The bridge to possible

Threat Modelling

It’s not just for developers

Tim (Wadhwa-)Brown
Security Research Lead, CX Technology & Transformation Group
March 2022

ATT&CK is a game changer and where it works, it can enable blue and red to co-exist and work effectively together

- However, what happens when it falls short and the threat intelligence and hypotheses doesn't exist?
- How do you build threat intelligence, threat models, threat simulations and threat hunt hypotheses from first principles?

Introduction

- TLDR
- # whoami
- # cat .plan

TLDR

- Not a data scientist
  - Could play one in a movie
  - No particular brief to think blue or red
- This is not a solved problem

whoami

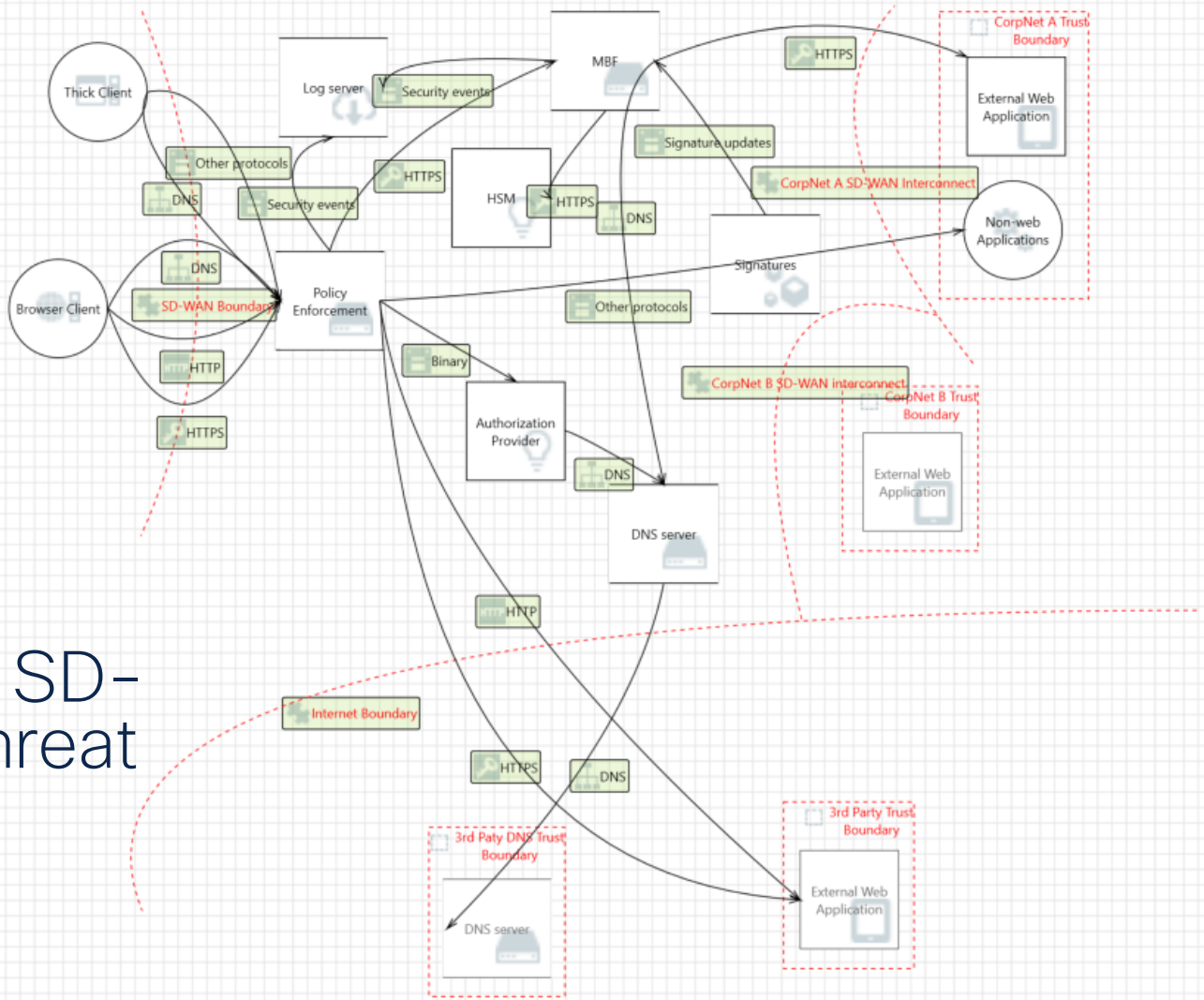
- Tim (Wadhwa-)Brown
  - Background in telecoms and financial services sectors
  - 15+ years at Portcullis (and now Cisco)
  - Security Research Lead, CX EMEAR Technology & Transformation Group
    - Ex-NCSC CHECK Team Leader (9 years)
    - CREST Registered Threat Intelligence Analyst
    - CREST Practitioner Intrusion Analyst
    - ISO 27001 LA
- >150 CVEs to my name
  - Covering Windows, Linux, AIX and Solaris platforms
    - Userland through to kernel
  - Most recent research: Where 2 Worlds Collide: Bringing Mimikatz et al to UNIX, Black Hat Europe 2018

cat .plan

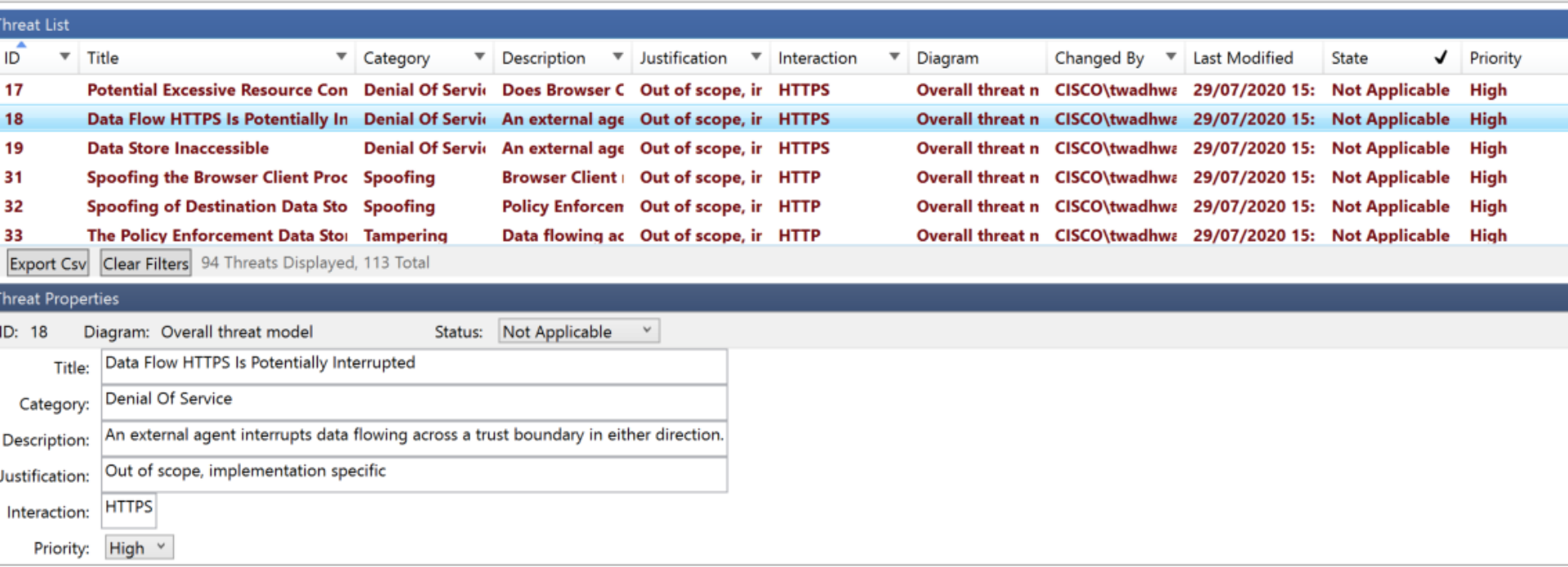
- Background
- Protecting a typical network
- Specific examples
  - Knowing your customer
  - Preparing for Black Hat
  - Managing (technical) debt
- Conclusions

Protecting a typical network

MEF88 SD-WAN threat model



Examining the SD-WAN threat model

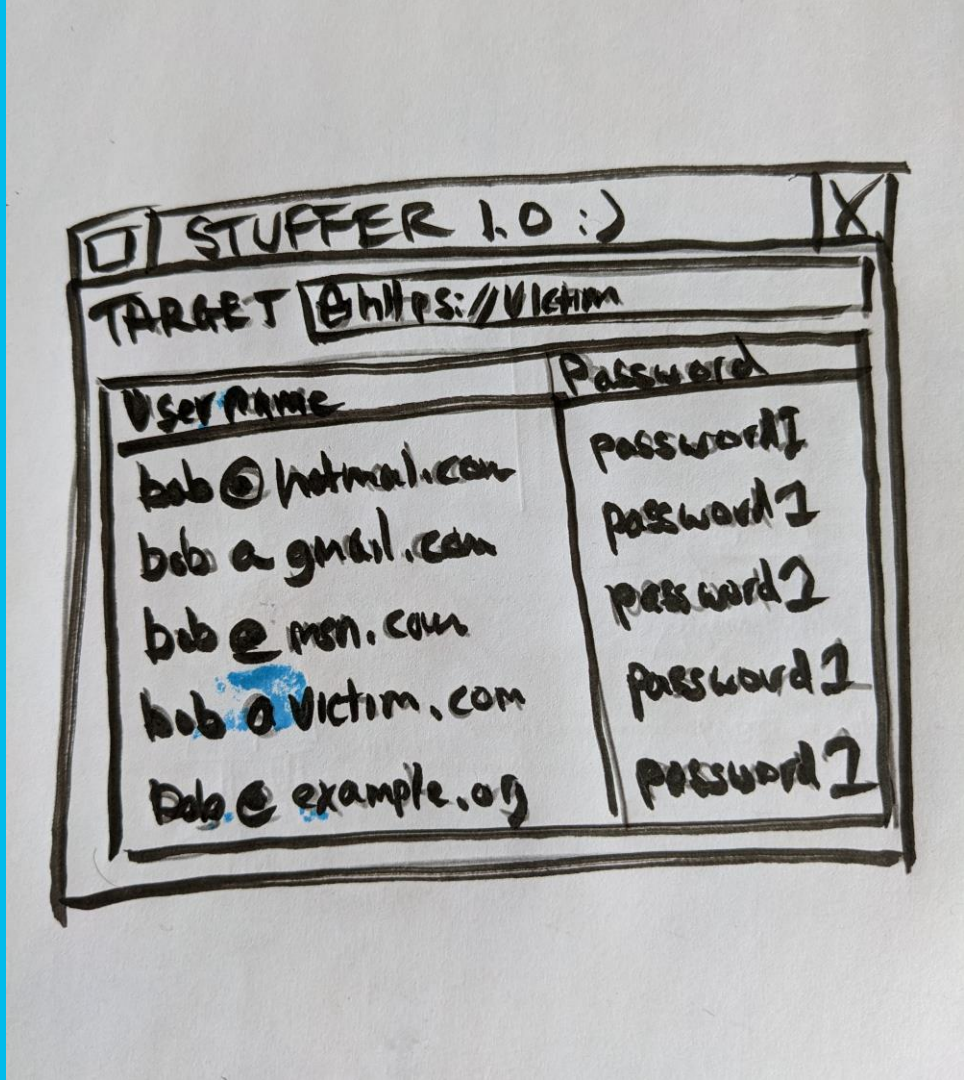


How would you validate it?

- Validating an SD-WAN implementation
  - Test against specification
- Focus
  - Implementation
  - Operation
  - Use case
- 94/113 threats in the current draft MEF88 threat model are deployment rather than design specific

Let’s look at some more specific examples

Knowing your customer



The incident

- Platform suffering from credential stuffing
  - T1078: Valid Accounts
    - Credential stuffing wasn't in my vocabulary
- Not sure what this is
- Tools/configs identified on a "hacking forum"

How did we defend against it?

- Realised it's using an old OpenSSL release
- Engineer block based on TLS ciphers
  - Compare ciphers?
  - Easier, look at size of selected suites
    - *SSL::cipher clientlist*
- Detection
  - DS0028: Logon Session
  - DS0002: User Account

Preparing for Black Hat

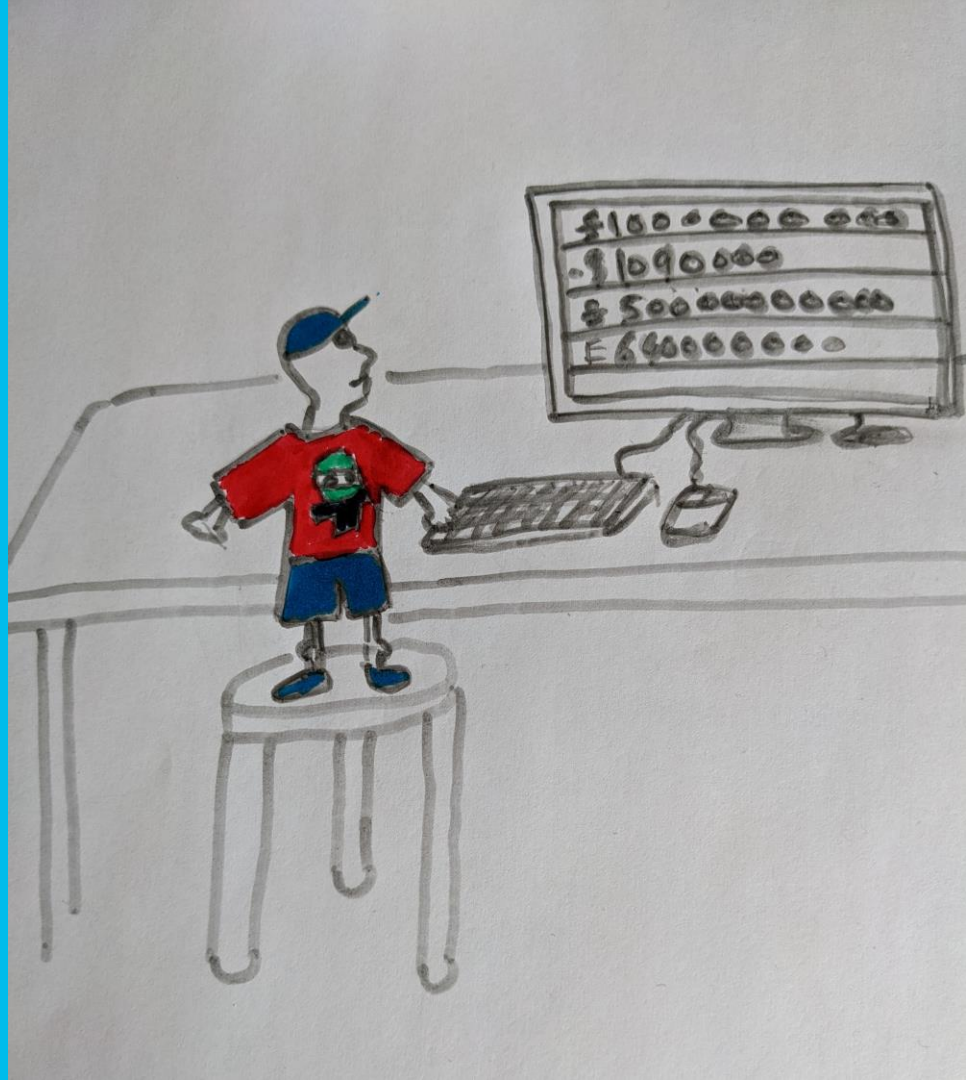
The research

- Where 2 worlds collide: Bringing Mimikatz et al to UNIX
  - T1003: OS Credential Dumping
  - T1558: Steal or Forge Kerberos Tickets
- I wrote Linikatz

How did would you defend against it?

- Auditd
  - Check the syscalls
  - Check file access
    - *-a always,exit -F dir=/var/lib/sss/db -F perm=rwx -k linikatz-sss*
  - Look for static numeric values to match on
    - Constants
    - Size parameters
    - *-a always,exit -F arch=b64 -S connect -F a2=0x2f -k linikatz-vas*
- Detection
  - DS0017: Command
  - DS0022: File
  - DS0009: Process

Managing (technical) debt



The vulnerability

- Insecure permissions on a retail banking application
  - T1005: Data from Local System
  - T1083: File and Directory Discovery
- Uncooperative vendor
- Legal moving slowly

How did I defend against it?

- ACLs and auditing
- Scripting the generation of an auditing policy and bespoke ACLs based on the output of `find'
- Detection
  - DS0017: Command
  - DS0009: Process

A dirty script

```
find /opt/component -name -perm -o+w | while read filename
do
        printf -- "-w %s -p r -k flag-%s-r\n" “${filename}” “$(printf ”%s” "${filename}" | tr \"/\" \"_\")”>>/etc/audit/rules.d/honeypot-component-dynamic.rules
        printf -- "-w %s -p w -k flag-%s-w\n" “${filename}” “$(printf ”%s” "${filename}" | tr \"/\" \"_\")”>>/etc/audit/rules.d/honeypot-component-dynamic.rules
        printf -- "-w %s -p w -k flag-%s-x\n" “${filename}” “$(printf ”%s” "${filename}" | tr \"/\" \"_\")”>>/etc/audit/rules.d/honeypot-component-dynamic.rules
        printf -- "-w %s -p a -k flag-%s-a\n" “${filename}” “$(printf "%s” "${filename}" | tr \"/\" \"_\")”>>/etc/audit/rules.d/honeypot-component-dynamic.rules
done
```

Conclusions

Putting it all together…

Technique	Detection	Technique	Detection	Technique	Detection
Initial Access		Credentialed Access		Discovery & Collection	
T1078: Valid Accounts	DS0028: Logon Session	T1003: OS Credential Dumping	DS0017: Command	T1083: File and Directory Discovery	DS0017: Command
	DS0002: User Account	T1558: Steal or Forge Kerberos Tickets	DS0022: File	T1005: Data from Local System	DS0009: Process
			DS0009: Process		

Final thoughts

- Security isn't all hashes, hostnames and IPs, at least not to begin with
- Be imaginative and encourage others to experiment
- Study your target
- Don't be afraid to break out the white board
- Think application-layer, not just network and transport
- ATT&CK gives blue and red a shared language, make full use of it
- Have fun!

Questions?

twadhwab@cisco.com

The bridge to possible

Bonus material

Useful links

- All of the threats – Intelligence, modelling, simulation and hunting through an ATT&CKers lens
- https://github.com/MEF-GIT/MEF-SDWAN-Application-Flow-Security-Threat-Model
- Where 2 worlds collide: Bringing Mimikatz et al to UNIX
- https://github.com/timb-machine/linux-malware

Useful links

- Microsoft Threat Modeling Tool
- https://github.com/cisco/joy/
- https://man7.org/linux/man-pages/man1/strace.1.html
- https://man7.org/linux/man-pages/man8/auditctl.8.html