MITRE Endpoint Security Testing: Interpreting Results

Prepared by Dr. Edward Amoroso
Chief Executive Officer, TAG Cyber
Distinguished Research Professor, NYU

MITRE endpoint security testing offers a means for gaining insight into how commercial vendors approach threat detection, and not for ranking vendors as if often mistakenly performed.
Test results for OpenText's endpoint solution are interpretated in this context.

Contents

Introduction

The importance of endpoint security has heightened in recent years with the shift toward de-perimeterized zero-trust computing in enterprise. Security teams have thus had to build stronger controls into their employee laptop and desktop systems because organizational firewalls are no longer sufficient to prevent threats from malicious actors or compromised insiders. (The case can be made that firewalls never provided this protection sufficiently.)

To help drive the best types of cyber security protections in all types of enterprise controls, the MITRE organization[1] developed a useful framework of adversary tactics and techniques known as the MITRE ATT&CK Framework[2] (see Appendix A). The security community has responded positively to the emergence of this framework, and many commercial vendors find the list to be a useful guide in comparing and contrasting their product features with competitors.

MITRE has since begun supporting commercial vendor evaluations of products against specific attack campaigns built from the MITRE ATT&CK framework. These evaluations are intended to provide guidance to buyers, rather than as a comparison of which vendors are better than others, as might be found in analyst quadrants and waves[3]. The insights that come from proper interpretation of MITRE testing are therefore useful – and will be addressed in this report.

In particular, we focus on two points: First, we hope to explain the proper interpretation of MITRE endpoint security testing – namely, as offering broad insights into the capabilities of a given tool, versus surgically precise comparison. Second, we choose to highlight the recent results for one vendor, OpenText[4], to illustrate how proper interpretation can lead to accurate and meaningful conclusions for enterprise security buyers.

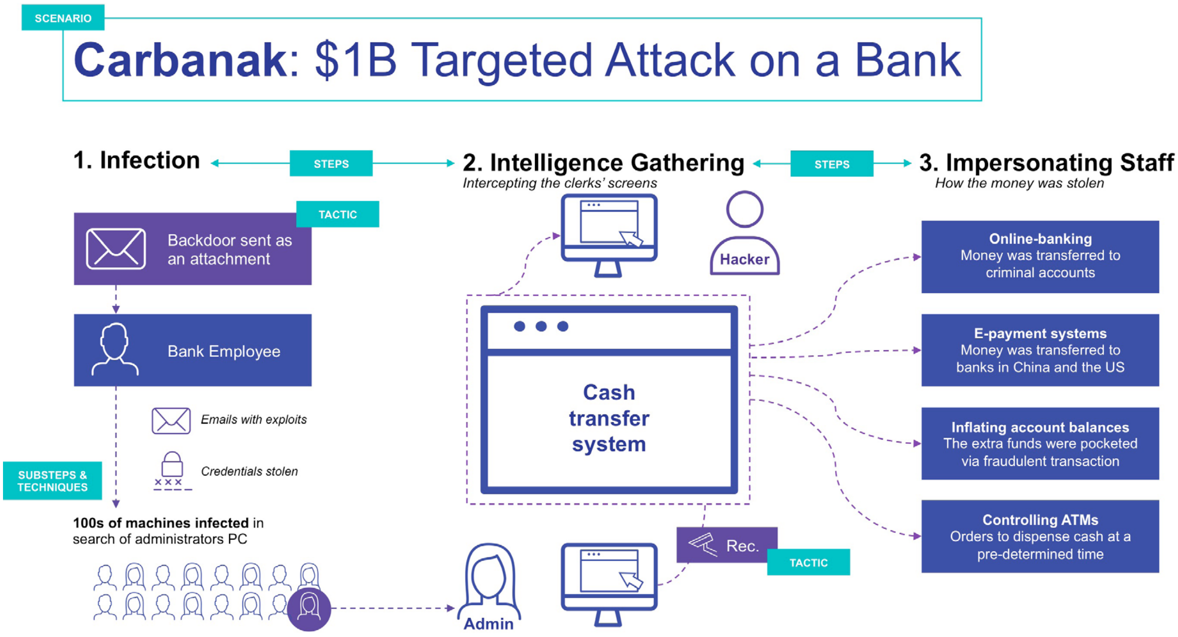

SCENARIO
Carbanak: $1B Targeted Attack on a Bank
1. Infection
STEPS
2. Intelligence Gathering
Intercepting the clerks' screens
STEPS
3. Impersonating Staff
How the money was stolen
TACTIC
Backdoor sent as an attachment
Bank Employee
Emails with exploits
Credentials stolen
SUBSTEPS & TECHNIQUES
100s of machines infected in search of administrators PC
Admin
Hacker
Cash transfer system
Rec.
TACTIC
Online-banking
Money was transferred to criminal accounts
E-payment systems
Money was transferred to banks in China and the US
Inflating account balances
The extra funds were pocketed via fraudulent transaction
Controlling ATMs
Orders to dispense cash at a pre-determined time

Figure 1: Carbanak Scenario

[1] MITRE is a federally funded research and development center (FFRDC). Information on MITRE is available at https://www.mitre.org/.

[2] Details on the MITRE ATT&CK Framework are available at https://attack.mitre.org/.

[3] Experts such as the analyst team at TAG Cyber warn that generalizations inherent in so-called vendors rankings are dangerously misleading since they cannot take into account the unique circumstances that are likely to be present in a typical environment.

[4] The OpenText management team commissioned this TAG Cyber report and was helpful during its development assisting the TAG Cyber analysts in their interpretation of test results for the recent MITRE ATT&CK testing round.

MITRE Endpoint Security Testing

Comparing the effectiveness of different cyber security products is a daunting effort – one that derives its roots from early Trusted Computer System Evaluation Criteria (TCSEC) evaluations[5]. The biggest operational challenge involves developing an accurate, meaningful, and reasonable test environment. Because so many different application, computing, networking, and other variables are involved, best-effort estimations will be required for any test environment.

That said, reasonable insights and benefit can be derived by running common benchmark tests on select software platforms to compare their respective performance, coverage, accuracy, or other targeted capabilities. The MITRE ATT&CK framework has been used to run a series of tests based on select advanced persistent threat (APT) campaigns to differentiate how various commercial endpoint solutions might detect (or not detect) the relevant indicators.

MITRE has been clear, however, that it does not intend for its endpoint security testing to be used as a marketing differentiator of commercial winners and losers in the marketplace based on test result performance. The following is a quote from their website:

Our real-world threat inspired methodologies are open and transparent. All results are publicly available and collaboratively produced with participants. There is no competitive analysis. We don't rank products against each other. And there is no "winner." Instead, we show how each vendor approaches threat detection through the language and structure of the MITRE ATT&CK® knowledge base and provide tools to allow the community to assess which product best fits their individual needs

This qualification of the MITRE testing is appropriate and should be reviewed carefully by any buyers who are intent on using published results to compare vendors. As the MITRE team explains, the results are used to show how "each vendor approaches threat detection."

Interpreting MITRE ATT&CK Results

As analysts, we believe that all enterprise buyers and security practitioners must develop a local approach to properly interpreting the results of MITRE ATT&CK and other independent testing[6]. To assist in this task, we propose two basic considerations that should influence interpretation of MITRE ATT&CK testing in a local environment. These considerations are offered to help balance the marketing messaging coming from security vendors after each round of testing.

Consideration 1: Buyers Should Recognize the Difference Between Security Test Environments and Live Networks

Take a moment to ponder this point: How easy would it be for a test environment to be established that can accurately simulate your own enterprise network with its unique applications, use-cases, network architecture, traffic types, and on and on? Obviously, if a proof-of-concept (POC) is performed, then the environment is better simulated, but when APT cases are executed for a short period of time in a test environment, the results can be skewed.[7]

[5] See https://en.wikipedia.org/wiki/Trusted_Computer_System_Evaluation_Criteria.

[6] Our points here are influenced by colleagues at Forrester who make comparable recommendations in an excellent blog post available here: https://go.forrester.com/blogs/winning-mitre-attck-losing-sight-of-customers/.

[7] An often-forgotten point in the interpretation of test results, including during POC tests, is that certain relevant events might occur at a frequency that does not overlap with the test period.

Recognizing this difference is therefore especially important for buyers with highly unique or one-off enterprise environments. For example, if a company intends to purchase a security solution for its Internet of Things (IoT) endpoints in a specialized operational technology (OT) environment filled with proprietary and non-standard systems and protocols, then any generic testing in MITRE's lab environment will not make any sense to extrapolate.

Even for enterprise buyers using a standard IT environment on familiar, off-the-shelf endpoints, the local environment is likely to have its own mix of tools, software, systems, agents, and even human behavior – all of which will have a direct impact on the success or failure of the endpoint tool in dealing with potential threats. This implies that interpretation of test results in too highly specific a manner will not be appropriate.

Consideration 2: Buyers Should View Security Test Results Broadly Versus Making Pinpoint Comparisons

This is the most important consideration for buyers. That is, when interpreting MITRE test results, enterprise teams should consider a broad assessment of the output data. This implies that if a vendor does particularly well – or if a vendor performs especially poorly – then this should serve as reasonable cause to investigate. This does not mean that a problem exists, but rather that the vendor should be encouraged to explain the disparity.

MITRE test results should not, therefore, be sufficient to select or de-select any vendor from a competitive assessment. Perhaps worse, any highly specific, pinpoint comparison of results (e.g., one vendor identified 3.45% more tactics than another) should be viewed as a useful data point, but never as a definitive security measurement. The key is to understand that endpoint test results require broad interpretation.

In fact, the MITRE guidance to use test reporting as a means to learn how vendors approach threat detection for the select APT campaigns is the best approach. Some vendors tout shift-left preventive solutions and will thus be good at detecting early indicators. Other vendors, however, tout their shift-right response capabilities and might include less intense technology to detect indicators in advance of an attack campaign.

Case Study: Interpreting OpenText Results

To illustrate appropriate interpretation of MITRE testing, we can review recent results for the OpenText EnCase Endpoint Security for the MITRE ATT&CK Round 3, which utilized the tactics of Carabanak and FIN7 as basis for the test execution. These actors were responsible for over a billion dollars in losses to financial service and hospitality groups – and while key members of both groups were arrested in 2018, the remaining groups are still active.

Broad Summary of OpenText Results

The OpenText EnCase Endpoint Security platform was tested over three days in October 2020 with results released on 4/20/2021. This third round in the MITRE test sequence addressed the familiar Carabanak/FIN7 campaign and included roughly 29 commercial participants. Testing included both step and tactic levels, which correspond to establishing an attack objective, and leveraging the corresponding ATT&CK tactic respectively.

For OpenText, as with all vendors, it is clear from the testing, that threats can be detected by the commercial tool in multiple ways. This should be clear when one reviews the process by which the Carabanak intruders worked. Their attack involved infection, harvesting intelligence, mimicking staff, and many other steps – each of which provided an opportunity for a commercial tool to detect or prevent some aspect of the campaign.

The results of tests are scored into numeric groupings, which in the case of this round of testing ranged from the low fifties to one hundred. As suggested above, TAG Cyber recommends more broad interpretation of results, perhaps grouping them into classes. A representative range might be 50 to 75, 75 to 90, and greater than 90, as reasonable equivalence classes to interpret the general test results. OpenText scored 77.16, which places them in the middle class.

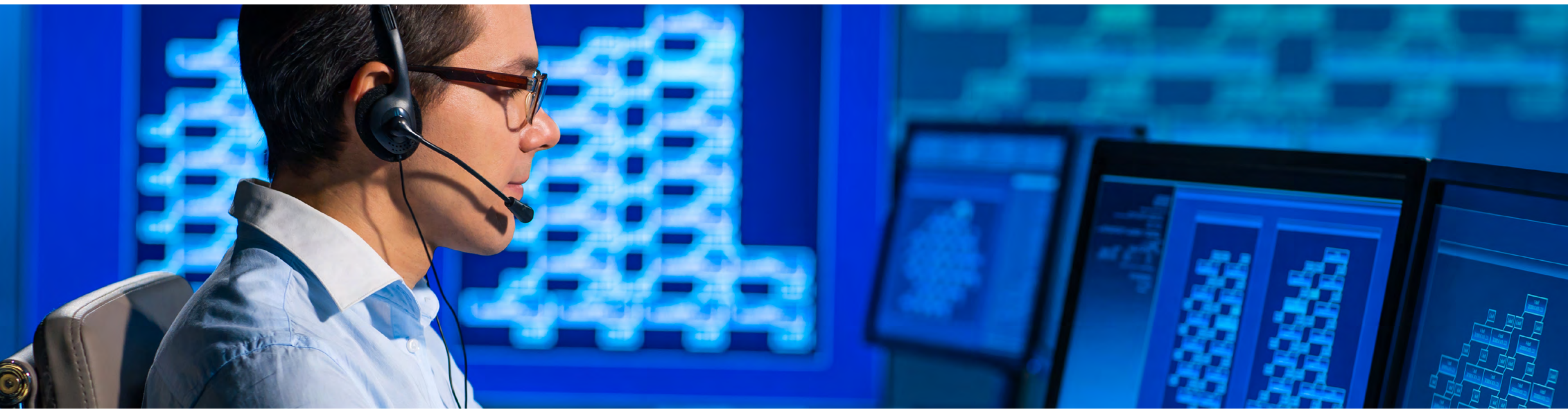
OpenText Test Result Interpretation

Two specific steps are recommended for interpreting and using test results: First, any truly outlier results should be discussed with the vendor – perhaps to include companies in the lowest class. This is not to say that they should be penalized, but rather that they should be given the opportunity to explain any test anomalies that might have contributed to the result. OpenText's score on this round would not warrant outlier investigation.

Second, the overall score should be complemented with more detailed analysis. For example, whereas OpenText scored 77.16 in the overall performance analysis, they did much better in the telemetry visibility with a 97.6 score. They also did well in the real time detection testing with a 99.6 score. Such comparison offers a slightly more in-depth view into security protections that might be particularly important in a given environment.

In contrast, the multiple configurations performed by OpenText during testing may be looked as a negative. In the real-world, however, new rules are often added on the fly and EnCase Endpoint Security offers considerable flexibility for this task using its rule builder tool. Such flexibility in adjusting a configuration could be viewed as a positive in this regard, so interpretations should take such issues into account, especially if they apply locally.

Additionally, the local environment might include a plethora of tools to address the general APT attack strategy which includes execution, exfiltration, persistence, lateral movement, and so on. If an enterprise security team is looking for a tool that complements its existing deployment, then focus on one or more aspects of this process might be more important than selecting a platform that offers general coverage.

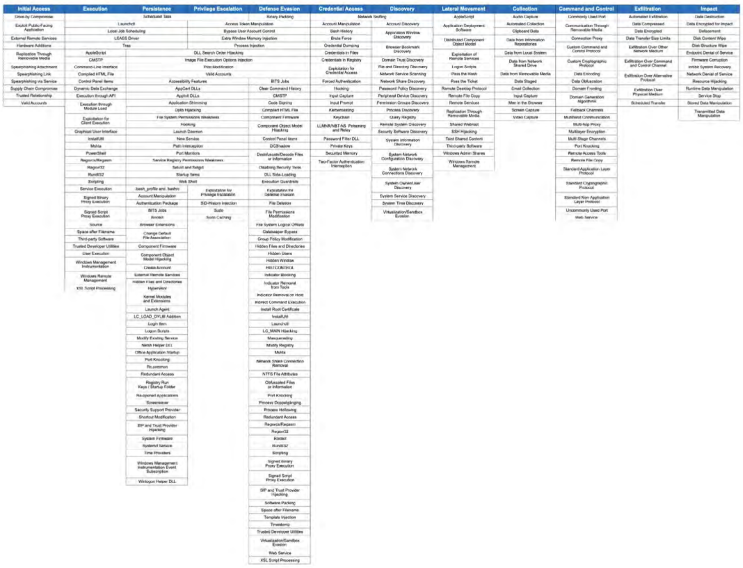


Action Plan

The TAG Cyber analyst team recommends that enterprise security teams commit to a local plan that properly interprets MITRE ATT&CK results as per the guidance offered above. This should include reinforcement of the guidelines during and after vendor marketing briefings that reference MITRE testing, or as part of any source selection process for endpoint security, digital forensics, or other enterprise cyber security tool or platform.

Appendix A: MITRE ATT&CK Framework

The MITRE ATT&CK Framework is a freely available matrix of tactics and techniques used by enterprise security teams, government agencies, and commercial cyber security vendors to classify cyber attacks and assess organizational risk by identifying gaps in cyber defenses and prioritizing their mitigation based on risk.



Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access
Drive-by Compromise	Scheduled Task	Scheduled Task	Scheduled Task	Binary Padding	Network Sniffing
Exploit Public-Facing Application	Launchctl	Launchctl	Access Token Manipulation	Access Token Manipulation	Account Manipulation
External Remote Services	Local Job Scheduling	Local Job Scheduling	Bypass User Account Control	Bypass User Account Control	Bash History
Hardware Additions	LSASS Driver	LSASS Driver	Extra Window Memory Injection	Extra Window Memory Injection	Brute Force
Replication Through Removable Media	Trap	Trap	Process Injection	Process Injection	Credential Dumping
Spearphishing Attachment	AppleScript	DLL Search Order Hijacking	DLL Search Order Hijacking	DLL Search Order Hijacking	Credentials in Files
Spearphishing Link	CMSTP	Image File Execution Options Injection	Image File Execution Options Injection	Image File Execution Options Injection	Credentials in Registry
Spearphishing via Service	Command-Line Interface	Plist Modification	Plist Modification	Plist Modification	Exploitation for Credential Access
Supply Chain Compromise	Compiled HTML File	Valid Accounts	Valid Accounts	Valid Accounts	Forced Authentication
Trusted Relationship	Control Panel Items	Accessibility Features	Accessibility Features	BITS Jobs	Hooking
Valid Accounts	Dynamic Data Exchange	AppCert DLLs	AppCert DLLs	Clear Command History	Input Capture
	Execution through API	AppInit DLLs	AppInit DLLs	CMSTP	Input Prompt
	Execution through Module Load	Application Shimming	Application Shimming	Code Signing	Kerberoasting
	Exploitation for Client Execution	Dylib Hijacking	Dylib Hijacking	Compiled HTML File	Keychain
	Graphical User Interface	File System Permissions Weakness	File System Permissions Weakness	Component Firmware	LLMNR/NBT-NS Poisoning and Relay
	InstallUtil	Hooking	Hooking	Component Object Model Hijacking	Password Filter DLL
	Mshta	Launch Daemon	Launch Daemon	Control Panel Items	Private Keys
	PowerShell	New Service	New Service	DCShadow	Securityd Memory
	Regsvcs/Regasm	Path Interception	Path Interception	Deobfuscate/Decode Files or Information	Two-Factor Authentication Interception
	Regsvr32	Port Monitors	Port Monitors	Disabling Security Tools	
	Rundll32	Service Registry Permissions Weakness	Service Registry Permissions Weakness	DLL Side-Loading	
	Scripting	Setuid and Setgid	Setuid and Setgid	Execution Guardrails	
	Service Execution	Startup Items	Startup Items	Exploitation for Defense Evasion	
	Signed Binary Proxy Execution	Web Shell	Web Shell	File Deletion	
	Signed Script Proxy Execution	.bash_profile and .bashrc	Exploitation for Privilege Escalation	File Permissions Modification	
	Source	Account Manipulation	SID-History Injection	File System Logical Offsets	
	Space after Filename	Authentication Package	Sudo	Gatekeeper Bypass	
	Third-party Software	BITS Jobs	Sudo Caching	Group Policy Modification	
	Trusted Developer Utilities	Bootkit		Hidden Files and Directories	
	User Execution	Browser Extensions		Hidden Users	
	Windows Management Instrumentation	Change Default File Association		Hidden Window	
	Windows Remote Management	Component Firmware		HISTCONTROL	
	XSL Script Processing	Component Object Model Hijacking		Indicator Blocking	
		Create Account		Indicator Removal from Tools	
		External Remote Services		Indicator Removal on Host	
		Hidden Files and Directories		Indirect Command Execution	
		Hypervisor		Install Root Certificate	
		Kernel Modules and Extensions		InstallUtil	
		Launch Agent		Launchctl	
		LC_LOAD_DYLIB Addition		LC_MAIN Hijacking	
		Login Item		Masquerading	
		Logon Scripts		Modify Registry	
		Modify Existing Service		Mshta	
		Netsh Helper DLL		Network Share Connection Removal	
		Office Application Startup		NTFS File Attributes	
		Port Knocking		Obfuscated Files or Information	
		Rc.common		Port Knocking	
		Redundant Access		Process Doppelganging	
		Registry Run Keys / Startup Folder		Process Hollowing	
		Re-opened Applications		Redundant Access	
		Screensaver		Regsvcs/Regasm	
		Security Support Provider		Regsvr32	
		Shortcut Modification		Rootkit	
		SIP and Trust Provider Hijacking		Rundll32	
		System Firmware		Scripting	
		Systemd Service		Signed Binary Proxy Execution	
		Time Providers		Signed Script Proxy Execution	
		Windows Management Instrumentation Event Subscription		SIP and Trust Provider Hijacking	
		Winlogon Helper DLL		Software Packing	
				Space after Filename	
				Template Injection	
				Timestomp	
				Trusted Developer Utilities	
				Virtualization/Sandbox Evasion	
				Web Service	
				XSL Script Processing	

Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Network Sniffing	AppleScript	Audio Capture	Commonly Used Port	Automated Exfiltration	Data Destruction
Account Discovery	Application Deployment Software	Automated Collection	Communication Through Removable Media	Data Compressed	Data Encrypted for Impact
Application Window Discovery	Distributed Component Object Model	Clipboard Data	Connection Proxy	Data Encrypted	Defacement
Browser Bookmark Discovery	Exploitation of Remote Services	Data from Information Repositories	Custom Command and Control Protocol	Data Transfer Size Limits	Disk Content Wipe
Domain Trust Discovery	Logon Scripts	Data from Local System	Custom Cryptographic Protocol	Exfiltration Over Other Network Medium	Disk Structure Wipe
File and Directory Discovery	Pass the Hash	Data from Network Shared Drive	Data Encoding	Exfiltration Over Command and Control Channel	Endpoint Denial of Service
Network Service Scanning	Pass the Ticket	Data from Removable Media	Data Obfuscation	Exfiltration Over Alternative Protocol	Firmware Corruption
Network Share Discovery	Remote Desktop Protocol	Data Staged	Domain Fronting	Exfiltration Over Physical Medium	Inhibit System Recovery
Password Policy Discovery	Remote File Copy	Email Collection	Domain Generation Algorithms	Scheduled Transfer	Network Denial of Service
Peripheral Device Discovery	Remote Services	Input Capture	Fallback Channels		Resource Hijacking
Permission Groups Discovery	Replication Through Removable Media	Man in the Browser	Multiband Communication		Runtime Data Manipulation
Process Discovery	Shared Webroot	Screen Capture	Multi-hop Proxy		Service Stop
Query Registry	SSH Hijacking	Video Capture	Multilayer Encryption		Stored Data Manipulation
Remote System Discovery	Taint Shared Content		Multi-Stage Channels		Transmitted Data Manipulation
Security Software Discovery	Third-party Software		Port Knocking		
System Information Discovery	Windows Admin Shares		Remote Access Tools		
System Network Configuration Discovery	Windows Remote Management		Remote File Copy		
System Network Connections Discovery			Standard Application Layer Protocol		
System Owner/User Discovery			Standard Cryptographic Protocol		
System Service Discovery			Standard Non Application Layer Protocol		
System Time Discovery			Uncommonly Used Port		
Virtualization/Sandbox Evasion			Web Service		

Figure 2. MITRE ATT&CK Matrix

About TAG Cyber

TAG Cyber is a trusted cyber security research analyst firm, providing unbiased industry insights and recommendations to security solution providers and Fortune 100 enterprises. Founded in 2016 by Dr. Edward Amoroso, former SVP/CSO of AT&T, the company bucks the trend of pay-for-play research by offering in-depth research, market analysis, consulting, and personalized content based on hundreds of engagements with clients and non-clients alike—all from a former practitioner perspective.

About OpenText

OpenText, The Information Company, enables organizations to gain insight through market leading information management solutions, on-premises or in the cloud. For more information about OpenText (NASDAQ: OTEX, TSX: OTEX) visit: opentext.com.

Connect with us:

- OpenText CEO Mark Barrenechea’s blog
- Twitter | LinkedIn

opentext.com/contact