WHEN THE HUNTER BECOMES THE HUNTED
HUNTING DOWN BOTNETS USING NETWORK TRAFFIC ANALYSIS
/ABOUT/ME

- Thomas Chopitea: Incident handler @CertSG
- Digital forensics & incident response (#DFIR), malware analysis, recent member of the Honeynet Project
- Twitter: @tomchop_
- Blog: http://tomchop.me/
- Also: we're hiring!
  https://cert.societegenerale.com/joinusnow!.html
• Common IR problems
• What is Malcom and how it leverages network traffic analysis and OSINT to solve them
• Malcom vs. botnets (demos, yay!)
• How you can use Malcom to deal with these problems
• How you can help Malcom grow stronger
I HAVE A LOT OF PROBLEMS
PROBLEM #1
KILL THE MALWARE
PROBLEM #1: KILL THE MALWARE

I need to:

- **Enumerate** domain names / IP addresses
- **Identify** resources (gates, dropzones, configs, etc.)
- **Gather** exchanged data (Configuration files? Stolen data?)
PROBLEM #1: KILL THE MALWARE

So I can:

- **Alert** the owners of stolen info & send **takedown** requests
- **Build** threat intelligence (so that I can refer to it later)
- **Start** incident remediation
NOT SURE IF CRITICAL

PROBLEM #2

OR JUST ANOTHER STRAIN OF ZEUS
PROBLEM #2: WTF IS THIS?

Sure, I could:

- Do an antivirus scan on it and get **Troj/Gen Suspicious**
- Reverse engineer it (3 samples a day? yeah right)
- Obvs: run it in a **sandbox** and do some behavioral analysis
- **x-ref** network artifacts against public blacklists
PROBLEM #3

I NEED TO DO IT **FAST**

('cause incidents keep popping up)
PROBLEM #3: GOGOGO

- **Don't want**: start Wireshark, text editor, snort, tcpflow, foremost, etc.
- **Want**: Drop my malware in a VM, and quickly know:
  - its behavior (which family does that?)
  - its peers (send the drones!)
- **Rinse & repeat**: Maybe get more intel, save the data to get results faster next time
YOU GUESSED IT
M ALCOM SOLVES MOST OF THESE PROBLEMS
MALCOM?

Malware communications analyzer

Available on GitHub: https://github.com/tomchop/malcom
WHAT IS MALCOM?
THREAT INTEL + MALWARE ANALYSIS

• **Gather intelligence** on network artifacts by syphoning the internet (and other sources)
• Match that intel with artifacts issued from ongoing malware analysis
• Identify targets, draw conclusions, act **FAST**!
WHAT IS MALCOM?

1 SLIDE ON ARCHITECTURE & FEATURES

- Python, scapy, mongodb (buzzword alert!), flask/bootstrap/d3js. Meant to be virtualised.
- Three modules:
  - Analytics & correlation engine
  - Feeding engine
  - Web interface
- Element types and tags
  - Each element has an analysis function
  - Each element is tagged according to its context
OPERATIONAL
THE ONLY THING I HAD IN MIND

• Quickly yield *actionnable intelligence*
• Other techniques may lead to more accurate / complete information, but I don't have *enough time*!
• Also, I wanted a visual tool
RECURRING TASK #1

'IS THIS [ARTIFACT] SOMETHING WE SHOULD WORRY ABOUT?'

- OSINT search for artifact: CBLs, blacklists, etc.
- Malcom gathers all badness in a single spot
- Easy to query, easy to hop from artifact to artifact
RECURRING TASK #2

'I HAVE RECEIVED THE FOLLOWING 0-DAY APT ATTACHMENT. IS IT MALICIOUS?'

- Yes/no/maybe/I don't know: throw it in a sandbox already!
  - If you do this often, you probably already have a sandbox with all the proper tools ready to use. That's ok.
- Your sandbox → Malcom → the Internetz
- Put Malcom in front of a Cuckoo?
RECURRING TASK #3

'WHO DO I HAVE TO SEND THIS ABUSE EMAIL TO?'

- Malcom will graph a host's network comms in **real-time**
  - You can also store them for later use, and replay them (PCAP)
  - You'll instantly know if you're dealing with one or many C&Cs, a P2P network, fast-flux architecture, or DGAs.
- And **cross-reference** them with stuff it already knows
  - You'll know if you (or someone) has run into the same artifacts
ENOUGH DEMO TIME.

(fingers crossed)
DEMONS

Show how Malcom graphs several types of communication

- C&C infrastructure
- Single and double fast-flux
- Domain flux (DGA)
- P2P botnets
C&C INFRASTRUCTURE

C&C == CnC == C2

- IRC → HTTP
  - Google / Facebook / Wikipedia ping
  - Fetch a configuration file from a central C&C server
  - Pony + Zeus Demo DEMO!

- Countermeasures
  - Quickly identify the malicious host. Strange domain name? Non-standard encryption? Weird file transfers? Strange x509 certificates?
  - Dig into database Demo on CERTSG's Malcom
'Flux' == 'change'

- Domain flux
  - Domain generation algorithms DEMO!

- Single and double fast-flux
  - Can be painful to manually sort everything out
  - Single FF: flux on the domains' A records DEMO!
  - Double FF: flux on domain A records and NS records DEMO!
PEER-TO-PEER

• Very resilient!
• No real single point of failure
• Taking these down usually involves cracking their protocol and hijacking the botnet
• In these cases, there's not much Malcom can do, besides:
  ■ Giving the initial peers' IP addresses
  ■ Pinpointing the "fallback" C&C used by the bot
• PHP.net pwnage dropping ZeroAccess DEMO!
SOUNDS COOL!

WHY SHOULD I TRY IT OUT?
EASY TO CUSTOMIZE

- **You choose** which sources Malcom will feed off
  - Internal / community / external data
- Easy to create a feed
  - Can read anything Python can!
- Elements have individual refresh rates
  - Important stuff gets refreshed more often
- Easy to add new element types (emails, IDs, specific hashes, etc.)
SHARING IS CARING

- **Share incident data!** You know how valuable it is 😊
- Open your Malcom instance to the world, let people feed off you
  - With another Malcom instance or any other tool
  - JSON feed for now, more formats soon
- API key enables access to specific tags (testing)
I still have 5 minutes left...

**ROADMAP - FEATURES**

- Yara rules in flows’ payload
  - Identify PEs, shellcode, nopsleds, etc.
- Application layer identified? → automatic payload extraction
- Compare communication patterns with known patterns (automatically)
  - Comms on non-standard ports, HELLO packets, etc.
  - Broids / suricata?
- “Pcap2bubbles”
  - Webservice to bubbleize your pcaps
  - Early early beta (not multiuser)
BACKEND

- Make it less “hack all the things”
  - Work on UI to add rules, feeds, etc.
- Use **redis** to synchronize processes
- Some performance improvements are in the scope
WANT TO HELP?

- Python / flask enthusiast
- Mongodb enthusiast
- Web / websockets / D3.js enthusiast (please!)

Poke around: https://github.com/tomchop/malcom
(the **dev branch** has waaay more features)