

RED TEAM - Development & Operations Book

This document has been created by Jose Pablo Urena Gutierrez to resolve, summarize, and study the different parts of the book:
Red Team Development and Operations.

Objective definition by a Red Team:

- What ability does an adversary have to access common areas?
- what ability does an adversary have to access restricted area?
- can an adversary use gained access to enable electronic capabilities?
- what impacts can an adversary have on key/critical system?
- what ability does an adversary have to move through a network freely?
- how long can an adversary live on target without discovery?
- what actions are required to trigger a detection response?

Homeworks

Homework #1

Red Team definition:

Red teams emulate a threat in order to test the physical, software, and specially human detection methods in a company, in order to identify the company's response malicious threats.

- How does RT decide what first attack to use in order to get a successful foothold?

Depends on the goal that the team wants to achieve, either inside or outside of the network. Remember that based on the decision made on the contract and what needs to be tested, will define the starting point for the RT:

- Insider
- Outsider
- Physical Intrusion
- Supply chain
- True insider

Homework #2

This homework asks to build:

- a RT methodology guide
- roles and responsibilities template
- threat profile template
- ROE template
- deconfliction template
- data handling guide

RT Methodology Guide

The RT should follow this path (branch process can and will occur but the team must maintain the path) in order to accomplish each engagement accordingly and successfully. The entire team must follow it's duties according to the plan and never mix engagement paths between each other.

Guide:

1- All parties come together (RT, Trusted agent, ECG, White Cell, determining the scalability, scope, objective, and other limits the team will have during its engagement).

⇒

2- All the agreements are detailed in the ROE, read carefully and signed by each member of the party.

⇒

3- The RT will start its actions based on TTPs as agreed, following the Threat Actor Profile as close as possible. The RT will divide its actions on three:

Recon → Stay In → Act

⇒

4- Any time the Blueteam, or other members of security and staff catch a member of the RT in its labors, they must report it immediately. Based on what was agreed by the parties in the ROE, the RT will decide how to continue the engagement and what parts of it. The ROE must specify what actions should a Red teamer take once caught

and who to communicate it's part of it's duties, not to any member of the company.

⇒

5- The Blueteam will deconflict the information between Red Team tests and real attackers logs, if the RT can't communicate it's actions until the end of the engagement or until the White Cell decides, then Blueteam should treat any finding as outside attackers and follow Incident procedure.

⇒

6- Impact phase, once in and established, the RT deploys the last actions to cause impact and determine if the threat can be successful or not by fulfilling the objective. Based on the last actions the team documents accordingly and continuously communicates with white cell and ECG if needed.

⇒

7- Collection, all the information and findings are gathered and provided to the correspondant parties and teams decided previously on the ROE, the reports will not be given to members outside the already agreed scope.

Roles & Responsibilities

Depending on the amount of team members, the responsibilities will be distributed among them.

The responsibilities of the team are as following:

- Team leader (coordinator between the team and management, decides on other team member responsibilities and attack flow of the engagement during action)
- Architect (in charge of building all the infrastructure for the team to use, C2s, software distribution, patching, physical devices)
- Social engineer (in charge of social engineering tasks, passing on human security)
- Tech Lead, but every member should be able to deploy attacks with the toolkit and manage inside the network accordingly
- Reporting (caught actions, impact, decisions made along the way with parties)
- Malware development
- Testing of malware in dev lab environment
- Phishing campaign creation and tracking

Threat Profile Template

This is a threat profile template that can be used to analyze a profile and then ask the team if they can fulfill what it takes to impersonate this threat, if the resources are sufficient:

Category	Description
Description	Why does mo
Goal and intend	What is the o
Key IOCs	Left behind in
C2 infrastructure	What C2 infr
TTPs	What TTPs w
Exploitation	Malware, cus
Persistence	How the thre

Once the team studies the threat profile of a threat actor, must ask this questions to itself:

- Can our RT do these actions?
- Can we emulate or create the software used in this profile?
- Can we last the same amount of time this threat profile requires?
- Do we have the staff capabilities to accomplish this with the same amount of members?

Rules of Engagement ROE Template

Rules, responsibility, relationship, and guidelines between RT, Trusted Agent, White Cell, Engagement Control Group, and remaining parties.

Contents:

Engagement Target

Company details	Description
Organization Name	
Address	
Specific groups or divisions	
Organizational Identifiers	
Senior Management Contact Info	

ECG Contact List

Member	Information
ECG personnel	
White Cell	
Trusted Agent	
Red Team Lead	
Red Tech Lead	

Engagement Objectives

Objective #1	Description
Conditions	
Threat Level	
Targeted objectives	
Targets of opportunity	
Measures of success/failure	

Authorized Target Space

Area	Zone
Network	
	IP Boundaries
	Domains and workgroups
	Areas off-limits and resources
	Off-limits machines, network equipment or apps.
	Maintenance Windows
Physical	
	Areas of the campus
	Buildings
	Offices
	Off-limits areas

Actions and Approval

Procedure	Activity
Authorized Action	
	#1 Approved activity for the engagement
Non Authorized Action	
	#1 Non authorized activity for the engagement
Approval Process	
	Approval process
	Contact list

Deconfliction Template

Basic deconfliction template:

Security Incident	IncidentNumber0001
Requested by:	Status:
Asset affected:	Source that identified it:
Location:	Threat Type (Internal/External)
Risk Score:	Dates (Occured, Detected, Responded, Contained):
Short Description:	
Activities performed:	
Full Description:	
Impact:	Priority:
Red Team Activity:	(Confirmed/Suspicious/Not Identified)
	Red Team Member Identified

Data Handling Template

On an engagement, Red Team will encounter many different types of data or actions to apply on that data, a correct matrix of data handling is necessary for the team to know

how to approach its usage:

Data/Process	Action
PII	Avoid - If enc
Exploit data	Allowed - RT
Data Modification	Denied - RT s
DoS	Avoid - Unles
Red Team Equipment	Guarded - Sa
Red Team OS	Encrypted
Password Policy	Obligatory or
Communications	All encrypted
Data collected	All encrypted
Software and Hardware used during engagement	All removed c

Red Team must sign a Non Disclosure aggreement and follow ROE instructions on data handling.

Data Repository and Storage Guidelines:

Data collected during an engagement must be stored following a standard and easy to access(for the team) including all details. Lets see the guideline.

- Per engagement we require a mount to store all the data (data repo), contents of the mount drive (file hierarchy):

→ Engagement_name

⇒ admin

⇒ osint

⇒ recon

⇒ targets

- domain_name

◇ exfil

- ip_hostname

◇ exfil

⇒ screenshots

- YYYYMMDD_HHMM_IP_Description.png

⇒ payloads

⇒ logs

⇒ Operators_daily

- operator_1

- operator_2

⇒ Readme.md

Activity Logs

Operator Logs

Source

Terminal Logs
Commercial Tools Logs
Custom Tools Logs
Source

Homework #3

- Data repository and storage has been expanded in the data handling template on Homework #2

Data Handling Workflow

Data Handling Workflow:

- 0- Daily, every red team operator must push his/her changes to the Mount where engagement data is collected.
- 1- RT lead must keep a table of objectives per day (may vary), changes made must be pushed
- 2- Every RT operator will download the logs from his terminal and rename every file (screenshots, bash history, etc).
- 3- Every operator will fill the daily operator template (extracted from Data Handling Template on Homework #2) according to his actions, findings and collection:

Operator Logs	
	Start & End Timestamp
	Source IP of attacker
	Source hostname
	Destination IP
	Destination hostname
	Destination Ports
	Destination System name
	Pivot IP
	Pivot Hostname
	Pivot Ports
	URL
	Tool/Application
	Action
	Commands & Output
	Description of activity
	Results
	System modifications
	Screenshots
	Operator Name

Remember, this is done daily.

- 4- Every RT operator will push the template and the data collected into the Mount of the team once all this info is ready and documented. Every file must go into its specific folder according to the File Hierarchy defined (Data Handling Template from Homework #2):

- Engagement_name
 - ⇒ admin
 - ⇒ osint
 - ⇒ recon
 - ⇒ targets
 - domain_name
 - ◇ exfil
 - ip_hostname
 - ◇ exfil
 - ⇒ screenshots
 - YYYYMMDD_HHMM_IP_Description.png
 - ⇒ payloads
 - ⇒ logs
 - ⇒ Operators_daily
 - operator_1
 - operator_2
 - ⇒ Readme.md

Templates will go inside the “Operators_daily” folder corresponding to the operator name.

5- The Red Team lead will unmount the share once all the members have pushed their changes.

6- Red Team lead will perform a backup of the mount.

Tradecraft Guidelines

Circumstances, changes, and new tools require a detailed analysis before bringing

them into the engagement.
These are the guidelines to consider those activities into the engagement, and have a better engagement:

Factor
Tool/Tecnology Name
Environment/Awareness
C2
Exploits

Tool Kit

This is a standard toolkit that you can use to setup your machines for an engagement:

<https://github.com/SlyJose/Red-Team-Knowledge.git>

List of tools:

Reconnaissance

Active Intelligence Gathering

- EyeWitness is designed to take screenshots of websites, provide some server header info, and identify default credentials if possible. <https://github.com/ChrisTruncer/EyeWitness>
- AWSBucketDump is a tool to quickly enumerate AWS S3 buckets to look for loot. <https://github.com/jordanpotti/AWSBucketDump>
- AQUATONE is a set of tools for performing reconnaissance on domain names. <https://github.com/michenriksen/aquatone>
- spoofcheck a program that checks if a domain can be spoofed from. The program checks SPF and DMARC records for weak configurations that allow spoofing. <https://github.com/BishopFox/spoofcheck>
- Nmap is used to discover hosts and services on a computer network, thus building a "map" of the network. <https://github.com/nmap/nmap>
- dnsrecon a tool DNS Enumeration Script. <https://github.com/darkoperator/dnsrecon>

Passive Intelligence Gathering

- Social Mapper OSINT Social Media Mapping Tool, takes a list of names & images (or LinkedIn company name) and performs automated target searching on a huge scale across multiple social media sites. Not restricted by APIs as it instruments a browser using Selenium. Outputs reports to aid in correlating targets across sites. https://github.com/SpiderLabs/social_mapper
- skiptracer OSINT scraping framework, utilizes some basic python webscraping (BeautifulSoup) of PII paywall sites to compile passive information on a target on a ramen noodle budget. <https://github.com/xillwillx/skiptracer>

- ScrapedIn a tool to scrape LinkedIn without API restrictions for data reconnaissance. <https://github.com/dchrastil/ScrapedIn>
- linkScrape A LinkedIn user/company enumeration tool. <https://github.com/NickSanzotta/linkScrape>
- FOCA (Fingerprinting Organizations with Collected Archives) is a tool used mainly to find metadata and hidden information in the documents its scans. <https://github.com/ElevenPaths/FOCA>
- theHarvester is a tool for gathering subdomain names, e-mail addresses, virtual hosts, open ports/ banners, and employee names from different public sources. <https://github.com/laramies/theHarvester>
- Metagoofil is a tool for extracting metadata of public documents (pdf,doc,xls,ppt,etc) availables in the target websites. <https://github.com/laramies/metagoofil>
- SimplyEmail Email recon made fast and easy, with a framework to build on. <https://github.com/killswitch-GUI/SimplyEmail>
- truffleHog searches through git repositories for secrets, digging deep into commit history and branches. <https://github.com/dxa4481/truffleHog>
- Just-Metadata is a tool that gathers and analyzes metadata about IP addresses. It attempts to find relationships between systems within a large dataset. <https://github.com/ChrisTruncer/Just-Metadata>
- typofinder a finder of domain typos showing country of IP address. <https://github.com/nccgroup/typofinder>
- pwnedOrNot is a python script which checks if the email account has been compromised in a data breach, if the email account is compromised it proceeds to find passwords for the compromised account. <https://github.com/thewhiteh4t/pwnedOrNot>
- GitHarvester This tool is used for harvesting information from GitHub like google dork. <https://github.com/metac0rtex/GitHarvester>
- pwndb is a python command-line tool for searching leaked credentials using the Onion service with the same name. <https://github.com/davidtavarez/pwndb/>

Frameworks

- Maltego is a unique platform developed to deliver a clear threat picture to the environment that an organization owns and operates. <https://www.paterva.com/web7/downloads.php>
- SpiderFoot the open source footprinting and intelligence-gathering tool. <https://>

github.com/smicallef/spiderfoot

- datasploit is an OSINT Framework to perform various recon techniques on Companies, People, Phone Number, Bitcoin Addresses, etc., aggregate all the raw data, and give data in multiple formats. <https://github.com/DataSploit/datasploit>
- Recon-ng is a full-featured Web Reconnaissance framework written in Python. <https://bitbucket.org/LaNMaSteR53/recon-ng>

Weaponization

- Composite Moniker Proof of Concept exploit for CVE-2017-8570. <https://github.com/rwx/CVE-2017-8570>
- Exploit toolkit CVE-2017-8759 is a handy python script which provides pentesters and security researchers a quick and effective way to test Microsoft .NET Framework RCE. <https://github.com/bhdresh/CVE-2017-8759>
- CVE-2017-11882 Exploit accepts over 17k bytes long command/code in maximum. <https://github.com/unamer/CVE-2017-11882>
- Adobe Flash Exploit CVE-2018-4878. <https://github.com/anbai-inc/CVE-2018-4878>
- Exploit toolkit CVE-2017-0199 is a handy python script which provides pentesters and security researchers a quick and effective way to test Microsoft Office RCE. <https://github.com/bhdresh/CVE-2017-0199>
- demiguise is a HTA encryption tool for RedTeams. <https://github.com/nccgroup/demiguise>
- Office-DDE-Payloads collection of scripts and templates to generate Office documents embedded with the DDE, macro-less command execution technique. <https://github.com/0xdeadbeefJERKY/Office-DDE-Payloads>
- CACTUSTORCH Payload Generation for Adversary Simulations. <https://github.com/mdsecactivebreach/CACTUSTORCH>
- SharpShooter is a payload creation framework for the retrieval and execution of arbitrary CSharp source code. <https://github.com/mdsecactivebreach/SharpShooter>
- Don't kill my cat is a tool that generates obfuscated shellcode that is stored inside of polyglot images. The image is 100% valid and also 100% valid shellcode. <https://github.com/Mr-Un1k0d3r/DKMC>
- Malicious Macro Generator Utility Simple utility design to generate obfuscated macro that also include a AV / Sandboxes escape mechanism. <https://github.com/Mr-Un1k0d3r/MaliciousMacroGenerator>
- SCT Obfuscator Cobalt Strike SCT payload obfuscator. <https://github.com/Mr-Un1k0d3r/SCT-obfuscator>
- Invoke-Obfuscation PowerShell Obfuscator. <https://github.com/danielbohannon/Invoke-Obfuscation>
- Invoke-DOSfuscation cmd.exe Command Obfuscation Generator & Detection Test Harness. <https://github.com/danielbohannon/Invoke-DOSfuscation>
- morphHTA Morphing Cobalt Strike's evil.HTA. <https://github.com/vysec/morphHTA>
- Unicorn is a simple tool for using a PowerShell downgrade attack and inject shellcode straight into memory. <https://github.com/trustedsec/unicorn>
- Shellter is a dynamic shellcode injection tool, and the first truly dynamic PE infector ever created. <https://www.shellterproject.com/>
- EmbedInHTML Embed and hide any file in an HTML file. <https://github.com/Arno0x/EmbedInHTML>

- SigThief Stealing Signatures and Making One Invalid Signature at a Time. <https://github.com/secretsquirrel/SigThief>
- Veil is a tool designed to generate metasploit payloads that bypass common anti-virus solutions. <https://github.com/Veil-Framework/Veil>
- CheckPlease Sandbox evasion modules written in PowerShell, Python, Go, Ruby, C, C#, Perl, and Rust. <https://github.com/Arvanaghi/CheckPlease>
- Invoke-PSImage is a tool to embed a PowerShell script in the pixels of a PNG file and generates a oneliner to execute. <https://github.com/peewpw/Invoke-PSImage>
- LuckyStrike a PowerShell based utility for the creation of malicious Office macro documents. To be used for pentesting or educational purposes only. <https://github.com/curiousJack/luckystrike>
- ClickOnceGenerator Quick Malicious ClickOnceGenerator for Red Team. The default application a simple WebBrowser widget that point to a website of your choice. <https://github.com/Mr-Un1k0d3r/ClickOnceGenerator>
- macro_pack is a tool by @EmericNasi used to automatize obfuscation and generation of MS Office documents, VB scripts, and other formats for pentest, demo, and social engineering assessments. https://github.com/sevagas/macro_pack
- StarFighters a JavaScript and VBScript Based Empire Launcher. <https://github.com/Cn33liz/StarFighters>
- nps_payload this script will generate payloads for basic intrusion detection avoidance. It utilizes publicly demonstrated techniques from several different sources. https://github.com/trustedsec/nps_payload
- SocialEngineeringPayloads a collection of social engineering tricks and payloads being used for credential theft and spear phishing attacks. <https://github.com/bhdresh/SocialEngineeringPayloads>
- The Social-Engineer Toolkit is an open-source penetration testing framework designed for social engineering. <https://github.com/trustedsec/social-engineer-toolkit>
- Phishery is a Simple SSL Enabled HTTP server with the primary purpose of phishing credentials via Basic Authentication. <https://github.com/ryhanson/phishery>
- PowerShdll run PowerShell with rundll32. Bypass software restrictions. <https://github.com/p3nt4/PowerShdll>
- Ultimate AppLocker ByPass List The goal of this repository is to document the most common techniques to bypass AppLocker. <https://github.com/api0cradle/UltimateAppLockerByPassList>
- Ruler is a tool that allows you to interact with Exchange servers remotely, through either the MAPI/HTTP or RPC/HTTP protocol. <https://github.com/sensepost/ruler>
- Generate-Macro is a standalone PowerShell script that will generate a malicious Microsoft Office document with a specified payload and persistence method. <https://github.com/enigma0x3/Generate-Macro>
- Malicious Macro MSBuild Generator Generates Malicious Macro and Execute Powershell or Shellcode via MSBuild Application Whitelisting Bypass. <https://github.com/infosecninja/MaliciousMacroMSBuild>
- Meta Twin is designed as a file resource cloner. Metadata, including digital signature, is extracted from one file and injected into another. <https://github.com/threatexpress/metatwin>
- WePWNise generates architecture independent VBA code to be used in Office documents or templates and automates bypassing application control and exploit mitigation software. <https://github.com/mwrlabs/wePWNise>
- DotNetToJScript a tool to create a JScript file which loads a .NET v2 assembly from

memory. <https://github.com/tyranid/DotNetToJScript>

- PSAmSI is a tool for auditing and defeating AMSI signatures. <https://github.com/cobbr/PSAmSI>
- Reflective DLL injection is a library injection technique in which the concept of reflective programming is employed to perform the loading of a library from memory into a host process. <https://github.com/stephenfewer/ReflectiveDLLInjection>
- ps1encode use to generate and encode a powershell based metasploit payloads. <https://github.com/CroweCybersecurity/ps1encode>
- Worse PDF turn a normal PDF file into malicious. Use to steal Net-NTLM Hashes from windows machines. <https://github.com/3gstudent/Worse-PDF>
- SpookFlare has a different perspective to bypass security measures and it gives you the opportunity to bypass the endpoint countermeasures at the client-side detection and network-side detection. <https://github.com/hlldz/SpookFlare>
- GreatSCT is an open source project to generate application white list bypasses. This tool is intended for BOTH red and blue team. <https://github.com/GreatSCT/GreatSCT>
- nps running powershell without powershell. <https://github.com/Ben0xA/nps>
- Meterpreter_Paranoia_Mode.sh allows users to secure your staged/stageless connection for Meterpreter by having it check the certificate of the handler it is connecting to. https://github.com/r00t-3xp10it/Meterpreter_Paranoia_Mode-SSL
- The Backdoor Factory (BDF) is to patch executable binaries with user desired shellcode and continue normal execution of the prepatched state. <https://github.com/secretsquirrel/the-backdoor-factory>
- MacroShop a collection of scripts to aid in delivering payloads via Office Macros. <https://github.com/khr0x40sh/MacroShop>
- UnmanagedPowerShell Executes PowerShell from an unmanaged process. <https://github.com/leechristensen/UnmanagedPowerShell>
- evil-ssdp Spoof SSDP replies to phish for NTLM hashes on a network. Creates a fake UPNP device, tricking users into visiting a malicious phishing page. <https://gitlab.com/initstring/evil-ssdp>
- Ebowla Framework for Making Environmental Keyed Payloads. <https://github.com/Genetic-Malware/Ebowla>
- make-pdf-embedded a tool to create a PDF document with an embedded file. <https://github.com/DidierStevens/DidierStevensSuite/blob/master/make-pdf-embedded.py>
- avet (AntiVirusEvasionTool) is targeting windows machines with executable files using different evasion techniques. <https://github.com/govolution/avet>
- Phantom-Evasion Interactive antivirus evasion tool written in python capable to generate (almost) FUD executable even with the most common 32 bit msfvenom payload (lower detection ratio with 64 bit payloads). <https://github.com/oddcod3/Phantom-Evasion> WARNING: Please be aware this does install a Minero miner by default. This can be changed in the Setup folder and edit Config.txt and edit Mining = False. If you want to support the project and developers, I would recommend keeping on.
- Invoke-CradleCrafter PowerShell remote download cradle generator and obfuscator. <https://github.com/danielbohannon/Invoke-CradleCrafter>

Delivery

Phishing

- King Phisher is a tool for testing and promoting user awareness by simulating real world phishing attacks. <https://github.com/securestate/king-phisher>
- FiercePhish is a full-fledged phishing framework to manage all phishing engagements. It allows you to track separate phishing campaigns, schedule sending of emails, and much more. <https://github.com/Raikia/FiercePhish>
- ReelPhish is a Real-Time Two-Factor Phishing Tool. <https://github.com/fireeye/ReelPhish/>
- Gophish is an open-source phishing toolkit designed for businesses and penetration testers. It provides the ability to quickly and easily setup and execute phishing engagements and security awareness training. <https://github.com/gophish/gophish>
- CredSniper is a phishing framework written with the Python micro-framework Flask and Jinja2 templating which supports capturing 2FA tokens. <https://github.com/ustayready/CredSniper>
- PwnAuth a web application framework for launching and managing OAuth abuse campaigns. <https://github.com/fireeye/PwnAuth>
- Phishing Frenzy Ruby on Rails Phishing Framework. <https://github.com/pentestgeek/phishing-frenzy>
- Phishing Pretexts a library of pretexts to use on offensive phishing engagements. <https://github.com/L4bF0x/PhishingPretexts>
- Modlishka is a flexible and powerful reverse proxy, that will take your ethical phishing campaigns to the next level. <https://github.com/drk1wi/Modlishka>

Watering Hole Attack

- BeEF is short for The Browser Exploitation Framework. It is a penetration testing tool that focuses on the web browser. <https://github.com/beefproject/beef>

Command and Control

Remote Access Tools

- Cobalt Strike is software for Adversary Simulations and Red Team Operations. <https://cobaltstrike.com/>
- Empire is a post-exploitation framework that includes a pure-PowerShell2.0 Windows agent, and a pure Python 2.6/2.7 Linux/OS X agent. <https://github.com/EmpireProject/Empire>
- Metasploit Framework is a computer security project that provides information about security vulnerabilities and aids in penetration testing and IDS signature development. <https://github.com/rapid7/metasploit-framework>
- SILENTTRINITY A post-exploitation agent powered by Python, IronPython, C#/.NET. <https://github.com/byt3bl33d3r/SILENTTRINITY>

- Pupy is an opensource, cross-platform (Windows, Linux, OSX, Android) remote administration and post-exploitation tool mainly written in python. <https://github.com/n1nj4sec/pupy>
- Koadic or COM Command & Control, is a Windows post-exploitation rootkit similar to other penetration testing tools such as Meterpreter and Powershell Empire. <https://github.com/zerosum0x0/koadic>
- PoshC2 is a proxy aware C2 framework written completely in PowerShell to aid penetration testers with red teaming, post-exploitation and lateral movement. <https://github.com/nettitude/PoshC2>
- Gcat a stealthy Python based backdoor that uses Gmail as a command and control server. <https://github.com/byt3bl33d3r/gcat>
- TrevorC2 is a legitimate website (browsable) that tunnels client/server communications for covert command execution. <https://github.com/trustedsec/trevorc2>
- Merlin is a cross-platform post-exploitation HTTP/2 Command & Control server and agent written in golang. <https://github.com/Ne0nd0g/merlin>
- Quasar is a fast and light-weight remote administration tool coded in C#. Providing high stability and an easy-to-use user interface, Quasar is the perfect remote administration solution for you. <https://github.com/quasar/QuasarRAT>

Staging

- Rapid Attack Infrastructure (RAI) Red Team Infrastructure... Quick... Fast... SimplifiedOne of the most tedious phases of a Red Team Operation is usually the infrastructure setup. This usually entails a team server or controller, domains, redirectors, and a Phishing server. <https://github.com/obscuritylabs/RAI>
- Red Baron is a set of modules and custom/third-party providers for Terraform which tries to automate creating resilient, disposable, secure and agile infrastructure for Red Teams. <https://github.com/byt3bl33d3r/Red-Baron>
- EvilURL generate unicode evil domains for IDN Homograph Attack and detect them. <https://github.com/UndeadSec/EvilURL>
- Domain Hunter checks expired domains, bluecoat categorization, and Archive.org history to determine good candidates for phishing and C2 domain names. <https://github.com/threatexpress/domainhunter>
- PowerDNS is a simple proof of concept to demonstrate the execution of PowerShell script using DNS only. <https://github.com/mdsecactivebreach/PowerDNS>
- Chameleon a tool for evading Proxy categorisation. <https://github.com/mdsecactivebreach/Chameleon>
- CatMyFish Search for categorized domain that can be used during red teaming engagement. Perfect to setup whitelisted domain for your Cobalt Strike beacon C&C. <https://github.com/Mr-Un1k0d3r/CatMyFish>
- Malleable C2 is a domain specific language to redefine indicators in Beacon's communication. <https://github.com/rsmudge/Malleable-C2-Profiles>
- Malleable-C2-Randomizer This script randomizes Cobalt Strike Malleable C2 profiles through the use of a metalanguage, hopefully reducing the chances of flagging signature-based detection controls. <https://github.com/bluscreenofjeff/Malleable-C2-Randomizer>
- FindFrontableDomains search for potential frontable domains. <https://github.com/rvrsh3ll/FindFrontableDomains>

- Postfix-Server-Setup Setting up a phishing server is a very long and tedious process. It can take hours to setup, and can be compromised in minutes. <https://github.com/n0pe-sled/Postfix-Server-Setup>
- DomainFrontingLists a list of Domain Frontable Domains by CDN. <https://github.com/vysec/DomainFrontingLists>
- Apache2-Mod-Rewrite-Setup Quickly Implement Mod-Rewrite in your infrastructure. <https://github.com/n0pe-sled/Apache2-Mod-Rewrite-Setup>
- mod_rewrite rule to evade vendor sandboxes. <https://gist.github.com/curiousJack/971385e8334e189d93a6cb4671238b10>
- external_c2 framework a python framework for usage with Cobalt Strike's External C2. https://github.com/Und3rf10w/external_c2_framework
- ExternalC2 a library for integrating communication channels with the Cobalt Strike External C2 server. <https://github.com/ryhanson/ExternalC2>
- cs2modrewrite a tools for convert Cobalt Strike profiles to modrewrite scripts. <https://github.com/threatexpress/cs2modrewrite>
- e2modrewrite a tools for convert Empire profiles to Apache modrewrite scripts. <https://github.com/infosecninja/e2modrewrite>
- redi automated script for setting up CobaltStrike redirectors (nginx reverse proxy, letsencrypt). <https://github.com/taherio/redi>
- Domain Fronting Google App Engine. <https://github.com/redteam-cyberark/Google-Domain-fronting>
- DomainFrontDiscover Scripts and results for finding domain frontable CloudFront domains. <https://github.com/peewpw/DomainFrontDiscover>
- Automated Empire Infrastructure <https://github.com/bneg/RedTeam-Automation>
- Serving Random Payloads with NGINX. <https://gist.github.com/jivoi/a33ace2e25515a31aa2ffbae246d98c9>
- meek is a blocking-resistant pluggable transport for Tor. It encodes adata stream as a sequence of HTTPS requests and responses. <https://github.com/arlolra/meek>
- CobaltStrike-ToolKit Some useful scripts for CobaltStrike. <https://github.com/killswitch-GUI/CobaltStrike-ToolKit>
- mkhtaccess_red Auto-generate an HTaccess for payload delivery -- automatically pulls ips/nets/etc from known sandbox companies/sources that have been seen before, and redirects them to a benign payload. https://github.com/violentlydave/mkhtaccess_red
- RedFile a flask wsgi application that serves files with intelligence, good for serving conditional RedTeam payloads. <https://github.com/outflanknl/RedFile>
- keyserver Easily serve HTTP and DNS keys for proper payload protection. <https://github.com/leolooeek/keyserver>
- DoHC2 allows the ExternalC2 library from Ryan Hanson (<https://github.com/ryhanson/ExternalC2>) to be leveraged for command and control (C2) via DNS over HTTPS (DoH). This is built for the popular Adversary Simulation and Red Team Operations Software Cobalt Strike (<https://www.cobaltstrike.com>). <https://github.com/SpiderLabs/DoHC2>
- cat-sites Library of sites for categorization. <https://github.com/audrummer15/cat-sites>

Lateral Movement

- CrackMapExec is a swiss army knife for pentesting networks. <https://github.com/byt3bl33d3r/CrackMapExec>

- PowerLessShell rely on MSBuild.exe to remotely execute PowerShell scripts and commands without spawning powershell.exe. <https://github.com/Mr-Un1k0d3r/PowerLessShell>
- GoFetch is a tool to automatically exercise an attack plan generated by the BloodHound application. <https://github.com/GoFetchAD/GoFetch>
- ANGRYPUPPY a bloodhound attack path automation in CobaltStrike. <https://github.com/vysec/ANGRYPUPPY>
- DeathStar is a Python script that uses Empire's RESTful API to automate gaining Domain Admin rights in Active Directory environments using a variety of techniques. <https://github.com/byt3bl33d3r/DeathStar>
- SharpHound C# Rewrite of the BloodHound Ingestor. <https://github.com/BloodHoundAD/SharpHound>
- BloodHound.py is a Python based ingestor for BloodHound, based on Impacket. <https://github.com/fox-it/BloodHound.py>
- Responder is a LLMNR, NBT-NS and MDNS poisoner, with built-in HTTP/SMB/MSSQL/FTP/LDAP rogue authentication server supporting NTLMv1/NTLMv2/LMv2, Extended Security NTLMSSP and Basic HTTP authentication. <https://github.com/SpiderLabs/Responder>
- SessionGopher is a PowerShell tool that uses WMI to extract saved session information for remote access tools such as WinSCP, PuTTY, SuperPuTTY, FileZilla, and Microsoft Remote Desktop. It can be run remotely or locally. <https://github.com/fireeye/SessionGopher>
- PowerSploit is a collection of Microsoft PowerShell modules that can be used to aid penetration testers during all phases of an assessment. <https://github.com/PowerShellMafia/PowerSploit>
- Nishang is a framework and collection of scripts and payloads which enables usage of PowerShell for offensive security, penetration testing and red teaming. Nishang is useful during all phases of penetration testing. <https://github.com/samratashok/nishang>
- Inveigh is a Windows PowerShell LLMNR/mDNS/NBNS spoofer/man-in-the-middle tool. <https://github.com/Kevin-Robertson/Inveigh>
- PowerUpSQL a PowerShell Toolkit for Attacking SQL Server. <https://github.com/NetSPI/PowerUpSQL>
- MailSniper is a penetration testing tool for searching through email in a Microsoft Exchange environment for specific terms (passwords, insider intel, network architecture information, etc.). <https://github.com/dafthack/MailSniper>
- WMIOps is a powershell script that uses WMI to perform a variety of actions on hosts, local or remote, within a Windows environment. It's designed primarily for use on penetration tests or red team engagements. <https://github.com/ChrisTruncer/WMIOps>
- Mimikatz is an open-source utility that enables the viewing of credential information from the Windows lsass. <https://github.com/gentilkiwi/mimikatz>
- LaZagne project is an open source application used to retrieve lots of passwords stored on a local computer. <https://github.com/AlessandroZ/LaZagne>
- mimipenguin a tool to dump the login password from the current linux desktop user. Adapted from the idea behind the popular Windows tool mimikatz. <https://github.com/huntergregal/mimipenguin>
- PsExec is a light-weight telnet-replacement that lets you execute processes on other systems, complete with full interactivity for console applications, without having to manually install client software. <https://docs.microsoft.com/en-us/sysinternals/downloads/psexec>

- KeeThief allows for the extraction of KeePass 2.X key material from memory, as well as the backdooring and enumeration of the KeePass trigger system. <https://github.com/HarmJ0y/KeeThief>
- PSAttack combines some of the best projects in the infosec powershell community into a self contained custom PowerShell console. <https://github.com/jaredhaight/PSAttack>
- Internal Monologue Attack Retrieving NTLM Hashes without Touching LSASS. <https://github.com/eladshamir/Internal-Monologue>
- Impacket is a collection of Python classes for working with network protocols. Impacket is focused on providing low-level programmatic access to the packets and for some protocols (for instance NMB, SMB1-3 and MS-DCERPC) the protocol implementation itself. <https://github.com/CoreSecurity/impacket>
- icebreaker gets plaintext Active Directory credentials if you're on the internal network but outside the AD environment. <https://github.com/DanMcInerney/icebreaker>
- Living Off The Land Binaries and Scripts (and now also Libraries) The goal of these lists are to document every binary, script and library that can be used for other purposes than they are designed to. <https://github.com/api0cradle/LOLBAS>
- WSUSpendu for compromised WSUS server to extend the compromise to clients. <https://github.com/AlsidOfficial/WSUSpendu>
- Evilgrade is a modular framework that allows the user to take advantage of poor upgrade implementations by injecting fake updates. <https://github.com/infobyte/evilgrade>
- NetRipper is a post exploitation tool targeting Windows systems which uses API hooking in order to intercept network traffic and encryption related functions from a low privileged user, being able to capture both plain-text traffic and encrypted traffic before encryption/after decryption. <https://github.com/NyTROST/NetRipper>
- LethalHTA Lateral Movement technique using DCOM and HTA. <https://github.com/codewhitesec/LethalHTA>
- Invoke-PowerThIEf an Internet Explorer Post Exploitation library. <https://github.com/nettitude/Invoke-PowerThIEf>
- RedSnarf is a pen-testing / red-teaming tool for Windows environments. <https://github.com/nccgroup/redsnarf>
- HoneypotBuster Microsoft PowerShell module designed for red teams that can be used to find honeypots and honeytokens in the network or at the host. <https://github.com/JavelinNetworks/HoneypotBuster>

Establish Foothold

- Tunna is a set of tools which will wrap and tunnel any TCP communication over HTTP. It can be used to bypass network restrictions in fully firewalled environments. <https://github.com/SECFORCE/Tunna>
- reGeorg the successor to reDuh, pwn a bastion webserver and create SOCKS proxies through the DMZ. Pivot and pwn. <https://github.com/sensepost/reGeorg>
- Blade is a webshell connection tool based on console, currently under development and aims to be a choice of replacement of Chooper. <https://github.com/wonderqs/Blade>
- TinyShell Web Shell Framework. <https://github.com/threatexpress/tinyshell>
- PowerLurk is a PowerShell toolset for building malicious WMI Event Subscriptions. <https://github.com/Sw4mpf0x/PowerLurk>

- DAMP The Discretionary ACL Modification Project: Persistence Through Host-based Security Descriptor Modification. <https://github.com/HarmJ0y/DAMP>

Escalate Privileges

Domain Escalation

- PowerView is a PowerShell tool to gain network situational awareness on Windows domains. <https://github.com/PowerShellMafia/PowerSploit/blob/master/Recon/PowerView.ps1>
- Get-GPPPassword Retrieves the plaintext password and other information for accounts pushed through Group Policy Preferences. <https://github.com/PowerShellMafia/PowerSploit/blob/master/Exfiltration/Get-GPPPassword.ps1>
- Invoke-ACLPwn is a tool that automates the discovery and pwnage of ACLs in Active Directory that are unsafe configured. <https://github.com/fox-it/Invoke-ACLPwn>
- BloodHound uses graph theory to reveal the hidden and often unintended relationships within an Active Directory environment. <https://github.com/BloodHoundAD/BloodHound>
- PyKEK (Python Kerberos Exploitation Kit), a python library to manipulate KRB5-related data. <https://github.com/SecWiki/windows-kernel-exploits/tree/master/MS14-068/pykek>
- Grouper a PowerShell script for helping to find vulnerable settings in AD Group Policy. <https://github.com/l0ss/Grouper>
- ADRecon is a tool which extracts various artifacts (as highlighted below) out of an AD environment in a specially formatted Microsoft Excel report that includes summary views with metrics to facilitate analysis. <https://github.com/sense-of-security/ADRecon>
- ADACLScanner one script for ACL's in Active Directory. <https://github.com/canix1/ADACLScanner>
- ACLight a useful script for advanced discovery of Domain Privileged Accounts that could be targeted - including Shadow Admins. <https://github.com/cyberark/ACLightv>
- LAPSToolkit a tool to audit and attack LAPS environments. <https://github.com/leoloobeek/LAPSToolkit>
- PingCastle is a free, Windows-based utility to audit the risk level of your AD infrastructure and check for vulnerable practices. <https://www.pingcastle.com/download>
- RiskySPNs is a collection of PowerShell scripts focused on detecting and abusing accounts associated with SPNs (Service Principal Name). <https://github.com/cyberark/RiskySPN>
- Mystique is a PowerShell tool to play with Kerberos S4U extensions, this module can assist blue teams to identify risky Kerberos delegation configurations as well as red teams to impersonate arbitrary users by leveraging KCD with Protocol Transition. <https://github.com/machosec/Mystique>
- Rubeus is a C# toolset for raw Kerberos interaction and abuses. It is heavily adapted from Benjamin Delpy's Kekeo project. <https://github.com/GhostPack/Rubeus>
- kekeo is a little toolbox I have started to manipulate Microsoft Kerberos in C (and for fun). <https://github.com/gentilkiwi/kekeo>

Local Escalation

- UACMe is an open source assessment tool that contains many methods for bypassing Windows User Account Control on multiple versions of the operating system. <https://github.com/hfiref0x/UACME>
- windows-kernel-exploits a collection windows kernel exploit. <https://github.com/SecWiki/windows-kernel-exploits>
- PowerUp aims to be a clearinghouse of common Windows privilege escalation vectors that rely on misconfigurations. <https://github.com/PowerShellMafia/PowerSploit/blob/master/Privesc/PowerUp.ps1>
- The Elevate Kit demonstrates how to use third-party privilege escalation attacks with Cobalt Strike's Beacon payload. <https://github.com/rsmudge/ElevateKit>
- Sherlock a powerShell script to quickly find missing software patches for local privilege escalation vulnerabilities. <https://github.com/rasta-mouse/Sherlock>
- Tokenvator a tool to elevate privilege with Windows Tokens. <https://github.com/0xbadjuju/Tokenvator>

Data Exfiltration

- CloakifyFactory & the Cloakify Toolset - Data Exfiltration & Infiltration In Plain Sight; Evade DLP/MLS Devices; Social Engineering of Analysts; Defeat Data Whitelisting Controls; Evade AV Detection. <https://github.com/TryCatchHCF/Cloakify>
- DET (is provided AS IS), is a proof of concept to perform Data Exfiltration using either single or multiple channel(s) at the same time. <https://github.com/sensepost/DET>
- DNSExfiltrator allows for transferring (exfiltrate) a file over a DNS request covert channel. This is basically a data leak testing tool allowing to exfiltrate data over a covert channel. <https://github.com/Arno0x/DNSExfiltrator>
- PyExfil a Python Package for Data Exfiltration. <https://github.com/ytisf/PyExfil>
- Egress-Assess is a tool used to test egress data detection capabilities. <https://github.com/ChrisTruncer/Egress-Assess>
- Powershell RAT python based backdoor that uses Gmail to exfiltrate data as an e-mail attachment. <https://github.com/Viralmaniar/Powershell-RAT>

Misc

Wireless Networks

- Wifiphisher is a security tool that performs Wi-Fi automatic association attacks to force wireless clients to unknowingly connect to an attacker-controlled Access Point. <https://github.com/wifiphisher/wifiphisher>
- Evilginx is a man-in-the-middle attack framework used for phishing credentials and session cookies of any web service. <https://github.com/kgretzky/evilginx>

- mana toolkit for wifi rogue AP attacks and MitM. <https://github.com/sensepost/mana>

Embedded & Peripheral Devices Hacking

- magspoof a portable device that can spoof/emulate any magnetic stripe, credit card or hotel card "wirelessly", even on standard magstripe (non-NFC/RFID) readers. <https://github.com/samyk/magspoof>
- WarBerryPi was built to be used as a hardware implant during red teaming scenarios where we want to obtain as much information as possible in a short period of time with being as stealth as possible. <https://github.com/secgroundzero/warberry>
- P4wnP1 is a highly customizable USB attack platform, based on a low cost Raspberry Pi Zero or Raspberry Pi Zero W (required for HID backdoor). <https://github.com/mame82/P4wnP1>
- malusb HID spoofing multi-OS payload for Teensy. <https://github.com/ebursztein/malusb>
- Fenrir is a tool designed to be used "out-of-the-box" for penetration tests and offensive engagements. Its main feature and purpose is to bypass wired 802.1x protection and to give you an access to the target network. <https://github.com/Orange-Cyberdefense/fenrir-ocd>
- poisonsnap exploits locked/password protected computers over USB, drops persistent WebSocket-based backdoor, exposes internal router, and siphons cookies using Raspberry Pi Zero & Node.js. <https://github.com/samyk/poisonsnap>
- WHID WiFi HID Injector - An USB Rubberduddy / BadUSB On Steroids. <https://github.com/whid-injector/WHID>

Software For Team Communication

- RocketChat is free, unlimited and open source. Replace email & Slack with the ultimate team chat software solution. <https://rocket.chat>
- Etherpad is an open source, web-based collaborative real-time editor, allowing authors to simultaneously edit a text document <https://etherpad.net>

Log Aggregation

- RedELK Red Team's SIEM - easy deployable tool for Red Teams used for tracking and alarming about Blue Team activities as well as better usability in long term operations. <https://github.com/outflanknl/RedELK/>
- CobaltSplunk Splunk Dashboard for CobaltStrike logs. <https://github.com/vyseccobaltsplunk/CobaltSplunk>
- Red Team Telemetry A collection of scripts and configurations to enable centralized logging of red team infrastructure. https://github.com/ztgrace/red_team_telemetry
- Elastic for Red Teaming Repository of resources for configuring a Red Team SIEM using Elastic. <https://github.com/SecurityRiskAdvisors/RedTeamSIEM>

C# Offensive Framework

- SharpSploit is a .NET post-exploitation library written in C# that aims to highlight the attack surface of .NET and make the use of offensive .NET easier for red teamers.

<https://github.com/cobbr/SharpSploit>

- GhostPack is (currently) a collection various C# implementations of previous PowerShell functionality, and includes six separate toolsets being released today- Seatbelt, SharpUp, SharpRoast, SharpDump, SafetyKatz, and SharpWMI. <https://github.com/GhostPack>
- SharpWeb .NET 2.0 CLR project to retrieve saved browser credentials from Google Chrome, Mozilla Firefox and Microsoft Internet Explorer/Edge. <https://github.com/djhohnstein/SharpWeb>
- reconerator C# Targeted Attack Reconnaissance Tools. <https://github.com/stufus/reconerator>
- SharpView C# implementation of harmj0y's PowerView. <https://github.com/tevora-threat/SharpView>
- Watson is a (.NET 2.0 compliant) C# implementation of Sherlock. <https://github.com/rasta-mouse/Watson>

Labs

- Detection Lab This lab has been designed with defenders in mind. Its primary purpose is to allow the user to quickly build a Windows domain that comes pre-loaded with security tooling and some best practices when it comes to system logging configurations. <https://github.com/clong/DetectionLab>
- Modern Windows Attacks and Defense Lab This is the lab configuration for the Modern Windows Attacks and Defense class that Sean Metcalf (@pyrotek3) and I teach. <https://github.com/jaredhaight/WindowsAttackAndDefenseLab>
- Invoke-UserSimulator Simulates common user behaviour on local and remote Windows hosts. <https://github.com/ubeeri/Invoke-UserSimulator>
- Invoke-ADLabDeployer Automated deployment of Windows and Active Directory test lab networks. Useful for red and blue teams. <https://github.com/outflanknl/Invoke-ADLabDeployer>
- Sheepl Creating realistic user behaviour for supporting tradecraft development within lab environments. <https://github.com/SpiderLabs/sheepl>

Scripts

- Aggressor Scripts is a scripting language for red team operations and adversary simulations inspired by scriptable IRC clients and bots.
- <https://github.com/invokethreatguy/CSASC>
- <https://github.com/secgroundzero/CS-Aggressor-Scripts>
- <https://github.com/Und3rf10w/Aggressor-scripts>
- <https://github.com/harleyQu1nn/AggressorScripts>
- <https://github.com/rasta-mouse/Aggressor-Script>
- <https://github.com/RhinoSecurityLabs/Aggressor-Scripts>

- <https://github.com/bluscreenofjeff/AggressorScripts>
- https://github.com/001SPARTaN/aggressor_scripts
- <https://github.com/360-A-Team/CobaltStrike-Toolset>
- A collection scripts useful for red teaming and pentesting
- <https://github.com/FuzzySecurity/PowerShell-Suite>
- <https://github.com/nettitude/Powershell>
- <https://github.com/Mr-Un1k0d3r/RedTeamPowershellScripts>
- <https://github.com/threatexpress/red-team-scripts>
- <https://github.com/SadProcessor/SomeStuff>
- <https://github.com/rvrsh3ll/Misc-Powershell-Scripts>
- <https://github.com/enigma0x3/Misc-PowerShell-Stuff>
- <https://github.com/ChrisTruncer/PenTestScripts>
- <https://github.com/bluscreenofjeff/Scripts>
- <https://github.com/xorrior/RandomPS-Scripts>
- <https://github.com/xorrior/Random-CSharpTools>
- <https://github.com/leechristensen/Random>
- <https://github.com/mgeeky/Penetration-Testing-Tools/tree/master/social-engineering>

C2 Deployment Plan

One of the most comfortable and free C2 out there is SilentTrinity:

Instalation:

- Intalar python 3.7

```
sudo apt-get install build-essential checkinstall
```

```
sudo apt-get install libreadline-gplv2-dev libncursesw5-dev libssl-dev \
    libsqlite3-dev tk-dev libgdbm-dev libc6-dev libbz2-dev libffi-dev zlib1g-dev
```

```
cd /usr/src
```

```
sudo wget https://www.python.org/ftp/python/3.7.7/Python-3.7.7.tgz
```

```
sudo tar xzf Python-3.7.7.tgz
```

```
cd Python-3.7.7
```

```
sudo ./configure --enable-optimizations
```

```
sudo make altinstall
```

```
python3.7 -
```

```
V
```

- Install impacket

```
sudo git clone https://github.com/SecureAuthCorp/impacket.git
cd impacket
sudo pip3 install -r require.txt
sudo python3 setup.py install
```

- Instalar SilentTRINITY

```
git clone https://github.com/byt3bl33d3r/SILENTTRINITY
cd SILENTTRINITY
python3.7 -m pip install -r requirements.txt
sudo pip3 install -r requirements.txt
```

Homework #4

Engagement Phases Document

This template can be used as main board to determine the status of the operation based on objectives:

Phases	Objectives
Engagement Planning	- Finish terms with Trusted cell, and board
Engagement Execution	-Apply the new log team's pr actions, reach
Engagement Culmination	- Defined obj terms in ROE actor simulat
Engagement Reporting	- Prepare and information, board, contra

Engagement Removal Checklist

This template will allow you to track and then delete all the files, tools, and different modifications made in the environment to make sure is clean:

File system modifications

Access and backdoors

Files dropped on operator's tools

File artifacts generated by the tools

System copies

Registry key modifications

Launch files

Startup scripts

Execution mechanisms

Log files generated on victim system

C2 Persistence mechanisms

Item

Description

C2 channels	
Connection monitoring and mechanisms	
Item	Description

In the middle column, the files or artifacts impacted would be detailed, during the entire engagement this document gets filled and the status should be checked and everything deleted at the end.

Executive Brief

As soon as the operation ends, it must be briefed to the executives findings and discoveries. These are the most important aspects to bring up considering people with no technical background can be present:

- Include organizational management (desicion makers) in the room
- Include key information security and technical staff
- Include legal staff
- Focus on chronological summary of observations

- Highlight critical observations
- Inform the audience this is a summary and everything will come in the report
- Interact with the blue team to educate and show where they might have missed the RT

Technical Brief

On this detailed meeting, you can explain the staff how the threat was impersonated and the objectives were reached.

- Explain the TPPs and intended IOCs
- Initial thought on process and objectives
- Steps through actions and activity with commands
- Why were actions executed
- Explain how each action triggered the next
- Recommendations and techniques to limit each action

allow the defensive team to clarify actions:

- ask how and why
- process to secure and defend
- identify alerts, triggers, anomalies
- steps through blue actions
- Identify how red could have been detected
- feedback to red on actions and recommendations
- post engagement analysis after report delivered

Homework #5

Report Template Sample

Sample: https://penconsultants.com/home/wp-content/uploads/2020/02/sampleReport_PENConsultants.pdf

Sections:

Executive Summary.....	4
Introduction.....	6
Statement of Work ("SOW").....	6
Scope.....	6
Date(s) of testing.....	6
Testers.....	7
Testing Source IPs.....	7
Labor Hours.....	7
Risk Rating Methodology.....	7
Pretext.....	8
General Pretext.....	8
Timeline.....	8
Findings and Recommendations.....	9
Assumptions.....	9
Limitations.....	10
Summary - Testing Results.....	10
Summary of Findings.....	11
Summary - Recommendations.....	11
Post-Testing Tasks (for Client).....	13
FR-001: Log Verbosity.....	13
FR-002: Elasticsearch.....	14
FR-003: Email Server Misconfiguration.....	15
FR-004: Brute-Force User Enumeration.....	17
FR-005: MFA.....	18
FR-006: Session Termination.....	20
FR-007: IWA Scope.....	21
FR-008: Account Lockout.....	22
FR-009: Application Password Policy.....	22
FR-010: AD Password Policy.....	25
FR-011: Reflected Cross Site Scripting (XSS).....	26
FR-012: SQL Injection.....	28
FR-013: Clickjacking.....	30
FR-014: Upload of Malicious Files.....	32
FR-015: Phishing Assessment.....	35
FR-016: Media Drop.....	39
FR-017: Unsupervised Access.....	40
Attack Scenarios.....	41
Overview.....	41
Attack #1 - SQLi:.....	41
Attack #2 - Well-known vulnerabilities:.....	42
Attack #3 - VPN:.....	42
Attack #4 - MiTM:.....	42

Observations

You can constantly add observations during the entire engagement to a file than later can be added as a section for the blue team in the final report.

Example:

```
*****
3.1 Observation: Client to client lateral movement      *
Rating: Category 2                                     *
                                                         *
Description:                                           *
                                                         *
Recomendation:                                         *
*****
```

Findings

The findings is another document that can be converted into a section of the final report. This will include vulnerabilities discovered in a pentest report format:

Vulnerability Summary

Quick View

The table below is designed to provide a quick view of all the identified findings and their respective risk ratings. Please see the following section for a detailed listing of the identified findings.

For information regarding our risk rating methodology, please see [Appendix B](#).

#	Finding Title	Instances	Rating
1.	Shared Local Administrator Password	1	Critical (8)
2.	SMB Signing Not Enabled	9	High (7)
3.	DNS Cache Snooping	3	Medium (4)
4.	Apache mod_negotiation (Apache MultiViews)	1	Low (1.25)

Total Findings: 14

This is the example of one finding:

Finding(s)

1. Shared Local Administrator Password | Critical (8)

Description:

During the internal testing, it was determined that the local Administrator password is shared among more than one computer. The local Administrator account is installed by default on Windows with the password set during the operating system setup. The account has full access to all files on the system.

Impact:

Generally, automated tools are used to install Windows in larger organizations. This causes an issue since all of the local Administrator passwords are the same unless changed after installation. If an attacker were to gain access to one system and gain the local Administrator password or hashed password (encrypted password) then all systems could easily be compromised. This is one of the most prevalent avenues for an attacker to pivot and escalate inside of an internal network.

Test(s) Conducted:

After accessing a system, each username and password hash is used from the system to attempt to log into other systems.

Finding Comments:

RedTeam was able to use a Metasploit module called PSEXEC to perform a pass-the-hash attack against each of the systems within the 192.168.1.0/24 network. This module allowed for testers to remotely gain access to the systems because of a shared administrator password.

Recommendations:

Utilize a solution that changes all local Administrator passwords regularly. LAPS (local Administrator password solution) is a tool which could be used to remediate this vulnerability. Alternatively, some other enterprise level password management tools can also help ensure you are not using shared passwords.

Affected System(s):

192.168.1.0/24

Affected System(s):

192.168.1.0/24

Instance(s):

1

Status:

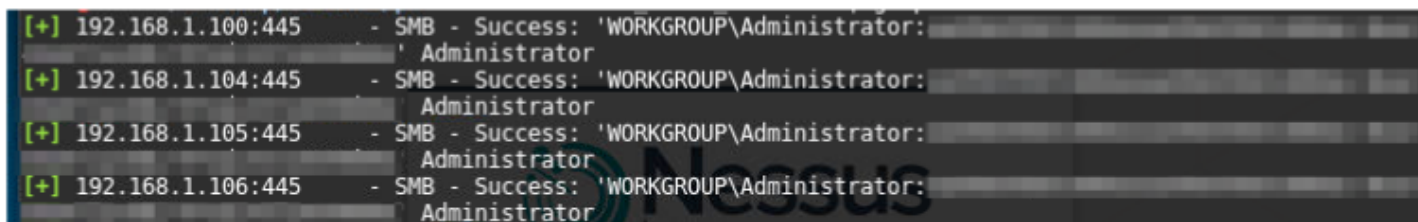
Not Remediated

Evidence:

RedTeam Security

6 of 21

ACME Corporation



```
[+] 192.168.1.100:445 - SMB - Success: 'WORKGROUP\Administrator:
Administrator
[+] 192.168.1.104:445 - SMB - Success: 'WORKGROUP\Administrator:
Administrator
[+] 192.168.1.105:445 - SMB - Success: 'WORKGROUP\Administrator:
Administrator
[+] 192.168.1.106:445 - SMB - Success: 'WORKGROUP\Administrator:
Administrator
```

Evidence notes:

The above screen capture shows that RedTeam was able to successfully authenticate to multiple systems using the same username and password.

Severity Calculation:

The process for calculating the finding's severity is derived by assigning a numeric value between 0 and 9 to four (4) criteria separated into Likelihood and Impact. The formula is best represented here: **Likelihood**(Threat Agents + Vulnerability Factors) / 2 + **Impact**(Technical Impact + Business Impact) / 2 = **Risk Rating**(Likelihood + Impact) / 2

Critical (8) = (Likelihood (8 + 8) / 2 = **Critical (8)** + Impact (8 + 8) / 2 = **Critical (8)**) / 2

Evidence notes:

The above screen capture shows that RedTeam was able to successfully authenticate to multiple systems using the same username and password.

Severity Calculation:

The process for calculating the finding's severity is derived by assigning a numeric value between 0 and 9 to four (4) criteria separated into Likelihood and Impact. The formula is best represented here: **Likelihood**(Threat Agents + Vulnerability Factors) /2 + **Impact**(Technical Impact + Business Impact) /2 = **Risk Rating**(Likelihood + Impact) /2

Critical (8) = (Likelihood (8 + 8) /2 = **Critical (8)** + Impact (8 + 8) /2 = **Critical (8)**) /2

Reference(s):

<https://www.microsoft.com/en-us/download/details.aspx?id=46899>

<https://www.offensive-security.com/metasploit-unleashed/psexec-pass-hash/>

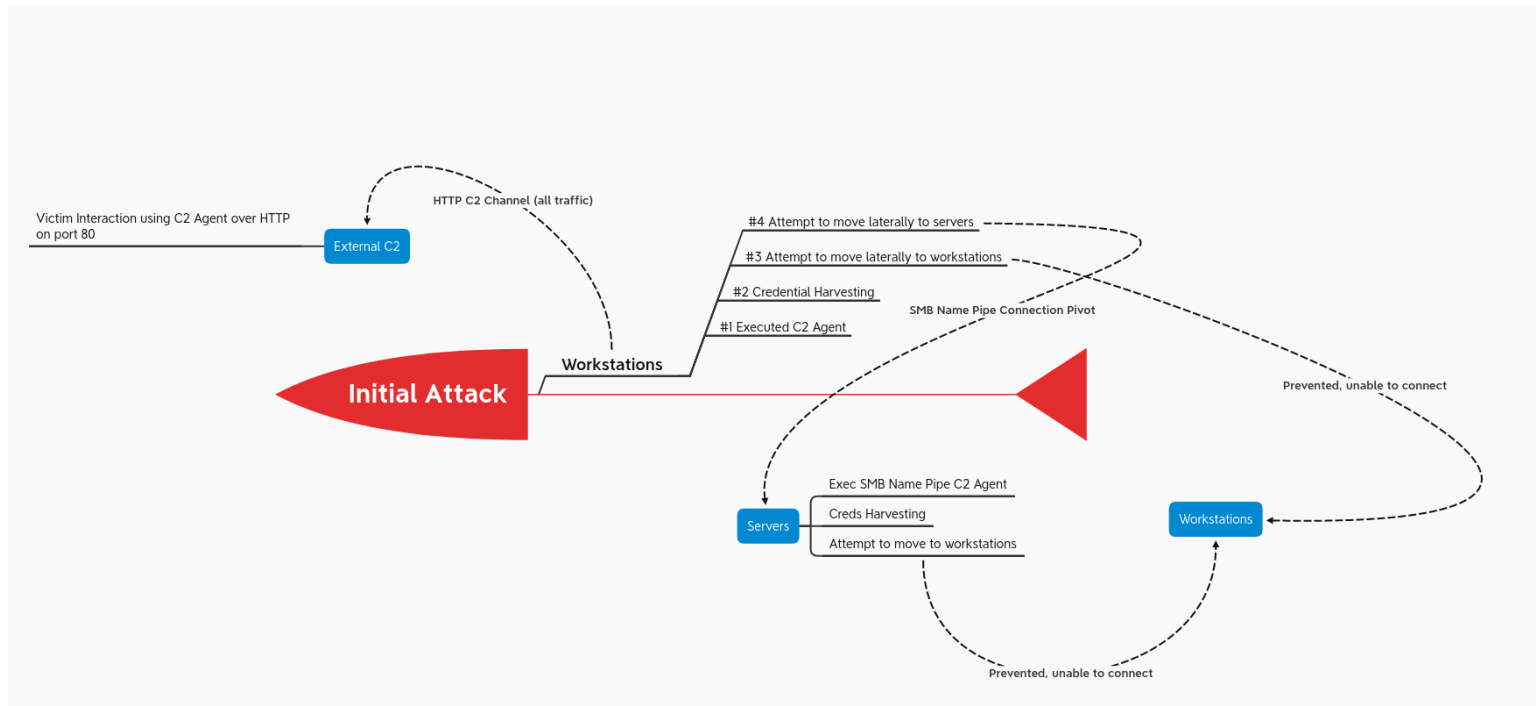
CVSS:

(AV:N/AC:L/Au:S/C:C/I:C/A:C)

[\[Back to Top\]](#)

Attack Flow Diagram

The attack flow diagram can be added to the final report with a narrative after it, detailing how all the engagement was conducted.



Glossary

Side Notes

Side notes to remember interesting concepts learned:

“Is” vs “Should be”

always think pointing to what it should be instead of conforming with “is”. Tools just do something but are they doing what they should be doing.