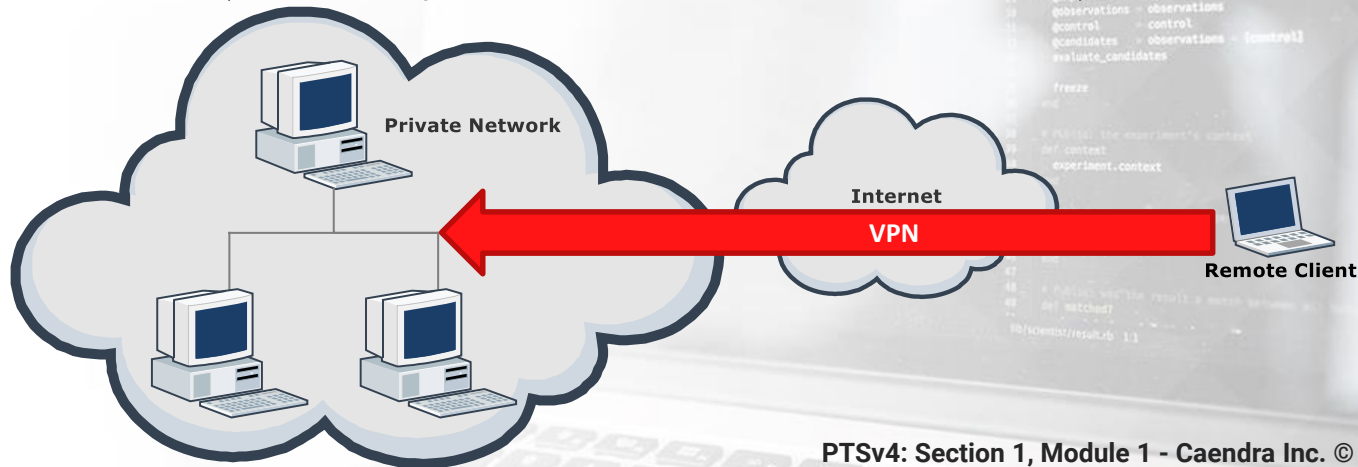
A great example of protocol tunneling is a **VPN**.



1.3.3 Virtual Private Networks

A **Virtual Private Network (VPN)** uses cryptography to extend a private network over a public one, like the Internet.

The extension is made by performing a protected connection to a private network (*such as your office or home network*).



1.3.3 Virtual Private Networks

From the client point of view, being in the VPN **is the same as being directly connected** to the private network.

For example, when you launch a *Hera Lab* scenario from your member's area, a VPN tunnel is created, letting you connect directly to the lab network.



1.3.3 Virtual Private Networks

When you are connected via VPN, you are actually running the very same protocols of the private network.

This lets you perform even low-level network operations. For example, you can use a packet sniffer like **Wireshark**.



Wireshark Introduction



1.4 Wireshark Introduction

Wireshark is a network sniffer tool. A sniffer allows you to see the data transmitted over the network to and from your computer.

Wireshark will be discussed in depth in the next modules. For now, we are going to see its basic usage just to understand the difference between clear-text and cryptographic protocols.



1.4.1 Video – HTTP and HTTPS Traffic Sniffing

In the following video, you will see how to use *Wireshark* to intercept communication between your computer and a web server placed inside a dedicated network within Hera Lab.

This will help you understand the difference between a clear-text protocol and a cryptographic one from an attacker point of view.



1.4.1 Video

HTTPS and HTTPS Traffic Sniffing

In this video, you will see how to use *Wireshark* to intercept communication between your computer and a web server placed inside a dedicated network within Hera Lab.

This will help you understand the difference between a clear-text protocol and a cryptographic one from an attacker point of view.



**Videos are only available in Full or Elite Editions of the course. To upgrade, click [HERE](#). To access, go to the course in your members area and click the resources drop-down in the appropriate module line.*

1.4.2 Hera Lab – HTTP and HTTPS Traffic Sniffing

Now it's time to practice what you have just learned in a hands-on lab!

This lab is the only one that follows the same steps of a video or a lesson. We created it so you can become familiar with the Hera Lab environment.

You will find additional information and instructions on the next slide.

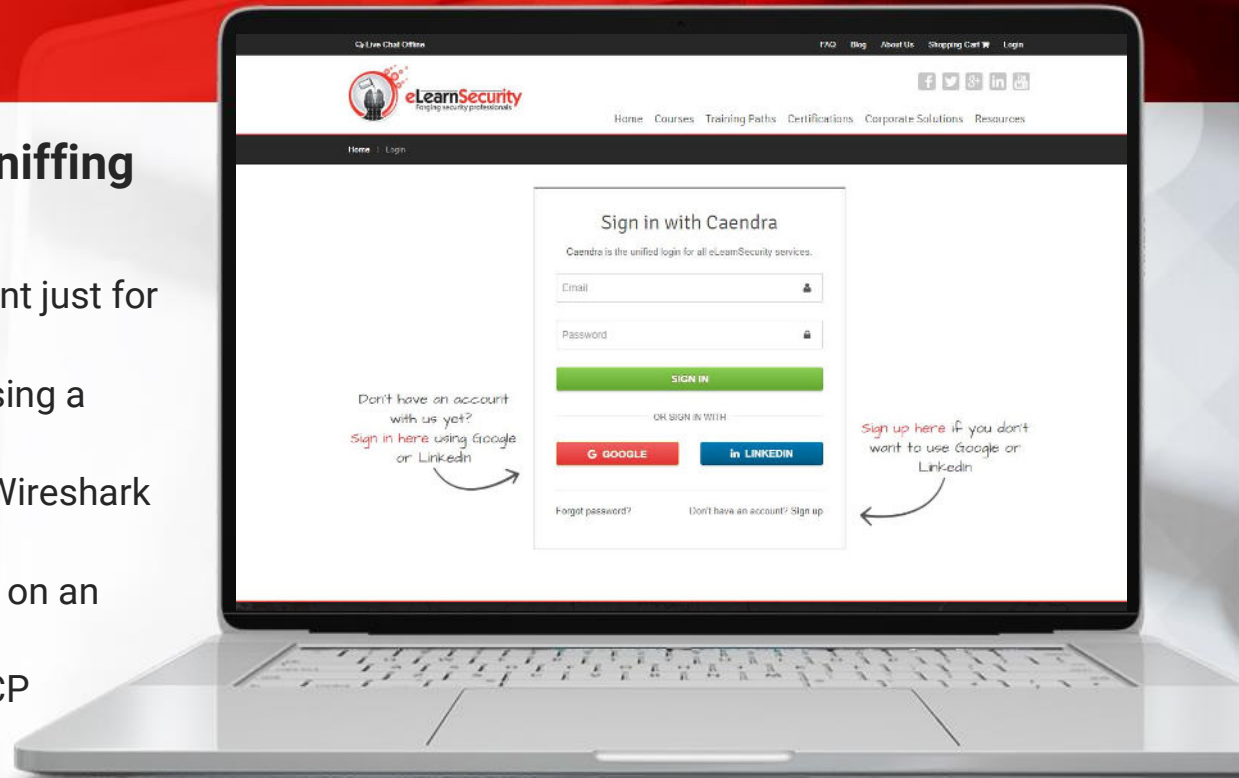


1.4.2 Hera Lab

HTTP and HTTPS Traffic Sniffing

In this lab you will:

- Start a dedicated lab environment just for you
- Login to a web application by using a username and a password
- Sniff the login credentials with Wireshark on a clear-text communication
- Try to sniff the login credentials on an encrypted communication
- Learn how to use the "Follow TCP stream" command



**Labs are only available in Full or Elite Editions of the course. To upgrade, click [HERE](#). To access, go to the course in your members area and click the labs drop-down in the appropriate module line or to the virtual labs tabs on the left navigation.*

Binary Arithmetic Basics



1.5 Binary Arithmetic Basics

How does this support my pentesting career?

- Computers represent data in binary format
- Network addressing
- Computer logic operation

1.5 Binary Arithmetic Basics

Computers represent any kind of data with just two symbols:

0 (zero)

1 (one)

In this section, you will see what a **binary number** is and how to convert a decimal number into binary format.

1.5.1 Decimal and Binary Bases

Decimal notation uses ten symbols (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) to represent numbers while **binary** notation uses only two symbols (0, 1).

Q

How can you represent "big" numbers by using just two symbols?

A

You can do so by using the same method that you use with base-ten. The only difference is the number of symbols at your disposal.

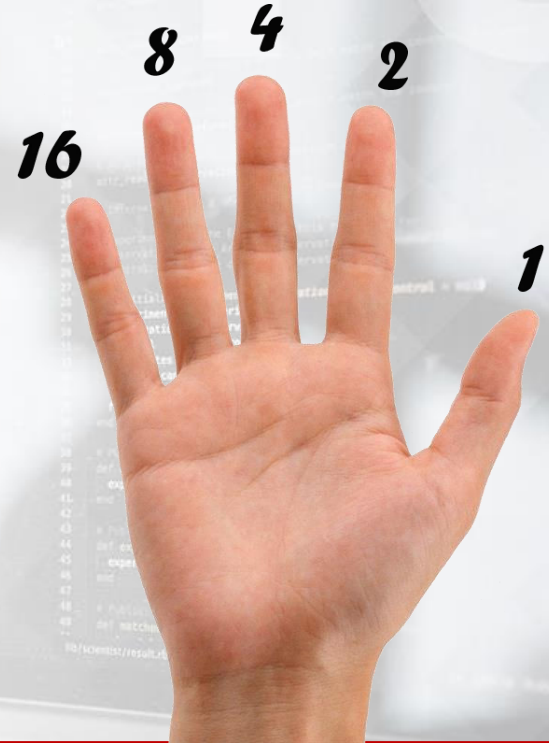
```
20 def initialize_experiment
21   @experiment = Experiment.new
22   @observations = []
23   @control = Control.new
24   @candidates = []
25 end
26
27 def initialize(experiment, observations = [], control = nil)
28   @experiment = experiment
29   @observations = observations
30   @control = control
31   @candidates = observations + [control]
32 end
33
34 def context
35   @experiment.context
36 end
```



1.5.1 Decimal and Binary Bases

You can use the same method in binary:

- You start counting from 0, the first symbol.
- When you reach 1, which is the last symbol, you increment the digit to the left of it.



1.5.1 Decimal and Binary Bases

$$1 + 1 = 10$$

- You have to increment 1, so you "add" a digit on the left and start back from 0

$$111 + 1 = 1000$$

- Here you increment the rightmost digit, then you have to increment the next and so on.



1.5.1 Decimal and Binary Bases

Counting

Decimal	Binary
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010

Last symbol

Last symbol



1.5.2 Converting from and to Binary

How do you convert **from binary to decimal format**?

You can use the **position** of the digits.

- $293_{10} = 3 \cdot 10^0 + 9 \cdot 10^1 + 2 \cdot 10^2$

#	10^0	10^1	10^2
293	3	9	2

1.5.2 Converting from and to Binary

You can use the same method in binary, the only difference is the base.

- $1101^2 = 1*2^0 + 0*2^1 + 1*2^2 + 1*2^3 = 13_{10}$

#	2^0	2^1	2^2	2^3
1101	1	0	1	1

1.5.2 Converting from and to Binary

To convert a decimal number into binary format, you have to:

- Divide it by 2 and keep a note of the remainder.
- Then, you do it again dividing the result of the previous step by 2. Keep a note of the remainder (0 or 1).
- Iterate the same operation until the dividend is zero.

EXAMPLE

1.5.2.1 Converting from Binary Example

$$13_{10} = ???_2$$

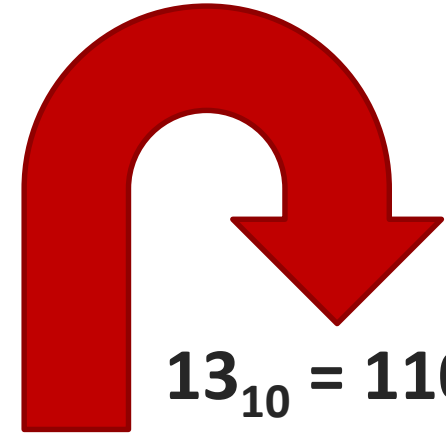
- $13 / 2 = 6$
- $6 / 2 = 3$
- $3 / 2 = 1$
- $1 / 2 = 0$

remainder: 1

remainder: 0

remainder: 1

remainder: 1



$$13_{10} = 1101_2$$



1.5.3 Bitwise operations

Now that you know how binary representation works let's look at some basic low-level operations.



1.5.3 Bitwise operations

A computer can directly manipulate bits by performing **bitwise operations**, which are used a lot in network programming and assembly programming.



1.5.3.1 NOT

NOT is a simple operation that flips the bits; zeroes become ones and ones become zeroes.

NOT works on a single number.



```
NOT 1101 = 0010
```



1.5.3.2 AND

AND performs a **Logical AND** between the bits of its operands.

If both bits in the comparing position are ones, the result is one; otherwise, it is zero.



```
1001 AND 1100 = 1000
```



1.5.3.3 OR

OR performs a **Logical OR** between the bits of its operands.

If **at least** one of the bits in the comparing position is one, the result is one.



$$1001 \text{ OR } 1100 = 1101$$



1.5.3.4 XOR

XOR performs a **Logical Exclusive OR** between the bits of its operands.

If **just one** of the bits in the comparing position is one, the result is one; otherwise, it is zero.

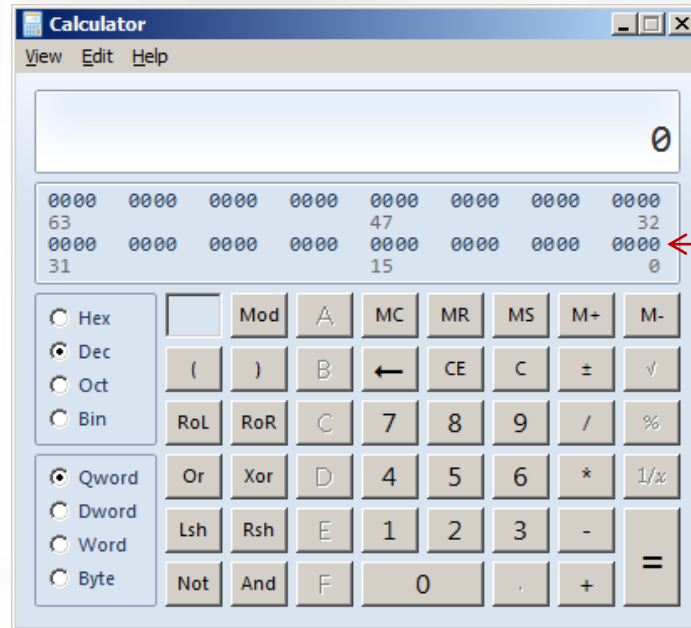


```
1001 XOR 1100 = 0101
```



1.5.4 Calculator

You can use a common calculator application and set the mode to "Programmer" to work in binary mode.

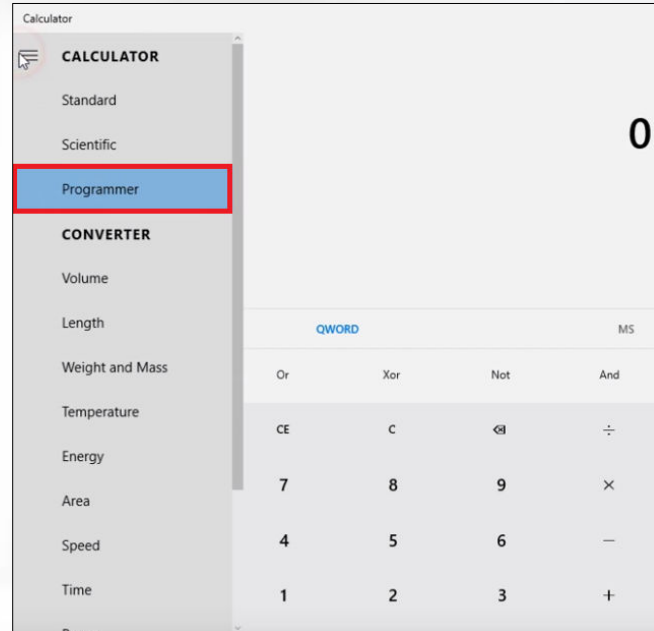


Click on a bit to flip it.

From 0 to 1 and vice-versa.

1.5.4 Calculator

The same on
Windows 10...



1.5.4 Calculator

You can also use your fingers to count and perform operations in binary mode.

Check out this [tutorial!](http://www.mathsisfun.com/numbers/binary-count-fingers.html)

1.5.5 Hexadecimal arithmetic

Numbers can also be presented in a format other than decimal or binary system. Another system that is widely used in computer science is the **hexadecimal system**.

This system works the same way as the binary or decimal system. Let's take a look at the diagram on the next slide.



1.5.5 Decimal and Hexadecimal Bases

Counting

Hexadecimal

0
1
2
3
4
5
6
7
8
9
A
B
C
D
E
F
10

Decimal

0
1
2
3
4
5
6
7
8
9
10

Last symbol

Last symbol

1.5.5 Hexadecimal arithmetic

Remember the last symbol? For a binary system, it is 1, while in decimal, it is 9. If we follow this format, then in hexadecimal, the maximal symbol is 15.

Since higher numbers are always built out of multiple digits, to avoid confusion, all double-digit numbers were switched to letters. Thus, we count 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

1.5.5 Hexadecimal arithmetic

In this case, the maximal digit is „F” (15 in decimal).

You probably noticed that not all hexadecimal numbers contain letters, so in order to distinguish them from decimal, we add „0x” at the beginning or „h” at the end.

Exemplary numbers may then look similar to those: 0x10 or 44h.

1.5.5 Hexadecimal arithmetic

You can convert hexadecimal to decimal and reverse in a similar manner as you did with binary.

Let's now see how it's possible to convert a hexadecimal number to decimal one. The following slides will guide you through the process.



1.5.5.1 Converting hexadecimal to decimal

We will be working with the exemplary number 0x3a1, which can also be written as 3a1h.

Basically, to inform that the given number is hexadecimal, we use „0x” at the beginning or „h” at the end.

First, we need to understand the target number’s decimal representation; so, we can write 3a1h as $(3 \cdot 10^2 + 10 \cdot 10^1 + 1 \cdot 10^0)$ h since „A” is „10” in decimal. This format will be helpful in further calculations.

1.5.5.1 Converting hexadecimal to decimal

How do you convert **from hexadecimal to decimal format?**

You can use the **position** of the digits.

- $0x3a1 = 0x(3\ 10\ 1)$
- $0x3a1 = 1*16^0 + 10*16^1 + 3*16^2 = 929_{10}$

#	16^0	16^1	16^2
3a1	1	a (10)	3

1.5.5.2 Converting decimal to hexadecimal

In order to convert a decimal number to a hexadecimal one, we will perform subsequent divisions by 16 (system base) and note down the remainders, as per below picture:

number	div by	result	hexadecimal
1019	16	63.6875	$0.6875 * 16 = 11$ (B)
63	16	3.9375	$0.9375 * 16 = 15$ (F)
3	16	0.1875	$0.1875 * 16 = 3$
0	Can't divide 0	-	Result is 0x3FB



1.5.5.2 Converting decimal to hexadecimal

First, we take 1019 and divide by 16 (system base), and we receive **63,6875**.

We note down **63** for further calculation, and use **0.6875** to calculate the last digit of result hexadecimal number.

Let's multiply **0.6875** by the system base (16), and the result is **11 (B in HEX)**.

B is then the **last digit of result hexadecimal number**.

1.5.5.2 Converting decimal to hexadecimal

Like in the previous slides, we'll do similarly with the noted **63**. Let's start by dividing it by **16** to receive **3,9375**.

Let's note down **3** for further calculations and use **0.9375** to get the second digit of result for our hexadecimal number.

Let's multiply **0.9375** by the base (**16**), and we receive **15 (F in HEX)**.

F will then be the next digit. **So far we have the following results for the last digits of our hexadecimal number – „FB”.**

1.5.5.2 Converting decimal to hexadecimal

Let's now turn our attention to the number we just noted down, **3**. When we divide it by **16**, we receive **0.1875**.

By proceeding in the same way as we have previously, we should note down **zero** for further division, but since it is not possible to divide **0**, we know that we are currently calculating the last digit for the result of our hexadecimal number.

0.1875 will let us know the last digit of result if we again follow previous instruction.

Multiplying the above value by **16** allows to obtain last digit: **3**.

1.5.5.2 Converting decimal to hexadecimal

Looking at the calculations, now we know result number: **0x3FB** which is hexadecimal form of decimal 1019.



1.5.5.3 Automated converting

You now know how to recognise hexadecimal numbers, and how to convert them to decimal form, in addition to converting decimal numbers to its hexadecimal form.

But during penetration testing work, you might want to speed things up. If so, then it's best to use converters like online resources. For example, you can check following websites:

- <https://www.binaryhexconverter.com/decimal-to-hex-converter>
- <https://www.binaryhexconverter.com/hex-to-decimal-converter>



Conclusion



Congratulations!

You have just finished your first module of the **Penetration Testing Student** course.

In the next modules, you are going to deepen your knowledge of cryptography, protocols, computer networks, penetration testing and much more!





References



References

[Captain Crunch](https://en.wikipedia.org/wiki/John_Draper)

https://en.wikipedia.org/wiki/John_Draper

[Biography of Kevin Mitnick](https://www.mitnicksecurity.com/about/kevin-mitnick-worlds-most-famous-hacker-biography)

<https://www.mitnicksecurity.com/about/kevin-mitnick-worlds-most-famous-hacker-biography>

[Richard Stallman](https://stallman.org/)

<https://stallman.org/>

[The Conscience of a Hacker](http://phrack.org/issues/7/3.html)

<http://phrack.org/issues/7/3.html>



References

[Wireshark](https://www.wireshark.org/)

<https://www.wireshark.org/>

[OpenVPN stable release](http://build.openvpn.net/downloads/releases/latest/)

<http://build.openvpn.net/downloads/releases/latest/>

[Binary fingers](http://www.mathsisfun.com/numbers/binary-count-fingers.html)

<http://www.mathsisfun.com/numbers/binary-count-fingers.html>

[Binary hex converter – Decimal to hexadecimal](https://www.binaryhexconverter.com/decimal-to-hex-converter)

<https://www.binaryhexconverter.com/decimal-to-hex-converter>



References

[Binary hex converter – Hexcidecimal to decimal](https://www.binaryhexconverter.com/hex-to-decimal-converter)

<https://www.binaryhexconverter.com/hex-to-decimal-converter>



Videos

HTTP(s) Traffic Sniffing & Wireshark Introduction

In this video, you will see how to use *Wireshark* to intercept communication between your computer and a web server placed inside a dedicated network within Hera Lab.

This will help you understand the difference between a clear-text protocol and a cryptographic one from an attacker point of view.

**Videos are only available in Full or Elite Editions of the course. To upgrade, click [HERE](#). To access, go to the course in your members area and click the resources drop-down in the appropriate module line.*



Labs

HTTP(S) Traffic Sniffing

Intercept traffic with Wireshark. Learn how to “Follow TCP Stream”.



**Labs are only available in Full or Elite Editions of the course. To upgrade, click [HERE](#). To access, go to the course in your members area and click the labs drop-down in the appropriate module line or to the virtual labs tabs on the left navigation.*