RED TEAMING & ADVERSARY EMULATION

Tsolmon Z.

#WhoAmI

Information Security Analyst at Unitel,
Pentesting, Red Teaming,
OSCP

Agenda

- Introduction
- Benefits of separating infosec team as red and blue
- Consider Purple Teaming
- Red Team Maturity / Who is Red Teamer
- How does it differ from pentesting
- What is “Adversary Emulation”?
- Why do red teaming & adversary emulation
- Some tools to keep in mind
- MITRE ATT&CK
- Adversary Emulation with Caldera
- What to take away from this presentation

Introduction

Sun Tzu

VI BCE

Modern red teaming

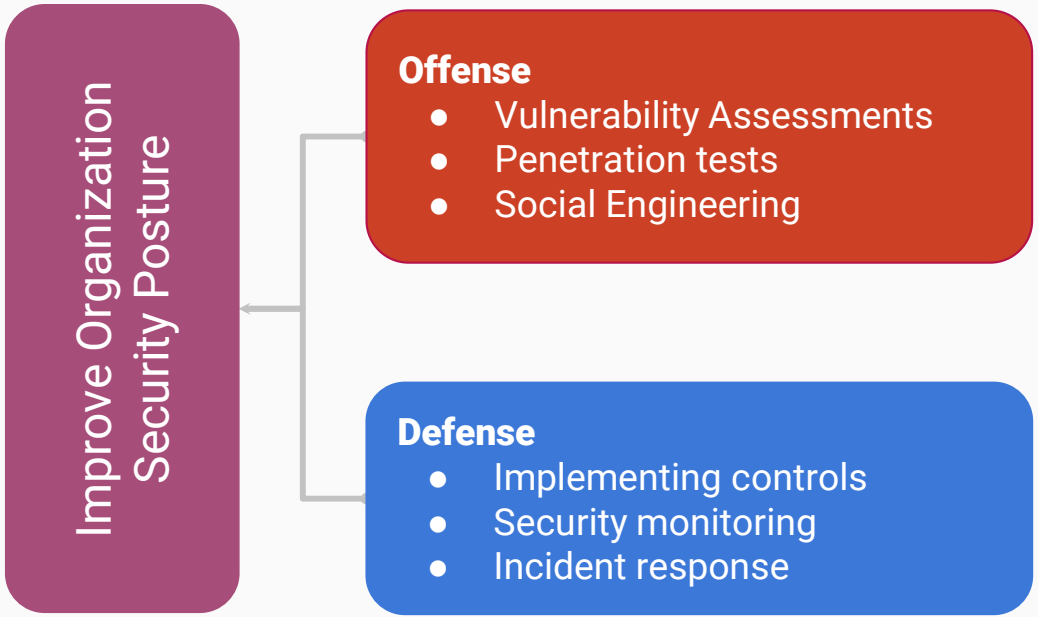
U.S Military

1960s

Red Teaming

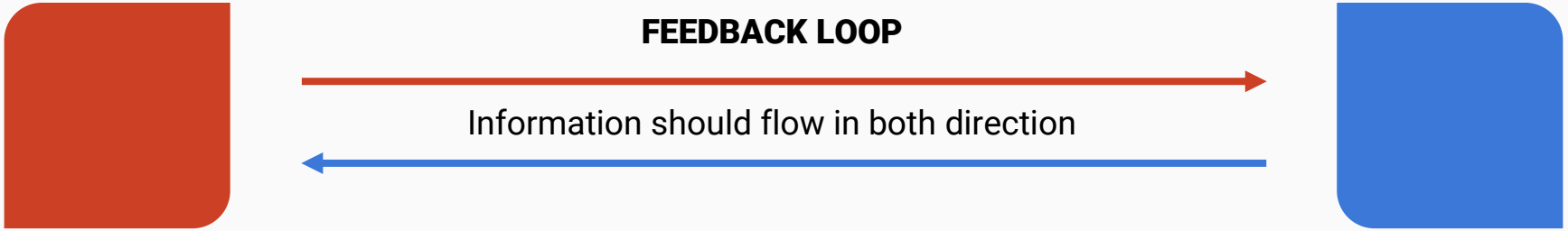
in cyber environment

Benefits of separating infosec team as RED & BLUE



Consider purple teaming

RED		BLUE
Report with many vulnerabilities = Well done!	VS	No alerts mean that preventive controls are working
Success is measured by # of failed controls		Large volume of alerts means detection controls are working
Blue team failure = red team success		Red team failure = blue team success



Red Team Maturity | Who is Red Teamer

Vuln Scan	External Penetration Test	Internal Penetration Test	Purple Team(s)	Red Team / ATT&CK	Non-scoped long term
Patch Management	Network Controls / Admin Rights	Configured endpoint / EDRs	Centralized Logging	Finely tuned Alerting and Response	Threat hunting

How does it differ from pentesting?

- Scope is much **LARGER**
- Much more continuous process (attack <-> defend)
- Full stacks of process (people, technology), much more than just vulnerability assessments / penetration tests

What is Adversary Emulation?

Adversary Emulation is an activity where security experts emulate how an adversary operates. The ultimate goal of course is to improve how resilient the organization is versus these adversary techniques. Both red and purple teaming can be considered as adversary emulation.

Adversary activities are described using **TTPs (Tactics, Techniques & Procedures)**, possibly described using a **kill chain.** TTPs are not as concrete as for example IOCs, but they describe how the adversary operates at a higher level. Adversary emulation should be based on TTPs. As such, a traditional vulnerability scan or internal penetration test that is not based on TTPs should not be considered adversary emulation.

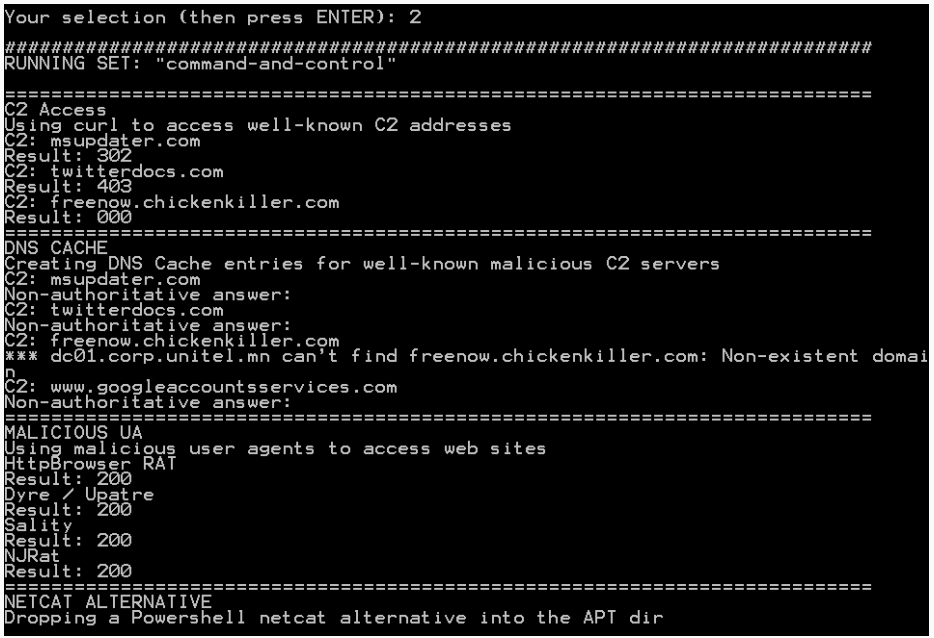
Why do red teaming & adversary emulation?

- Understand your current exposure to a realistic, relevant, threat
- On top of vulnerability identification, assess detection capability as well
- Also includes testing of human reaction as well
- Repeatable, structured process that provides key areas for improvement

Some tools to keep in mind

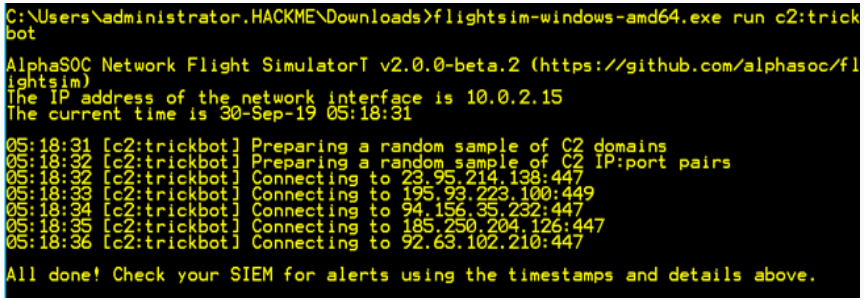
- APTSimulator (Endpoint)
- FlightSim (Network)
- MITRE ATT&CK

APTSimulator



- Windows Batch script that uses a set of tools and output files to make a system look as if it was compromised.
- Don’t need to run a web server, database or any agents

FlightSim



- Generation of malicious network traffic
- Helping security teams to evaluate security controls and network visibility.
- DNS tunneling, DGA traffic, request to know active C2 destinations

MITRE ATT&CK

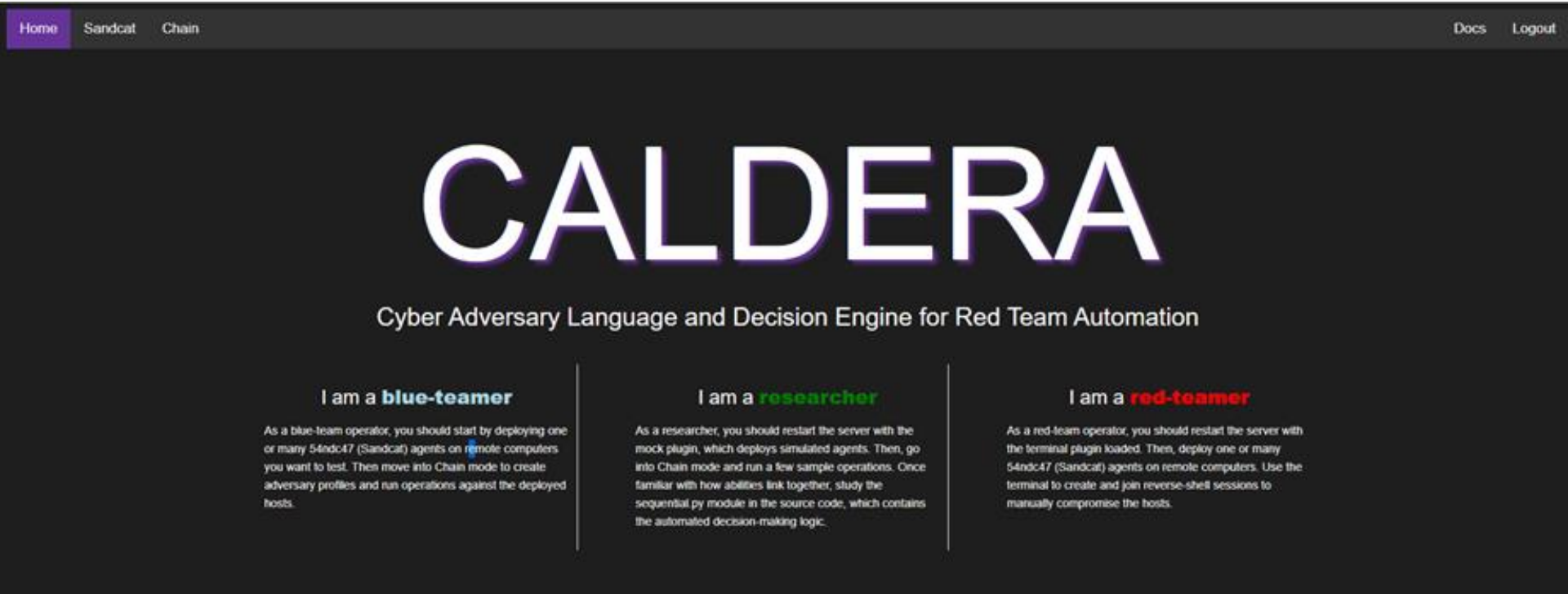
"MITRE ATT&CK is a globally-accessible knowledge base of adversary tactics and techniques based on real-world observations."

ATT&CK Matrix for Enterprise

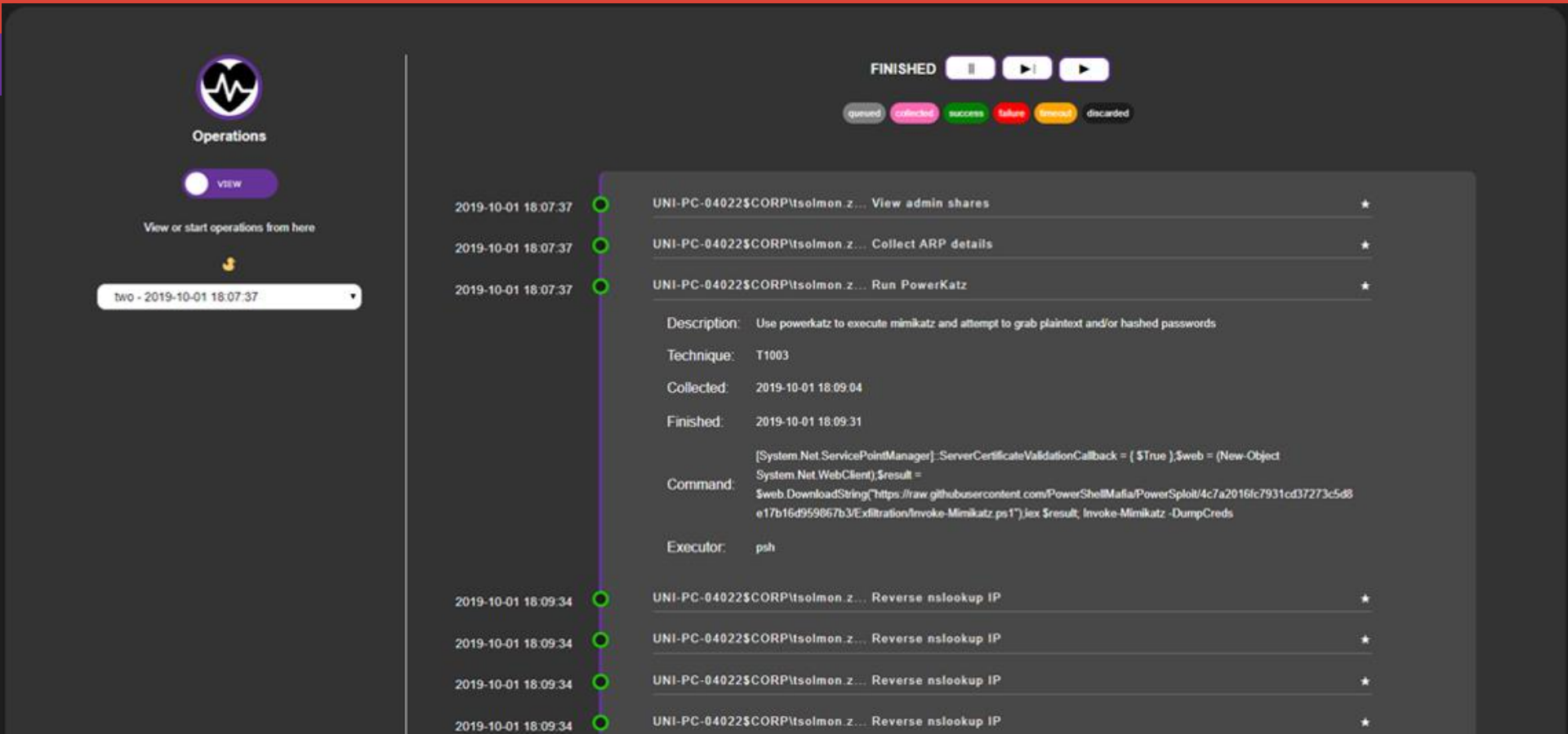
Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Commonly Used Port	Automated Exfiltration	Data Destruction
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Communication Through Removable Media	Data Compressed	Data Encrypted for Impact
External Remote Services	Command-Line Interface	Account Manipulation	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Discovery	Distributed Component Object Model	Clipboard Data	Connection Proxy	Data Encrypted	Defacement
Hardware Additions	Compiled HTML File	AppCert DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	Domain Trust Discovery	Exploitation of Remote Services	Data from Information Repositories	Custom Command and Control Protocol	Data Transfer Size Limits	Disk Content Wipe
Replication Through Removable Media	Control Panel Items	AppInit DLLs	Application Shimming	Clear Command History	Credentials in Files	File and Directory Discovery	Logon Scripts	Data from Local System	Custom Cryptographic Protocol	Exfiltration Over Alternative Protocol	Disk Structure Wipe
Spearphishing Attachment	Dynamic Data Exchange	Application Shimming	Bypass User Account Control	CMSTP	Credentials in Registry	Network Service Scanning	Pass the Hash	Data from Network Shared Drive	Data Encoding	Exfiltration Over Command and Control Channel	Endpoint Denial of Service
Spearphishing Link	Execution through API	Authentication Package	DLL Search Order Hijacking	Code Signing	Exploitation for Credential Access	Network Share Discovery	Pass the Ticket	Data from Removable Media	Data Obfuscation	Exfiltration Over Other Network Medium	Firmware Corruption
Spearphishing via Service	Execution through Module Load	BITS Jobs	Dylib Hijacking	Compile After Delivery	Forced Authentication	Network Sniffing	Remote Desktop Protocol	Data Staged	Domain Fronting	Exfiltration Over Physical Medium	Inhibit System Recovery
Supply Chain Compromise	Exploitation for Client Execution	Bootkit	Exploitation for Privilege Escalation	Compiled HTML File	Hooking	Password Policy Discovery	Remote File Copy	Email Collection	Domain Generation Algorithms	Scheduled Transfer	Network Denial of Service
Trusted Relationship	Graphical User Interface	Browser Extensions	Extra Window Memory Injection	Component Firmware	Input Capture	Peripheral Device Discovery	Remote Services	Input Capture	Fallback Channels		Resource Hijacking
Valid Accounts	InstallUtil	Change Default File Association	File System Permissions Weakness	Component Object Model Hijacking	Input Prompt	Permission Groups Discovery	Replication Through Removable Media	Man in the Browser	Multi-hop Proxy		Runtime Data Manipulation
	Launchctl	Component Firmware	Hooking	Control Panel Items	Kerberoasting	Process Discovery	Shared Webroot	Screen Capture	Multi-Stage Channels		Service Stop
	Local Job Scheduling	Component Object Model Hijacking	Image File Execution Options Injection	DCShadow	Keychain	Query Registry	SSH Hijacking	Video Capture	Multiband Communication		Stored Data Manipulation
	LSASS Driver	Create Account	Launch Daemon	Deobfuscate/Decode Files or Information	LLMNR/NBT-NS Poisoning and Relay	Remote System Discovery	Taint Shared Content		Multilayer Encryption		Transmitted Data Manipulation
	Mshta	DLL Search Order Hijacking	New Service	Disabling Security Tools	Network Sniffing	Security Software Discovery	Third-party Software		Port Knocking		
	PowerShell	Dylib Hijacking	Path Interception	DLL Search Order Hijacking	Password Filter DLL	System Information Discovery	Windows Admin Shares		Remote Access Tools		
	Regsvcs/Regasm	External Remote Services	Plist Modification	DLL Side-Loading	Private Keys	System Network Configuration Discovery	Windows Remote Management		Remote File Copy		

Adversary Emulation with Caldera

CALDERA is an automated adversary emulation system, built on the MITRE ATT&CK™ framework.



Adversary Emulation with Caldera



- Structuring infosec team as **red** and **blue** is efficient.
- Validate that logging the correct events and information doing **red teaming.**
- Adversary emulation can greatly improve your chances of preventing and detecting a breach.

What to take away from this presentation?

Thanks!

For your undivided attention.