

# Penetration Testing Student

# Penetration Testing

Section 01 | Module 04

© Caendra Inc. 2019  
All Rights Reserved

# Table of Contents

## Module 04 | Penetration Testing

---

- 4.1 Introduction
- 4.2 Lifecycle of a penetration test
- 4.3 References



# Learning Objectives

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

- Penetration testing general terms and conditions
- How to approach a penetration testing engagement from legal and organizational standpoint



# Introduction



# 4.1 Introduction

In this module, you will learn the basic principles of **Penetration Testing**.

A penetration tester, much like an experienced hacker, performs a deep investigation of the remote system's security flaws. This activity requires methodology and skills!

## 4.1 Introduction

Penetration testers, unlike hackers, must **test for any and all vulnerabilities**, not just the ones that may grant them root access to a system. **Penetration testing is not about getting root!**

Furthermore, Penetration Testers cannot destroy their client's infrastructure; professional pentesting requires a thorough understanding of attack vectors and their potential.

```
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
```

## 4.1 Introduction

In this module, you will see the **process** behind a penetration test, starting from the initial engagement to the final report.

This module will help you better understand the scope of attacks and techniques you will learn during the *Penetration Testing* section of this course!



# Lifecycle of a Penetration Test





## 4.2 Lifecycle of a Penetration Test

### How does this support my pentesting career?

- Become a real pentester, not just a skilled hacker
- Understand the role of a penetration test in a corporate environment
- Be able to perform effective pentests



## 4.2 Lifecycle of a Penetration Test

A Penetration Test is both a **complex** and a very **delicate** process.

You have to thoroughly test your client's systems to find **any and every vulnerability** while, at the same time, you must guarantee that your activity will have the least impact possible on the production systems and services; this is crucial and is the difference between a **real professional** and an amateur.

## 4.2 Lifecycle of a Penetration Test

It is important to carefully select the right tools and techniques to use during your tests to avoid **overloading your client systems and networks**.

Deep understanding of what you are doing also allows you to communicate to your client what steps to take should anything go wrong during the pentest.

```
24 # The experiment will result in a list of observations
25 # observations - an array of Observations, in this case
26 # control - the control observation
27
28 def initialize(experiment, observations = [], control = nil)
29   @experiment = experiment
30   @observations = observations
31   @control = control
32   @candidates = observations - [control]
33   evaluate_candidates
34
35   freeze
36
37   @context =
38     {
39       experiment_name:
40         experiment.name
41     }
42
43   # A final step: the result is a match between an observation
44   def matched?
45     !!@candidates[result.sub 1]
46   end
47
48   # A final step: the result is a match between an observation
49   def matched?
```



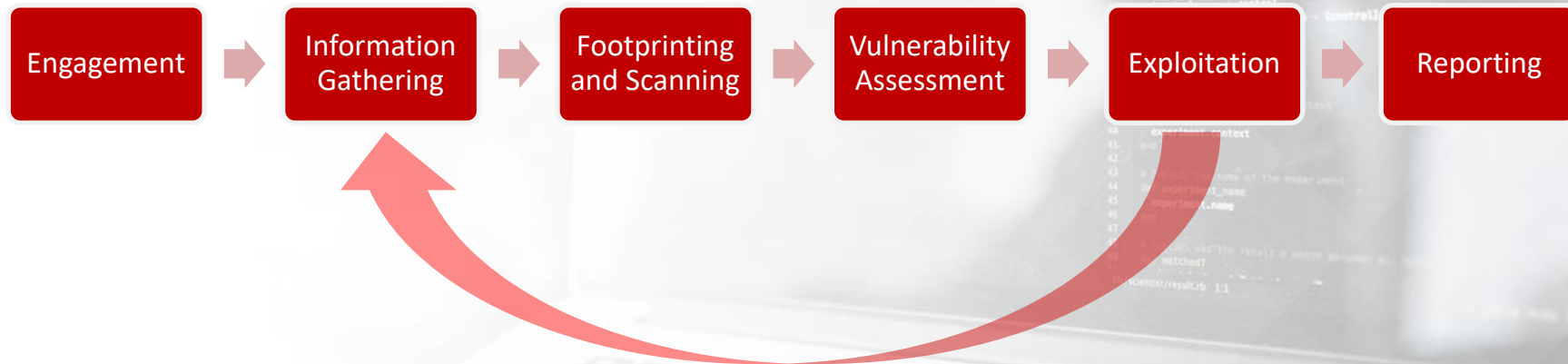
## 4.2 Lifecycle of a Penetration Test

Considering the penetration test as a **process**, rather than an unstructured block of tasks, ensures that every potential vulnerability or security weakness gets tested, with the lowest possible overhead.

As you will see in a moment, the success of a task depends on the success of the preceding tasks.

## 4.2 Lifecycle of a Penetration Test

Let's now look at every phase of the **penetration testing process**. Do not underestimate the value of every step!



### 4.2.1 Engagement

All the details about the penetration test are established during the **Engagement** phase.

[illegible]

### 4.2.1.1 Quotation

At the **Quotation** stage, a professional pentester defines the fee for the penetration test of a network, a web application or the whole organization.

The fee will vary according to:

- Type of engagement (Black Box, Gray Box, etc.)
- How time-consuming the engagement is
- The complexity of the applications and services in scope
- The number of targets (IP addresses, domains, etc.)

```

19 # Create a new experiment
20 exp = Experiment()
21 # Create a new experiment
22 exp = Experiment()
23 # Create a new experiment
24 exp = Experiment()
25 # Create a new experiment
26 exp = Experiment()
27 # Create a new experiment
28 exp = Experiment()
29 # Create a new experiment
30 exp = Experiment()
31 # Create a new experiment
32 exp = Experiment()
33 # Create a new experiment
34 exp = Experiment()
35 # Create a new experiment
36 exp = Experiment()
37 # Create a new experiment
38 exp = Experiment()
39 # Create a new experiment
40 exp = Experiment()
41 # Create a new experiment
42 exp = Experiment()
43 # Create a new experiment
44 exp = Experiment()
45 # Create a new experiment
46 exp = Experiment()
47 # Create a new experiment
48 exp = Experiment()
49 # Create a new experiment
50 exp = Experiment()
51 # Create a new experiment
52 exp = Experiment()
53 # Create a new experiment
54 exp = Experiment()
55 # Create a new experiment
56 exp = Experiment()
57 # Create a new experiment
58 exp = Experiment()
59 # Create a new experiment
60 exp = Experiment()
61 # Create a new experiment
62 exp = Experiment()
63 # Create a new experiment
64 exp = Experiment()
65 # Create a new experiment
66 exp = Experiment()
67 # Create a new experiment
68 exp = Experiment()
69 # Create a new experiment
70 exp = Experiment()
71 # Create a new experiment
72 exp = Experiment()
73 # Create a new experiment
74 exp = Experiment()
75 # Create a new experiment
76 exp = Experiment()
77 # Create a new experiment
78 exp = Experiment()
79 # Create a new experiment
80 exp = Experiment()
81 # Create a new experiment
82 exp = Experiment()
83 # Create a new experiment
84 exp = Experiment()
85 # Create a new experiment
86 exp = Experiment()
87 # Create a new experiment
88 exp = Experiment()
89 # Create a new experiment
90 exp = Experiment()
91 # Create a new experiment
92 exp = Experiment()
93 # Create a new experiment
94 exp = Experiment()
95 # Create a new experiment
96 exp = Experiment()
97 # Create a new experiment
98 exp = Experiment()
99 # Create a new experiment
100 exp = Experiment()

```



## 4.2.1.1 Quotation

Evaluating and quoting these aspects requires experience that you will gain in the field.

If you are not able to quantify the amount of work required by an engagement, you can provide an hourly fee.

```
25 # @experiment - the experiment result is a dict
26 # @observations - an array of Observations, in experiment
27 # @control - the control observation
28
29 def initialize(experiment, observations = [], control = null)
30   @experiment = experiment
31   @observations = observations
32   @control = control
33   @candidates = observations - [control]
34   evaluate_candidates
35
36   freeze
37
38   @context =
39     experiment.context
40
41   @experiment_name =
42     experiment.name
43
44   nil
45
46   # @match? - did the result a match between an experiment
47   def match?
48     # ...
49   end
50
51   @experiment.result.rb 1.1
```



## 4.2.1.2 Proposal Submittal

The best way to win a job is by providing a **sound and targeted proposal**.

You should write the proposal keeping in mind the client's **needs and infrastructure**.

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



## 4.2.1.2 Proposal Submittal

The proposal should include:

- The understanding of the client's needs. In other words, what you understood of their requirements.
- The approach and methodology you want to use, like the use of automated scanning tools, manual testing, onsite testing and any other information that fits.

## 4.2.1.2 Proposal Submittal

Furthermore, it should also include:

- How you want to address their needs and what kind of value the pentest will bring to their business. Think in terms of **risks and benefits**, like business continuity, improved confidentiality, avoidance of money and reputation loss due to data breaches.
- A quotation in terms of price and an estimate of the time required to perform your job.

## 4.2.1.2 Proposal Submittal

Finally, any proposal must address:

- The type of engagement. Is your activity a penetration test or vulnerability assessment? Is it remote or onsite? And so on.
- **The scope of engagement** in terms of IP addresses, network blocks, domain names or any other information useful in defining the scope.

## 4.2.1.3 Staying in Scope

As a professional penetration tester, you should be aware that your client might not have enough knowledge of some IT areas, especially when communicating the target to you.



## 4.2.1.3 Staying in Scope

You should always make sure that the target of your engagement is the property of your client. Be careful especially when asked to perform an engagement (e.g., on a single website).

If it is a part of shared hosting, you **must not** conduct an assessment on such a target unless you are given written permission from the hosting provider.







## 4.2.1.4 Incident Handling

When conducting a penetration test, you should take into consideration that **incidents happen**.

An **incident** is an unplanned and unwanted situation that affects the client's environment and disrupt its services.

## 4.2.1.4 Incident Handling

Even when sticking to all of the best practices and performing every test very carefully, **there is always a likelihood of damaging the tested assets**, especially when you have little knowledge about the tested environment and cannot predict the result of every single operation.



## 4.2.1.4 Incident Handling

**You should always aim not to damage the target.**

In case of planning some intensive or risky tests, **you might want to communicate with the customer.** For instance, if there are some preferred hours when possible service stoppage will be less painful to them.

## 4.2.1.4 Incident Handling

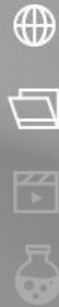
It is a best practise to have an **incident handling procedure**.

Many large organisations already have such processes set up, while the smaller ones might not have implemented such procedures within them.



## 4.2.1.4 Incident Handling

An **incident handling procedure** is a set of instructions that need to be executed by both you and your customer on how to proceed when an **Incident** (e.g., service damage or unavailability) occurs.



## 4.2.1.4 Incident Handling

If there is no fixed procedure established by the client, the simplest way to handle an incident is to **have an emergency contact**, a technical person on the client's site **that is available** (via phone or another form of contact) that might coordinate further **incident handling** for the customer's company.





## 4.2.1.4 Incident Handling

Once the emergency contact is set, it is worth adding a statement to the **rules of engagement**:

*In case of technical inquiries regarding the target assets, Pentester will contact [bob@itservice.corp](mailto:bob@itservice.corp). In the event of suspecting that a major incident took place (e.g., service unavailability), Pentester will immediately contact Bob of IT Service at phone number [+12 345 678 90](tel:+1234567890)*

## 4.2.1.5 Legal Work

Once the previous steps are completed, you have to deal with the legal responsibilities of each party involved; this is done by producing some legal paperwork.

Sometimes you will need to involve a lawyer as information security laws vary a lot from country to country. Other times, professional insurance is required, and it is strongly advised to have it as it only costs a few hundred dollars per year and can turn out to be very useful just in case.

## 4.2.1.5 Legal Work

Companies usually want you to sign one or more Non-Disclosure Agreements (NDAs). These documents enforce your full confidentiality regarding any information or confidential data you may come across during your engagement.

It does not matter if you have been exposed to private data, information on secret processes or products, it is your duty to **keep them private and encrypted on your PC.**

## 4.2.1.5 Legal Work

With an NDA, a company ensures that you will not divulge any confidential information to any third party. Confidentiality is just one of the legal aspects of pentesting. Another key point is outlining what you **can and cannot do**.

**All of the steps seen thus far apply if you are a Freelance Penetration tester. If you work for an IT Security services company, the legal department will deal with it, and your penetration testing process will start from the next step.**

## 4.2.1.5 Legal Work

**Rules of Engagement** is another document that will define the scope of engagement and will put on paper what you are entitled to do and when; this includes the time window for your tests and your contacts in the client's organization.



## 4.2.1.5 Legal Work

You will want these contacts (client's employees or managers) to coordinate activities, or to promptly communicate with if you accidentally break something during your tests.

Once everything is clearly documented, you can move on to the practical part of the engagement, starting from information gathering.

## 4.2.2 Information Gathering

**Information gathering** is the first and one of the most fundamental stages of a successful penetration test.

Most beginners tend to overlook or rush this phase. If you want to perform an effective pentest; do not do that!





## 4.2.2.1 General Information

Information gathering can start once the legal paperwork is complete but **not before** the beginning of the testing period. You don't want the client to find anything in their logs before that start date.

During this stage, you are an investigator who wants to harvest information about the client's company.

## 4.2.2.1 General Information

Such information includes:

- Board of directors
- Investors
- Managers and employees
- Branch location and addresses

Names and email addresses

The above information is extremely useful if **Social Engineering** is allowed by the rules of engagement, as you will be able to mount effective targeted attacks.

## 4.2.2.2 Understanding the Business

As the goal of a penetration test is to mimic the effects of a black hat hacker attack, you need to understand what are the risks involved and what are the client's critical infrastructures.

Having an understanding of the business is a key aspect in understanding what is important for your client; this allows you to know what is critical and vital for the client, thus allowing you to rate the risks associated with a successful attack.

### 4.2.2.3 Infrastructure Information Gathering

After collecting the *General Information* and you have an *Understanding of the Business*, the **Infrastructure Information Gathering** can begin.

In this phase, you transform the IP addresses or the domains in scope into actionable information about servers, operating systems and much more.

## 4.2.2.3 Infrastructure Information Gathering

If the scope is defined as a list of IP addresses, you can move on to the next step.

If the scope is the whole company or some of their domains, you will have to harvest the relevant IP blocks by using **WHOIS** and other **DNS information**.

```
20 def __init__(self):
21     self.scope = None
22     self.candidates = None
23     self.experiment = None
24     self.observations = None
25     self.control = None
26     self.evaluate_candidates = None
27
28 def initialize(self, scope=None, control=None):
29     self.scope = scope
30     self.observations = observations
31     self.control = control
32     self.candidates = None
33     self.evaluate_candidates = None
34
35 def run(self):
36     self.initialize()
37     self.candidates = self.harvest_candidates()
38     self.evaluate_candidates()
39
40 def harvest_candidates(self):
41     """Harvest the candidates from the scope"""
42     # TODO: Harvest the candidates from the scope
43     # TODO: Harvest the candidates from the scope
44     # TODO: Harvest the candidates from the scope
45     # TODO: Harvest the candidates from the scope
46     # TODO: Harvest the candidates from the scope
47
48 def evaluate_candidates(self):
49     """Evaluate the candidates"""
50     # TODO: Evaluate the candidates
51     # TODO: Evaluate the candidates
52     # TODO: Evaluate the candidates
53     # TODO: Evaluate the candidates
54     # TODO: Evaluate the candidates
55
56 def __str__(self):
57     return f"Experiment: {self.experiment}, Observations: {self.observations}, Control: {self.control}, Candidates: {self.candidates}"
58
59 if __name__ == '__main__':
60     experiment = Experiment()
61     experiment.run()
```



## 4.2.2.3 Infrastructure Information Gathering

The goal of this phase is to give **meaning to every IP address in scope** by determining:

- If there is a live host or server using it.
- If there are one or more websites using that IP address.
- What OS is running on the host or the server.

```
24 # def initialize(experiment, observations = [], control = null)
25 #   @experiment = experiment
26 #   @observations = an Array of Observations, in ascending order
27 #   @control = the control Observation
28 # end
29
30 # def initialize(experiment, observations = [], control = null)
31 #   @experiment = experiment
32 #   @observations = observations
33 #   @control = control
34 #   @candidates = observations - [control]
35 #   evaluate_candidates
36 # end
37
38 # freeze
39 # def context
40 #   experiment.context
41 # end
42 # def name
43 #   # build the name of the experiment
44 #   def experiment_name
45 #     experiment.name
46 #   end
47 # end
48 # def candidates
49 #   # build the candidates
50 #   candidates = observations - [control]
```



## 4.2.2.3 Infrastructure Information Gathering

This will help you:

- Focus your efforts to actual live clients and servers.
- Target your attacks.
- Sharpen your tools for the exploitation phase, when you have to find out the vulnerabilities and the exploitability of the client systems.

```
14 def initialize_experiment(experiment, observations = [], control = null)
15   # Create the experiment's context
16   # Create the experiment's context
17   # Create the experiment's context
18   # Create the experiment's context
19   # Create the experiment's context
20   # Create the experiment's context
21   # Create the experiment's context
22   # Create the experiment's context
23   # Create the experiment's context
24   # Create the experiment's context
25   # Create the experiment's context
26   # Create the experiment's context
27   # Create the experiment's context
28   # Create the experiment's context
29   # Create the experiment's context
30   # Create the experiment's context
31   # Create the experiment's context
32   # Create the experiment's context
33   # Create the experiment's context
34   # Create the experiment's context
35   # Create the experiment's context
36   # Create the experiment's context
37   # Create the experiment's context
38   # Create the experiment's context
39   # Create the experiment's context
40   # Create the experiment's context
41   # Create the experiment's context
42   # Create the experiment's context
43   # Create the experiment's context
44   # Create the experiment's context
45   # Create the experiment's context
46   # Create the experiment's context
47   # Create the experiment's context
48   # Create the experiment's context
49   # Create the experiment's context
50   # Create the experiment's context
51   # Create the experiment's context
52   # Create the experiment's context
53   # Create the experiment's context
54   # Create the experiment's context
55   # Create the experiment's context
56   # Create the experiment's context
57   # Create the experiment's context
58   # Create the experiment's context
59   # Create the experiment's context
60   # Create the experiment's context
61   # Create the experiment's context
62   # Create the experiment's context
63   # Create the experiment's context
64   # Create the experiment's context
65   # Create the experiment's context
66   # Create the experiment's context
67   # Create the experiment's context
68   # Create the experiment's context
69   # Create the experiment's context
70   # Create the experiment's context
71   # Create the experiment's context
72   # Create the experiment's context
73   # Create the experiment's context
74   # Create the experiment's context
75   # Create the experiment's context
76   # Create the experiment's context
77   # Create the experiment's context
78   # Create the experiment's context
79   # Create the experiment's context
80   # Create the experiment's context
81   # Create the experiment's context
82   # Create the experiment's context
83   # Create the experiment's context
84   # Create the experiment's context
85   # Create the experiment's context
86   # Create the experiment's context
87   # Create the experiment's context
88   # Create the experiment's context
89   # Create the experiment's context
90   # Create the experiment's context
91   # Create the experiment's context
92   # Create the experiment's context
93   # Create the experiment's context
94   # Create the experiment's context
95   # Create the experiment's context
96   # Create the experiment's context
97   # Create the experiment's context
98   # Create the experiment's context
99   # Create the experiment's context
100  # Create the experiment's context
```



## 4.2.2.4 Web Applications

If there is any web application in scope, in this phase you will harvest:

- Domains
- Subdomains
- Pages (website crawling)
- Technologies in use, like PHP, Java, .NET and so on.
- Frameworks and content management systems in use, like Drupal, Joomla, Wordpress, and others.

## 4.2.2.4 Web Applications

You should treat web applications as completely separate entities, that require a separate study.

You can gather information about web applications by browsing and inspecting through application proxies such as Burp.



## 4.2.3 Footprinting and Scanning

During the **Footprinting and Scanning** phase, you deepen your knowledge of the in-scope servers and services.



## 4.2.3.1 Fingerprinting the OS

Fingerprinting the Operating System of a host not only gives you information about the OS running on the system, but also helps you narrow down the number of potential vulnerabilities to check in the next phases.

*You would never check for a typical MS Windows vulnerability on a Linux host!*



## 4.2.3.1 Fingerprinting the OS

There are tools that can make educated guesses about the OS, the version and even the patch level of a remote system.

Those tools exploit some singularities you can find in the network stack implementation of every operating system.

```
def skillsize(experiment, observations = [], control = null)
  26
  27
  28
  29 @experiment = experiment
  30 @observations = observations
  31 @control = control
  32 @candidates = observations - !control
  33 evaluate_candidates
  34
  35 freeze
  36 end
  37
  38 # PANDA: the experiment's context
  39 def context
  40   @context
  41 end
  42
  43 # PANDA: the result of the experiment
  44 def experiment
  45   @experiment
  46 end
  47
  48 # PANDA: was the result a match between all
  49 def match?
  50   @match
  51 end
  52
  53 PANDA/result: 1.1
```

## 4.2.3.2 Port Scanning

After having detected and fingerprinted the live hosts, it's time for **port scanning**!

With a scan of live hosts, you can determine which **ports** are open on a remote system; this is a crucial phase of the engagement because any mistake made here will impact the next steps.

## 4.2.3.2 Port Scanning



Currently, the de facto port scanner is **nmap**. With nmap, a penetration tester can exploit different scanning techniques to reveal open, closed and filtered ports.

You will see how `nmap` works in the penetration testing part of the course.

```
23 def initialize(experiment, observations)
24   @experiment = experiment
25   @observations = observations
26   @control = control
27   @candidates = observations - @control
28   evaluate_candidates
29
30   freeze
31 end
32
33 # Returns the experiment's context
34 def context
35   experiment.context
36 end
37
38 # Returns the result a match between an
39 def matcher
40   @experiment/resultub 1.1
41 end
```



## 4.2.3.3 Detecting Services

Knowing that a port is open is just half of the job.

Next, you will need to know what is the service listening on that port!



## 4.2.3.3 Detecting Services

In fact, knowing just the port is not enough because, as you know from the *Networking* module, a system administrator can configure a service to listen to any TCP or UDP port.

To detect which service is listening on a port, you can use `nmap` or other fingerprinting tools.

## 4.2.3.3 Detecting Services

By knowing the services running on a machine, a penetration tester can infer:

- The **operating system**.
- The **purpose** of a particular IP address; for example, if it is a server or a client.
- The **importance** of the host in the client's business. For example, an e-commerce enterprise will heavily rely upon its website and its database servers.

## 4.2.3.3 Detecting Services

After a map of the network infrastructure and the services running on it is built, you can start the vulnerability assessment using a vulnerability scan and/or manual inspection.



## 4.2.4 Vulnerability Assessment

The **vulnerability assessment** phase is aimed at building a **list of the vulnerabilities present** on the target systems.

The penetration tester has to carry out a vulnerability assessment on **each target** found in the previous steps.

```
23 # ... the experiment results ...
24 # ... the experiment results ...
25 # ... the experiment results ...
26 # ... the experiment results ...
27 # ... the experiment results ...
28 # ... the experiment results ...
29 # ... the experiment results ...
30 # ... the experiment results ...
31 # ... the experiment results ...
32 # ... the experiment results ...
33 # ... the experiment results ...
34 # ... the experiment results ...
35 # ... the experiment results ...
36 # ... the experiment results ...
37 # ... the experiment results ...
38 # ... the experiment results ...
39 # ... the experiment results ...
40 # ... the experiment results ...
41 # ... the experiment results ...
42 # ... the experiment results ...
43 # ... the experiment results ...
44 # ... the experiment results ...
45 # ... the experiment results ...
46 # ... the experiment results ...
47 # ... the experiment results ...
48 # ... the experiment results ...
49 # ... the experiment results ...
50 # ... the experiment results ...
51 # ... the experiment results ...
52 # ... the experiment results ...
53 # ... the experiment results ...
54 # ... the experiment results ...
55 # ... the experiment results ...
56 # ... the experiment results ...
57 # ... the experiment results ...
58 # ... the experiment results ...
59 # ... the experiment results ...
60 # ... the experiment results ...
61 # ... the experiment results ...
62 # ... the experiment results ...
63 # ... the experiment results ...
64 # ... the experiment results ...
65 # ... the experiment results ...
66 # ... the experiment results ...
67 # ... the experiment results ...
68 # ... the experiment results ...
69 # ... the experiment results ...
70 # ... the experiment results ...
71 # ... the experiment results ...
72 # ... the experiment results ...
73 # ... the experiment results ...
74 # ... the experiment results ...
75 # ... the experiment results ...
76 # ... the experiment results ...
77 # ... the experiment results ...
78 # ... the experiment results ...
79 # ... the experiment results ...
80 # ... the experiment results ...
81 # ... the experiment results ...
82 # ... the experiment results ...
83 # ... the experiment results ...
84 # ... the experiment results ...
85 # ... the experiment results ...
86 # ... the experiment results ...
87 # ... the experiment results ...
88 # ... the experiment results ...
89 # ... the experiment results ...
90 # ... the experiment results ...
91 # ... the experiment results ...
92 # ... the experiment results ...
93 # ... the experiment results ...
94 # ... the experiment results ...
95 # ... the experiment results ...
96 # ... the experiment results ...
97 # ... the experiment results ...
98 # ... the experiment results ...
99 # ... the experiment results ...
100 # ... the experiment results ...
```



## 4.2.4 Vulnerability Assessment

The next phase, exploitation, will go through this list to exploit the systems.

The bigger the list, the more the chances to exploit the systems in scope.

```
24 # @param result - the experiment result as a dict
25 # @param observations - an array of Observations, in experiment
26 # @param control - the control observation
27
28 def skillsize(experiment, observations = [], control = null)
29   @experiment = experiment
30   @observations = observations
31   @control = control
32   @candidates = observations - [control]
33   evaluate_candidates
34
35   freeze
36
37   def context
38     experiment.context
39   end
40
41   # TODO: the name of the experiment
42   def experiment_name
43     experiment.name
44   end
45
46   # TODO: what the result a match between an
47   def matched?
48     ...
49   end
50   @score/result.rb 1.1
```



## 4.2.4 Vulnerability Assessment

You can carry out a vulnerability assessment:

- **Manually** by using data collected in the previous phases
- By utilizing **automated tools**

Vulnerability assessment tools are scanners that send probes to the target systems to detect whether a host has some well-known vulnerabilities.



## 4.2.4 Vulnerability Assessment

Once the vulnerability scan is complete, the scanner will deliver a report that the pentester can use in the exploitation phase.

As automated scanners can perform a huge number of probes, it is **extremely** important to properly configure them leveraging the information collected in the previous steps.

```
24 # @param @experiment - the experiment to be evaluated
25 # @param @observations - an array of Observations, in which
26 # @param @control - the control observation
27
28 def skillRanking(experiment, observations = [], control = null)
29   @experiment = experiment
30   @observations = observations
31   @control = control
32   @candidates = observations - (@control)
33   evaluate_candidates
34
35   freeze
36   @context
37   experiment.context
38
39   @experiment_name
40   experiment.name
41
42   @context
43   context
44
45   @result
46   result
47
48   @result
49   result
50
51   @result
52   result
53
54   @result
55   result
56
57   @result
58   result
59
60   @result
61   result
62
63   @result
64   result
65
66   @result
67   result
68
69   @result
70   result
71
72   @result
73   result
74
75   @result
76   result
77
78   @result
79   result
80
81   @result
82   result
83
84   @result
85   result
86
87   @result
88   result
89
90   @result
91   result
92
93   @result
94   result
95
96   @result
97   result
98
99   @result
100  result
```

## 4.2.4 Vulnerability Assessment

Otherwise, the scanner will blindly perform all its probes, even the ones that do not apply to your targets; this would increase the chances of crashing services and would also take more time than necessary to complete.



## 4.2.4 Vulnerability Assessment

Most of the time this phase is done by using both automated scanners and manual inspection.

**Automated tools can help carry out a penetration test, but they will not perform a penetration test on their own.**

```
def initialize(experiment, observations = [], control = nil)
  @experiment = experiment
  @observations = observations
  @control = control
  @candidates = observations + [control]
  evaluate_candidates

  freeze

  @context =
    experiment.context

  def experiment_name
    experiment.name
  end

  def match?
    # Check whether the result is a match between an
  end

  def result
    result
  end
end
```

## 4.2.5 Exploitation

At this point, it's time to verify if the vulnerabilities really exist. The **exploitation** phase takes care of exploiting all the vulnerabilities found during the previous step.

During the exploitation phase a penetration tester **checks and validates a vulnerability** and also **widens and increases** the pentester's privileges on the target systems and networks.

## 4.2.5 Exploitation

A successful exploit of a machine helps to investigate the target network further, to discover new targets and to **repeat the process** from the information gathering phase!

A penetration test is indeed a **cyclic process**.



## 4.2.5 Exploitation

The process ends when there are no more systems and services **in-scope** to exploit.

Remember, a penetration test is used to find **any and all vulnerabilities**.



## 4.2.6 Reporting

Lastly, the final **penetration test report** is as important as the whole testing phase, as it is your way to officially deliver and communicate the results of your tests with:

- Executives
- IT Staff
- Development team

The report shows and explains the result of your tests and is the actual deliverable of your professional engagement.

```
21 # Overall context of the experiment
22
23 # Experiment - the experiment will result in 2
24 # observations - an array of Observations, in this case
25 # control - the control observation
26
27
28 def initialize(experiment, observations = [], control = nil)
29   @experiment = experiment
30   @observations = observations
31   @control = control
32   @candidates = observations - [control]
33   evaluate_candidates
34
35   freeze
36 end
37
38 # Return the experiment's context
39 def context
40   experiment.context
41 end
42
43 # Return the name of the experiment
44 def experiment_name
45   experiment.name
46 end
47
48 def display
49   puts "Experiment: #{@experiment.name}"
50   puts "Observations: #{@observations.map(&:name).join(", ")}"
```



## 4.2.6.1 The Report

The report must address:

- Techniques used
- Vulnerabilities found
- Exploits used
- Impact and risk analysis for each vulnerability
- Remediation tips

Targeted tips on how to effectively remediate each vulnerability are the **real value** for the client.

## 4.2.6.1 The Report

Remember that the work of a penetration tester is much more appreciated if, other than his elite exploitations skills, it provides **useful suggestions and techniques** the client can use to resolve their security issues.



## 4.2.6.2 Consultancy

Penetration testers are often asked to provide some hours of consultancy after delivering the report; this is an additional service to the client should they need further clarification or help regarding your findings.



## 4.2.6.2 Consultancy

After the consultancy step, the engagement is closed and the penetration tester must keep the report **encrypted in a safe place**, or better yet, **destroy** it.

## 4.2.7 The Secret of an Effective Pentest

Q

*Why wouldn't an experienced penetration tester just skip to the exploitation phase? In the end, it's what they are paid for, isn't it?*

Imagine the systems in scope as a **target**. The bigger the target, the more chances you have to hit it with your darts.

Stages like information gathering and fingerprinting do just that; they **make your target wider!**



## 4.2.7 The Secret of an Effective Pentest

In technical jargon, this activity is called "**widening the attack surface**".

Using your time at widening the attack surface is much more valuable than shooting darts at an unknown target. You do not know where to shoot, and you do not know which technique is the best to use.

## 4.2.7 The Secret of an Effective Pentest

On the other hand, a targeted attack has many more chances to succeed! Your main goal as a pentester is to first increase your chances of success and then shoot your darts.

**Sticking to the process you've just seen is the real secret for an effective pentest!**

```
def initialize(experiment, observations = {}, candidates = {}):
    @experiment = experiment
    @observations = observations
    @control = control
    @candidates = observations - {control}
    evaluate_candidates

    freeze

    end

# Returns the experiment's context
def context
    @experiment.context

end

def experiment_name
    @experiment.name

end

# Returns whether the result is a match between an observation and the control
def match?
    @observations[result] == @control
end
```





## 4.2.7 The Secret of an Effective Pentest

In fact, highly motivated and experienced hackers spend most of their time investigating their victims and gathering information about them using as many sources as possible; this helps them launch highly targeted attacks that do not trigger alarms in the victim's defense system.



## 4.2.7 The Secret of an Effective Pentest

A successful and stealthy attack is made possible by a deep understanding of the target, which comes from a thorough information gathering phase.



## 4.2.7 The Secret of an Effective Pentest

During the penetration testing part of this course, you will see the tools and techniques to carry out each and every step of the penetration testing process we have studied in this module: information gathering, scanning, vulnerability assessment, and exploitation!



# References





## Nmap scanner

<http://nmap.org/>

# References

