



GHOST
PROTOCOL

RP

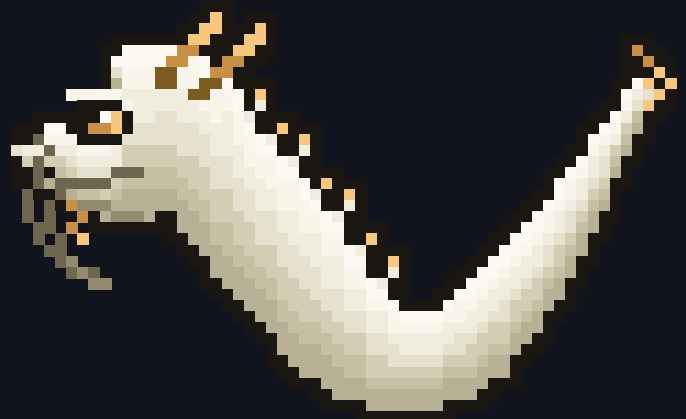
THE FIELD LIBRARY

OPERATIONS

OBJECTIVES, NOT COVERAGE

Red and Purple Team

Red team versus pentest, the attack lifecycle, tradecraft and the line, purple teaming, running and reporting.



Ghost Protocol (Pvt) Ltd · Colombo · ghosts.lk

Security internship training · The Field Library · Edition 1 · 2026

THE GHOST PROTOCOL FIELD LIBRARY · OPERATIONS

Red and Purple Team

Red team versus pentest, the attack lifecycle, tradecraft and the line, purple teaming, running and reporting.

Ghost Protocol (Pvt) Ltd · Colombo

The Field Library, Edition 1 · 2026

Copyright 2026 Ghost Protocol (Private) Limited. All rights reserved.

Sold as a PDF for the buyer's own reading. Please do not share or resell the file.

The library

The Field Library is a whole shelf, not one book. The Core Path is the spine everyone reads. The rest is depth and breadth, taken on as the work calls for it. The shelf is complete now, and you are reading the one marked here.

A · THE CORE PATH

O The Oath BUILT

I Groundwork BUILT

II The Web BUILT

III The Wider Field BUILT

IV The Work BUILT

V The Craft BUILT

B · DEEP TRACKS

AI and Agents BUILT

Reading Code BUILT

Internal and Active Directory BUILT

The Web, Deeper BUILT

Cloud and Containers BUILT

Mobile and APIs BUILT

Reverse Engineering BUILT

Hardware and the Edges BUILT

C · THE OPERATIONS

Red and Purple Team YOU ARE HERE

OSINT and Recon BUILT

Report Writing Mastery BUILT

The Consulting Craft BUILT

Bug Bounty BUILT

D · THE OTHER SIDE

Detection and Response BUILT

Threat Hunting and Forensics BUILT

Hardening and Architecture BUILT

E · THE GROUND RULES

The Law, Deeper BUILT

Standards and Compliance BUILT

Ethics and Disclosure BUILT

F · COMPANIONS

The Reference Pack BUILT

The Lab Workbook BUILT

Case Files BUILT

The Bestiary BUILT

The Glossary and the Ledger BUILT

Contents

- Opening 5
- 1** Red Team versus Pentest 6
- 2** The Attack Lifecycle 9
- 3** Tradecraft and the Line 14
- 4** Purple Teaming 18
- 5** Running and Reporting the Engagement 22
- Worked answers 27

Objectives, not noise



The word on the scope was red team, and Blink lit up at it, already imagining going loud.

BLINK "Red team. So we get to really let go this time."

SOL "Not quite. Red team is not permission to be reckless. It is a change of goal. A pentest asks how many doors are unlocked. A red team picks one objective the client cares about and asks whether a real adversary could reach it without being caught. Quieter, not louder. More disciplined, not less. And the line you have carried since Volume 0 does not move an inch because the engagement has a scarier name."

Wyrms's eye stayed open on the authorisation, the one that says exactly how far quiet is allowed to go.

WYRM "the line holds."

So the book is the craft of the objective-led engagement, and the discipline that keeps it clean: evade the defender's sensors because measuring that is the test, never to hide from the client, and turn the whole thing, in the end, toward making their blue team better.

CHAPTER ONE

Red Team versus Pentest

A penetration test and a red team engagement are not the same job sold at two prices. They answer different questions. Learn which question a client is really asking before you quote either one.

A penetration test asks "what is broken here?" A red team engagement asks "if a capable adversary came for us, would we see them, and could we stop them?" Same trade, different question, and the second one is only worth asking once the first has been answered well.

You already know the pentest from the rest of this Library: a scoped system, a window, and a tester who finds as many real weaknesses as the time allows, each one proven and written up. A **red team engagement** keeps the authorisation and the professionalism and changes the goal. Instead of covering a system and listing its flaws, it picks an objective a real attacker would want, reaches for it the way a real attacker would, and measures one thing above all: how far the defence let it get before anyone noticed. The flaws along the way matter less than the path and the silence.

Coverage versus objectives

A pentest is measured by coverage. Did you test every endpoint, every parameter, every role? The deliverable is a set of findings sorted by severity, and a good one is close to complete for its scope. A red team engagement is measured by an objective: reach the payment system, obtain domain administrator, pull a copy of the crown-jewel dataset. You do not enumerate every weakness. You find the one chain that gets you to the goal, and you report whether you got there, how, and where the blue team saw you. A red team that found forty medium issues and never attempted the objective has done a pentest and mislabelled it.

The practical difference shows in what you ignore. On a pentest, an exposed admin panel on an unrelated host is a finding you write up. On a red team, if it is not on the path to the objective, you note it and move on, because your client is not paying you to inventory the estate this time. They are paying to learn whether their detection and response work against someone with a goal.

Assumed breach

Perimeters fail. Given enough time, a determined attacker gets a foot inside through a phished user, an unpatched edge service, or a supplier. Spending three weeks of a red team budget re-proving that is usually a poor trade, so many engagements start from an **assumed breach**: the client grants you a starting position, a low-privilege account or a planted agent on one workstation, and the clock starts there. That lets the engagement spend its time on the expensive question, which is what happens after the first machine falls, not on the well-understood fact that a perimeter can be crossed.

Be clear about what assumed breach is. It is a position the client hands you in writing, inside the scope, so you can emulate the part of an intrusion that actually tests their defence. It is not a licence to break in for real to create that position, and it is not an excuse to skip the paperwork because "we are pretending to be attackers". The pretending stops exactly at the authorisation.

Threat-led, not generic

The strongest red team engagements are **threat-led**. Rather than a generic "attack us", they emulate a specific adversary the client plausibly faces: a commodity ransomware crew, a disgruntled insider, an operator after a particular dataset. Threat intelligence shapes the scenario, so the techniques you use are the ones that threat actually uses, and the test measures defence against a real threat instead of a checklist. In regulated finance abroad, this is formalised. Frameworks such as TIBER-EU in the European Union and CBEST in the United Kingdom run intelligence-led red team tests against banks under regulator oversight. Those are foreign regulatory schemes, not Sri Lankan law, and this book flags them as a model to understand rather than a rule that binds us here. The shape is the lesson: a test is more useful when it emulates a named threat than when it attacks in the abstract.

When a client needs one

A red team engagement is the wrong product for most first-time clients, and saying so is part of doing this honestly. If an organisation has no logging, no one watching alerts, and a last pentest full of criticals still unfixed, a red team will succeed on night one and teach them nothing they did not already owe themselves: patch, log, and watch. They need a pentest and basic hygiene first. A red team earns its cost when there is something to test: a blue team or a security operations centre, endpoint detection, central logging, an incident runbook. Then the question "would we catch a real one?" has a real answer, and the money buys that answer.

■ THE LINE

A red team engagement is wider, quieter, and longer than a pentest, which makes the written authorisation more important, not less. Before any action, the same three questions from Volume 0 hold, and two more besides:

- Is the objective, the scenario, and any assumed-breach starting position named in a signed authorisation you have read?
- Is every system you are about to touch inside the scope and the window, including the quiet lateral steps a long engagement tempts you into?

Objectives-based does not mean open-ended. The goal is written down; the boundary is written down; you stay inside both. Assumed breach is a position you were granted, never one you take.



SOL SAYS

The red team is the impressive-sounding product, and that is exactly why you must be the one in the room who will talk a client out of it when it is wrong for them. Tell them what their money actually buys. A client who is not ready for a red team, and is sold one anyway, pays to watch you win in an hour and learns nothing. A client you steer to the pentest they needed remembers who was straight with them when the red team is finally the right call.

■ ON YOUR FEET

A prospect asks for a "full red team" because a competitor mentioned having one. On the call it emerges they have no endpoint detection, no central logging, and the criticals from their last pentest a year ago are still open. The budget would stretch to one or the other. What do you recommend, and how do you explain it without talking yourself out of the work or overselling?

■ EXERCISE

Build the lab you will use for the rest of this volume, and learn to tell the two jobs apart.

1. On a host you own, stand up a small isolated lab on a host-only network: one Windows VM with Sysmon installed, using a public configuration such as SwiftOnSecurity's sysmon-config or Olaf Hartong's sysmon-modular, and its logs shipped to a collector you control, and one Kali VM as the attacker. For the collector, a free Wazuh server (add the Sysmon event channel to its Windows agent), or Elasticsearch and Kibana fed by Winlogbeat, is enough; one small Linux VM on the same host-only network will hold it. Nothing touches a network that is not yours.
2. Write a one-paragraph **scenario card** for a mock red team engagement against that lab: the threat actor you are emulating, the single objective, the assumed-breach starting position (if any), the window, and two things explicitly out of scope.
3. For the same lab, write the first lines of a **pentest scope** instead: the systems to be covered and the kinds of flaws to enumerate. Note in one sentence how the two documents differ in what they ask you to produce.

■ VERIFY IT

Before you trust this lab to tell you anything later, prove the instrument works. On the Windows VM, run a harmless command such as `whoami`. Then open your collector and find the matching Sysmon process-creation event (Event ID 1), with that command line in it. If you cannot find the event, your logging is not wired up, and every "we detected it" later in this volume would be a guess. Fix the pipeline now, while the only thing at stake is a `whoami`.



WHAT WYRM KEEPS

A pentest asks what is broken. A red team asks whether you would see a real attacker and stop them. Sell the one the client needs, not the one that sounds impressive.

- Pentest is measured by coverage and a list of findings; a red team is measured against an objective and by how far the defence let it get before noticing.
- Assumed breach is a starting position granted in writing so the test can spend its time on detection and response, not on re-proving that perimeters fall. It is granted, never taken.
- Threat-led beats generic: emulate a named adversary the client actually faces. TIBER-EU and CBEST are foreign regulatory models, flagged, not Sri Lankan law.
- A red team is wasted on an org with no detection to test. They need a pentest and basic hygiene first, and telling them so is clean conduct, not lost work.

CHAPTER TWO

The Attack Lifecycle

Real intrusions move through recognisable stages, and each stage leaves traces. Learn the stages as a defender's map and a shared language, not as a recipe to run against production.

Attackers do not arrive with a single magic exploit. They move through stages: get in, run something, keep their foothold, climb, look around, spread, and act on what they came for. Each stage is a chance for a defender to see them, and naming the stages gives you and the blue team one vocabulary to talk about both the attack and the gaps in catching it.

Two maps

The first map is the **Cyber Kill Chain**, Lockheed Martin's seven linear phases: reconnaissance, weaponisation, delivery, exploitation, installation, command and control, and actions on objectives. It is a good first mental model because it reads left to right and ends where the attacker wanted to be. Its weakness is that real intrusions are not a tidy line.

The second map is **MITRE ATT&CK**, and it is the one the industry now shares. ATT&CK is a matrix of **tactics** (the attacker's goal at a moment, the "why") and **techniques** (the "how"), built from observed real-world intrusions. The Enterprise matrix has fifteen tactics: Reconnaissance, Resource Development, Initial Access, Execution, Persistence, Privilege Escalation, Stealth, Defense Impairment, Credential Access, Discovery, Lateral Movement, Collection, Command and Control, Exfiltration, and Impact. The matrix moves between editions, so tag the version you cite. This book cites ATT&CK v20 (October 2026); its predecessor, v19 (April 2026), split the former single Defense Evasion tactic into Stealth, which kept the id `TA0005`, and the new Defense Impairment (`TA0112`), which is why older material still lists fourteen. A technique sits under a tactic and often has sub-techniques: credential dumping is `T1003` (OS Credential Dumping), and reading it out of memory is the sub-technique `T1003.001` (LSASS Memory). When you tell a blue team "we tested T1003.001 and nothing fired", you have said something precise that they can act on. That precision is the whole point of the shared language.

Get in	Run & stay	Climb & take	Evade	Move & find	Act on goal
Reconnaissance Resource Development Initial Access	Execution Persistence	Privilege Escalation Credential Access	Stealth Defense Impairment	Discovery Lateral Movement	Collection Command and Control Exfiltration Impact

15 tactics (ATT&CK v20). Read as goals the attacker pursues, not steps in order.

The fifteen ATT&CK v20 tactics, grouped by what the attacker is doing. The map is for spotting your own blind spots, not a running order.

■ NOTE

ATT&CK tactics are not a sequence you must follow in order. A real intrusion skips stages, loops back, and runs several at once: an attacker may land with Valid Accounts and need no Execution to begin with, or re-enter Discovery every time they reach a new host. Read the matrix as a set of goals an adversary pursues, not as steps one through fifteen.

Walking the stages, from the defender's chair

For each stage below, the question that matters to us is not "how do I do it" but "what would this look like in the logs". That is the view that makes you useful on a red team and on a blue team both.

Reconnaissance and getting in

Reconnaissance ([T1595](#) Active Scanning, [T1589](#) Gather Victim Identity Information) is the attacker learning about the target. Much of it happens off your network and you cannot stop it, so the defender's realistic posture is to reduce the attack surface that recon can find and to watch the external edge rather than expect to block the looking. **Initial Access** ([T1566.001](#) Spearphishing Attachment, [T1190](#) Exploit Public-Facing Application) is the first foothold. Here the defender has real controls: mail filtering and user reporting for phishing, fast patching of the edge, and web and proxy logs that record the first unusual request.

Execution and staying

Execution ([T1059.001](#) PowerShell, [T1047](#) Windows Management Instrumentation) is the attacker running their code on the host. The defender's gold here is command-line visibility: Security Event ID 4688 with the full command line enabled, Sysmon Event ID 1, and PowerShell script-block logging, which writes Event ID 4104 to the Microsoft-Windows-PowerShell/Operational log. **Persistence** ([T1547.001](#) Registry Run Keys, [T1053.005](#) Scheduled Task) is how they survive a reboot. Scheduled task creation raises Security Event ID 4698; a new value under a Run key raises Sysmon Event ID 13; a baseline of autostart entries turns "a new one appeared" into an alert.

Climbing and taking credentials

Privilege Escalation ([T1078](#) Valid Accounts, and the broad [T1548](#) Abuse Elevation Control Mechanism) is the attacker getting more rights. **Credential Access** ([T1003.001](#) LSASS Memory) is where they harvest the secrets that make the rest easy. Reading credentials out of LSASS shows up when a process opens a handle to [lsass.exe](#) with suspicious access rights, visible as Sysmon Event ID 10, and it is what protections such as LSA Protection (RunAsPPL) and Credential Guard exist to blunt. The Kerberos-specific credential attacks and the Active Directory depth behind them belong to the internal and Active Directory course; reference it there rather than repeating it here.

Stealth and defence impairment, as stages to detect

Stealth and **Defense Impairment** are the two tactics an adversary uses to get past the defence, and in ATT&CK v19 they replaced the former single Defense Evasion. Stealth is blending in and concealing actions so the activity looks legitimate; Defense Impairment is actively degrading the controls themselves, clearing logs, disabling tooling, modifying a firewall rule. We name them so you recognise them, and we deliberately do not teach their techniques, because a method whose only purpose is to defeat