Hacking Airports for Fun and Education (and better security monitoring, too!)

Meredith Kasper & Tom Kopchak

Hurricane Labs

Setting the scene

Who are we?

Tom Kopchak

Director of Technical Operations,
Technical Account Manager
@ Hurricane Labs.
CPTC competition director

Meredith Kasper

Director of Technical Services
@ Hurricane Labs. CPTC competition director, former CPTC competitor.

What is CPTC?

Offensive Security + Custom Environment + Business = CPTC

- **CPTC**: A premier international offensive security competition.
- **Challenge:** Conduct a penetration test of a fictitious company, and deliver the results to management.

Started @ RIT in 2015.
Still going strong 10 years later.

CPTC Themes

We create a new theme (target organization) every year.

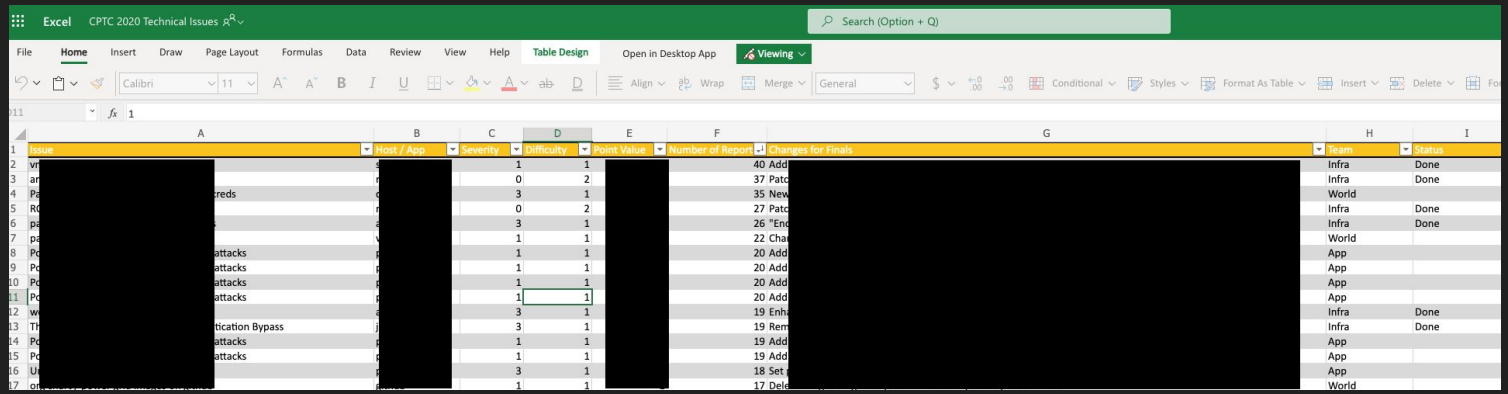
Themes of recent years:

- **2024 –** Social Media Company
- **2023 –** Airport
- **2022 –** Hotel
- **2021 –** Candy Manufacturing Co.
- **2020** – Public Utility
- **2019** – Financial Institution
- **2018** – Transportation App
- **2017** – Elections Provider


MINI-BANK
MINI-BANK


OVERHEAD TRAIN

Building the environment



New Year = New Environment

Typical Environment = 20-40 Hosts

- Business Hosts
  - Windows & Linux servers
  - Working AD environment
- Custom Applications

TONS of Vulnerabilities

- Typically 150+ known issues by the time we're finished

We Log **EVERYTHING**

- Our preferred tool of choice: Splunk
- Splunk agents (Universal Forwarders) deployed to all systems that support it in the environment.
- If there’s data to be collected, we try to do it.
- Most Windows + Linux inputs enabled, higher collection thresholds than “normal” for increased visibility.
- Custom inputs to support the competition.

Key Sourcetypes

- Splunk Stream (HTTP, DNS, TCP and UDP)
- WinEventLog:Security (Authentication and Change)
- Sysmon (Process Logging)
- WinNetMon (Traffic Logs by Process)
- Bash_history & powershell transcripts
- Office365 admin/ message trace
- AWS VPC flow
- Really stupid (™) file integrity monitoring
- ps and netstat

Robert A. Kalka Metropolitan Skyport (RAKMS)

Deep Dive – Simulating Airport Systems

For some reason, no one would let us use an actual airport for the competition…

Solution: Simulate the various airport systems

- People Mover, Baggage Claim, Ticketing, Radio, Multiple Airlines
- Alerts for team activity that impacted airport operations
- Leveraged automation to make our lives easier

Incident Investigation

When teams try to tell us it wasn’t them, we have the logs…

- WinNetMon
- Linux ps logging
- Stream TCP & sometimes HTTP
- Bash history

Host

All | vdi-kali01 | vdi-kali02 | vdi-kali03 | vdi-kali04 | vdi-kali05 | vdi-kali06

time	hostname	cmd
2022-11-19 17:56:11 EST	vdi-kali03	vim test
2022-11-19 17:43:01 EST	vdi-kali01	we a re too dumb
2022-11-19 17:42:57 EST	vdi-kali01	we cant even play the game
2022-11-19 17:42:51 EST	vdi-kali01	and please god five us creds
2022-11-19 17:42:47 EST	vdi-kali01	but i know someone is reading this
2022-11-19 17:42:43 EST	vdi-kali01	i don't know who is reading this\
2022-11-19 17:42:37 EST	vdi-kali01	i know someone is reading this
2022-11-19 17:42:29 EST	vdi-kali01	please send me creds
2022-11-19 17:42:27 EST	vdi-kali01	i suck at this pentesting thing
2022-11-19 17:42:22 EST	vdi-kali01	i dont know what is happening

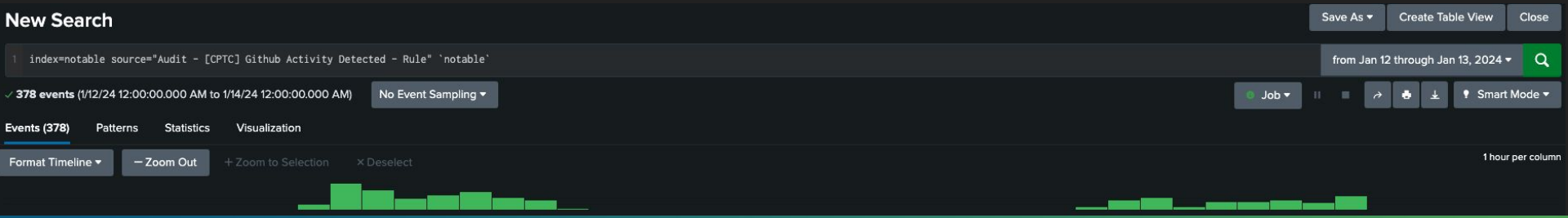
You have a lot of (repetitive) alerts, what do?

Automation!

Automating Incident Response

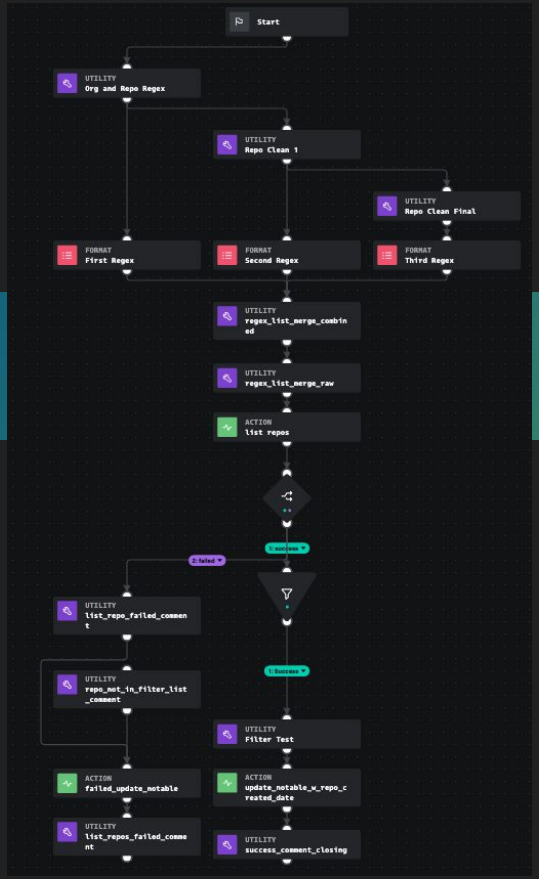
Github Activity Detected Alert

- Noisy alert with a good detection: **378 True-Positive Detections** over the course of the 2 day competition
- Very repetitive investigation process
- Great candidate for SOAR!

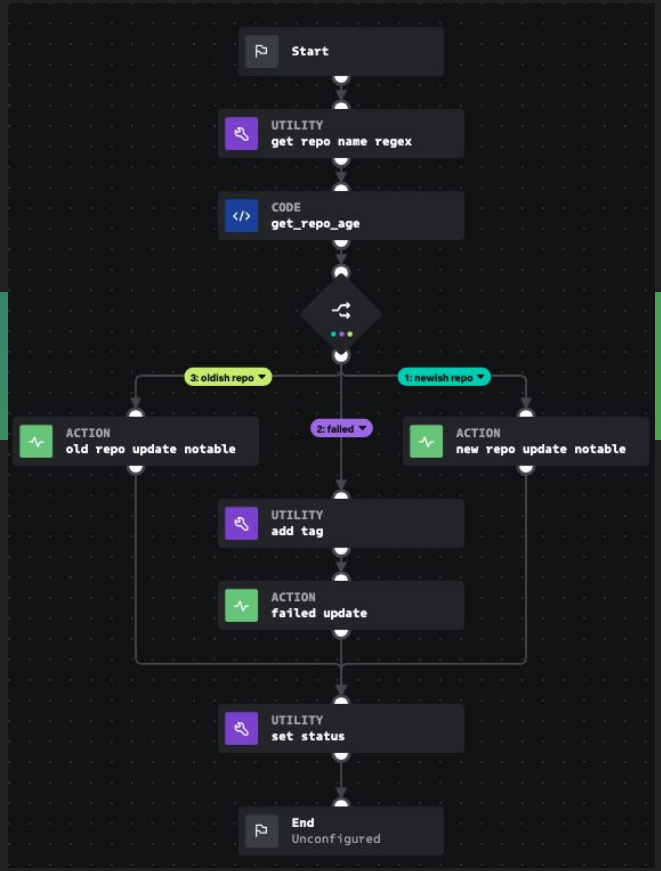


Github activity screenshots

Revision 1 + 2:



Revision 3:



Before automation

After Automation

Everything Else

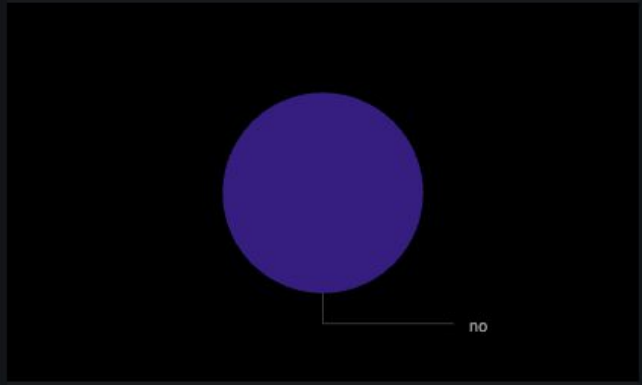
Totable Notable Count

290

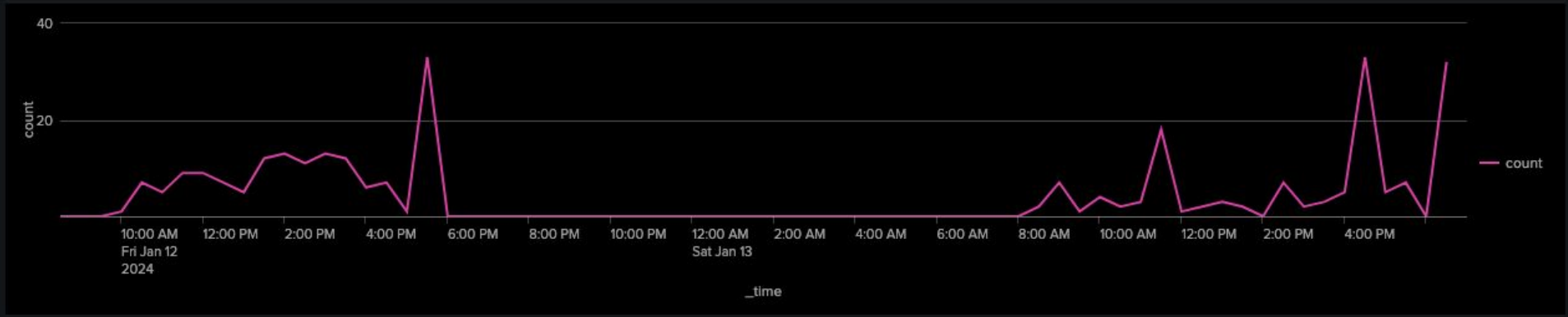
Average Time to Close

36.0 m

Handled By Automation



Notables Over Time



What’s the worst thing you’ve seen happen during a pentest?

Zerologin

What happens?

- Vulnerable Domain Controller with remote authentication enabled
- Shitty netlogon crypto allows for some nonsense to be guessable
- Running the exploit - you become the Domain Admin (the password is set to an empty string)
- Your entire domain ceases to function (once your DC reboots)
- Your users get angry
- Your company has a bad time

But nobody would *actually* run Zerologin on a production domain controller, right?

Time for security!

We used Splunk Enterprise Security (ES) for in-game alerting.

~35 active correlation searches (for now).

Detections included:

- Account Lockouts
- DNS tunneling
- Unexpected authentications/logons
- Scanning of public IPs
- Outbound transfers
- Password changes
- And more

Incident Response

- Account lockouts
- Deleting Accounts
- Changing Passwords
- Using Zerologin on a Domain Controller

Search

```
index=windows_security EventCode=4742 user=*$ NOT PasswordLastSet="-" src_user="ANONYMOUS LOGON"
| stats values(EventCode) as EventCode values(name) as name values(PasswordLastSet) as PasswordLastSet count by src_user user host
| `cptc_get_team_from_host(host)`
```

Bad Pentesting Behavior – Example #1

What would be a common mistake? Account lockouts
What logs are useful? Windows authentication logs
What searches would detect how a team cheated?

- Account Lockout Search
  - Custom correlation search
  - Alerts on any accounts in the environment being locked out
  - Same alert logic used for client environments
  - Goal: identify password spraying/cracking attacks that would impact the availability of the environment to end users
  - Large number of locked out accounts: very bad

- Unexpected Password Change
  - Custom correlation search
  - Alerts on any password changes in the environment
  - Assumption: no users changing passwords legitimately during the pentest
  - Alert logic can be adapted to client environments (RBA/risk alert, some accounts, eg, break glass shouldn't normally change)

Alerting & Competition Integrity – Example #2

How would one cheat? Exfil competition data, even entire hosts.

What logs are useful? Splunk stream logs.

What search would detected how we cheated?

- Large Outbound Transfer
  - Custom correlation search
  - Sum up bytes out by source and destination and alert whenever more than a gigabyte is sent to a public IP address
  - Search deployed in client environments

- DNS Tunneling
  - Custom correlation search
  - Use the built in truncate domain macro in Splunk to get the parent domain of any subdomains
  - Get a district count of queries and sum the length of the queries by domain and source
  - If the length was long and the number of different queries high we'd get an alert.
  - Search deployed in client environments

Bad Pentesting Behavior – Example #3

What would be a problem? Zerologin attack
What logs are useful? Windows authentication logs
What searches would detect how a team cheated?

- Zerologin activity search
  - Custom correlation search
  - Alerts on suspected Zerologin activity
  - Same alert logic used for client environments (rare to see at clients, less rare at CPTC)
  - Effective validation of search logic
- Result
  - 9 teams attempted Zerolgin during their assessment
  - 7 teams were unable to recover from running the exploit
  - Lessons learned: Understand the impact of tools/exploits before using them

Alerting & Competition Integrity – Example #4

Catching out-of-scope activity:

- Teams scanning public IPs from their pentest hosts
- Port scanning of public IPs
  - A lot of different ports against a destination from the same source in a short amount of time
- Directory brute force of public IPs
  - Large number of URLs against the same destination in a short amount of time
  - Teams often brute force directories on the public website (scope violation)
  - Protip: Pay attention to DNS!

Lessons Learned + Areas of Improvement

Automation is super useful

- Look for repetitive alerts and automate responses
- Save human investigation for deep dives

Collect more data, more efficiently

- Always more ways to get more data & visibility
- Deployment and scaling challenges

Utilize process logging

- WinNetMon/Sysmon for Linux are treasure troves of information
- Not always practical to collect but really helpful in an investigation

Lessons Learned + Areas of Improvement (continued)

More Dashboards!

- Notables are great when you're not watching and need an alert, but monitoring in real time is easier with a live dashboard
- Real time searches are almost useful here, but still cause resource issues

Tell the story

- Use the logging in Splunk to develop an attack path and timeline of activities (what the teams were doing)

How we use this to help our customers

- Many of the searches we develop have real applications outside of CPTC:
  - 30+ searches developed for CPTC have become part of our use case collection
  - Searches we improve using CPTC data get pushed to customers too
- Use CPTC dataset for testing searches
- PowerShell Transcript App (https://splunkbase.splunk.com/app/4984/) developed using CPTC data (in support of customer use cases)
- Validate alerting searches in an environment where “bad” activity is happening
- Threat hunting/investigation practice
- Risk-based alerting (RBA)
- Supports our continuous improvement initiatives

Hurricane Labs

Call to action

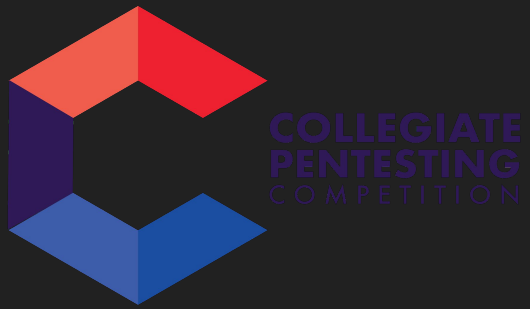
To the competitors watching:
If you cheated, let us know, so we can write better detections.

To Splunk enthusiasts:
Join our Splunk/monitoring team!

Other areas we need support:
Infrastructure building, application development, outreach & education, world/scenario building and in-character interactions, logistics & registration

The dataset is publicly available: http://mirror.rit.edu/cptc/
(Datasets for 2018-2023)

Get in touch!



Follow CPTC on Twitter:
@GlobalCPTC

Tom Kopchak
tom@hurricanelabs.com
tom@cp.tc
@tomkopchak

Meredith Kasper
mkasper@hurricanelabs.com
Meredith@cp.tc
@mistressven0m