Welcome to

3. Mitigate and Monitor

Introduction to Incident Response Elective, KEA

Henrik Kramselund he/him han/ham xhek@kea.dk @kramse

Slides are available as PDF, kramse@Codeberg https://codeberg.org/kramse/
3-Mitigate-and-Monitor.tex in the repo security-courses

Goals for today

```
hlk@debian-lab-11:~/Downloads$ sudo vol -f cridex.vmem windows.pstree.PsTree
Volatility 3 Framework 2.4.1
Progress:  100.00  PDB scanning finished
PID PPID ImageFileName Offset(V) Threads Handles SessionId Wow64 CreateTime ExitTime

4 0 System 0x823c89c8 53 240 N/A False N/A N/A
* 368 4 smss.exe 0x822f1020 3 19 N/A False 2012-07-22 02:42:31.000000  N/A
** 584 368 csrss.exe 0x822a0598 9 326 0 False 2012-07-22 02:42:32.000000  N/A
** 608 368 winlogon.exe 0x82298700 23 519 0 False 2012-07-22 02:42:32.000000  N/A
*** 664 608 lsass.exe 0x81e2a3b8 24 330 0 False 2012-07-22 02:42:32.000000  N/A
*** 652 608 services.exe 0x81e2ab28 16 243 0 False 2012-07-22 02:42:32.000000  N/A
**** 1056 652 svchost.exe 0x821dfda0 5 60 0 False 2012-07-22 02:42:33.000000  N/A
**** 1220 652 svchost.exe 0x82295650 15 197 0 False 2012-07-22 02:42:35.000000  N/A
...
```

- Live Response and Memory analysis
- Get annoyed – memory analysis is tricky, and why we practice!
- Note how the book and processes **break down the problems into smaller bits, FOCUS**

Plan for today

- IDIR chapters F3EAD continued
- Getting data while observing the Order of Volatility
- Practice more

Exercise theme:

- Live Response
- Memory analysis with Volatility
- Checklists inspired by NIST SP800-61r2 – why, who, what, where, when, ...

Time schedule

- 1) Going over my presentation, – first 45min
- 2) Live Response, get to know your systems – 45 min
- Break 15min
- 3) Memory Analysis – 2x45min including 4)
- 4) Summary of course and today plus questions

Reading Summary

Intelligence-Driven Incident Response (IDIR) Scott Roberts. Rebekah Brown, ISBN: 9781098120689

> Once you have identified the threats that you are facing and investigated how those threats have accessed and moved through your network, it is time to remove the threats. This phase is known as Finish and involves not only eradicating the footholds that malicious actors have put in your network, but also working to remediate whatever enabled them to get access in the first place.

- Chapter 6: Finish

Reading Summary, browse *Computer Security Incident Handling Guide* NIST SP800-61r2

- Chapter 3: Handling an Incident

Live Response

One of the less appreciated but often effective analysis methods is live response. Live response is analysis of a potentially compromised system without taking it offline.

Most forensics analysis requires **turning the system offline**, losing system state information such as active processes. It also risks tipping off the attacker and is widely dis- ruptive to users as well.

Live response pulls the following information:

- Configuration information
- System state
- Important file and directory information
- Common persistence mechanisms
- Installed applications and versions

Live Response tools

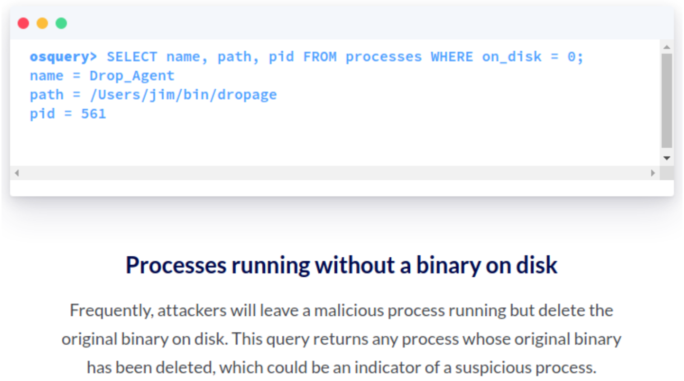
Book list a few tools:

- OSXCollector https://yelp.github.io/osxcollector/
- Kansa powershell https://github.com/davehull/Kansa
- Tools like these have been around for many years, some wrote to 3,5"floppies

Pause, let's discuss which ones would suit your environments, the class, your services, mine etc.

This is the common case, you are in the middle of an incident – which tools do you know, which are available, do you have licenses for commercial ones, etc.

Other tools that might help with data



```
osquery> SELECT name, path, pid FROM processes WHERE on_disk = 0;
name = Drop_Agent
path = /Users/jim/bin/dropage
pid = 561
```

Processes running without a binary on disk

Frequently, attackers will leave a malicious process running but delete the original binary on disk. This query returns any process whose original binary has been deleted, which could be an indicator of a suspicious process.

Source: https://www.osquery.io/

Many other tools can help identify problems

- osquery https://www.osquery.io/ if installed collect end point information
- Lynis https://cisofy.com/lynis/ audit your system

Exercise

Now lets do the exercise

⚠ Live Response 45 min

which is number **18** in the exercise PDF.

Memory Analysis

> Similar to live response, memory analysis focuses on collecting volatile system state in memory. Given that every process on a system requires memory to run, this tech- nique provides an excellent vantage point to gather information, especially from tools that attempt to run stealthily with limited system footprint.

Source: Source: *Intelligence-Driven Incident Response* (IDIR)

- FireEye's Redline memory analysis tool (created by Mandiant)

Volatility memory-analysis framework

> Volatility is a Python-based, open source memory-analysis framework. Volatility does not gather memory itself the way Redline does. Instead, it reads the memory formats from a wide variety of collection tools that run on a wide variety of operating systems. What Volatility provides is a framework and set of scripts for analyzing memory; detecting malware running in memory, extracting cryptographic keys—in fact any- thing you can find a plug-in to do.

Source: Source: *Intelligence-Driven Incident Response* (IDIR)

- https://www.volatilityfoundation.org/
- Again, more tools exist for Memory Analysis
- Often virtualisation can provide memory dumps, pause VM, take snapshot and download file – easy access!

Disk Analysis – not today

> Traditional disk forensics typically involves using specialized tools to extract filesys- tem information from the raw bits and bytes on a hard drive. The information on a hard drive is unintelligible at first glance. It contains endlessly nested structures at the hardware, filesystem, operating system, and data-format level, similar to the OSI model. Peeling through these layers is a process called file carving.

Source: Source: *Intelligence-Driven Incident Response* (IDIR)

- Important subject: We will do this another day

Exercise

Now lets do the exercise

⚠ Volatility framework 45 min

which is number **19** in the exercise PDF.

IDIR Chapter 6: Finish

> Finish involves **more than removing malware** from a system, which is why we spend so much time in the Find and Fix stages. To properly finish an attacker's activity, it is critical to **understand how that threat actor operates** and to remove not just malware or artifacts left behind by an attack, but **also communications channels, footholds, redundant access, and any other aspects of an attack** that we uncovered in the Fix phase. Properly finishing an adversary requires a **deep understanding of the attacker, their motives, and their actions**, which will allow you to act with confidence as you secure the systems and regain control of your network.

Source: Source: *Intelligence-Driven Incident Response* (IDIR)

- This is critical to success
- When handling real incidents, make sure to work in teams and schedule time off!

Stages of Finish

> The Finish phase has three stages: **mitigate, remediate, and rearchitect**. These stages acknowledge that you **can't do everything at once**. Even after a comprehensive investigation, some tactical response actions can take place quickly, but many **strategic response actions, such as rearchitecting, will take longer**. We will discuss the three phases next.

Source: Source: *Intelligence-Driven Incident Response* (IDIR)

- Mitigate
- Remediate
- Rearchitect

Mitigate

> Mitigation is the process of taking **temporary steps** to keep an intrusion from **getting worse** while **longer-term corrections** are taken. Ideally, mitigation should take place **quickly and in a coordinated fashion** to avoid giving the adversary a chance to react before you have cut off their access. Mitigation takes place at several phases of the **kill chain**, including delivery, command and control, and actions on target.

Source: Source: *Intelligence-Driven Incident Response* (IDIR)

- Stop the delivery by blocking known entry ways
- Patch remaining systems – to avoid new infections
- Block known bad IP addresses

Remediate

> Remediation is the process of **removing all adversary capabilities and invalidating any compromised resources** so that they can no longer be used by the adversary to conduct operations. **Remediation** generally focuses on a different set of kill-chain phases than mitigation does, most notably **exploitation, installation, and actions over target**, which we will break down in this section.

Source: Source: *Intelligence-Driven Incident Response* (IDIR)

- Clean and Patch systems

Rearchitect

> One of the most effective uses of **intelligence-driven incident-response data** is an advanced form of remediation: the incident-response team looks at **past incident trends, identifies common patterns, and works to mitigate these at a strategic level**. These mitigations are generally not small changes, and may range from small things like tweaks to system configurations or additional user training, to massive shifts in tooling such as the development of a new security tools or even complete network rearchitecture.

Source: Source: *Intelligence-Driven Incident Response* (IDIR)

- Why did the attacks succeed in the first place, can we change the environment

Taking action

> Active defense is frequently equated with the idea of **hack back**, or attempting to attack a malicious actor directly. Although this qualifies as one aspect of active defense, five other useful pieces of active defense are **far more common**. This mix-up is based on a fundamental misunderstanding of the purpose of active defense.

I do NOT recommend hacking back! Please do inform the administrators responsible for computing resources that attacks you!

Active Defense (5D)

- Deny
- Disrupt
- Degrade
- Deceive
- Destroy

Deny

> The idea of denying an adversary is so straightforward and common that most organizations wouldn't even imagine it's a type of active defense; most are actively doing deny-style actions. If we go by our traditional definition of disrupting the adversary's tempo, though, this is a perfect example. **Denying can be simple, such as implementing a new firewall rule to block an adversary's command and control**, applying a system patch for a vulnerability, or shutting down access for a compromised email account. The key to denial is **preemptively excluding** a capability or infrastructure from the malicious actor.

- Most likely what you want to do
- Use intelligence feeds to know what to block
- Systems like CrowdSec can also be used https://www.crowdsec.net/
- The log4j case is an example where a list of systems performing scanning was shared in real time by CrowdSec https://www.crowdsec.net/blog/detect-block-log4j-exploitation-attempts

Disrupt

> If the deny action preemptively excludes a capability or infrastructure from the malicious actor, then **disrupt actively excludes a resource from the malicious actor**. In most cases, disruption requires active observation of an adversary in order to know when they're active so that they can be disrupted in real time. This could mean **cutting off a command-and-control channel while it's being used or interrupting the exfiltration of a large archive file**.

- There are systems to actively monitor the network for exfiltration, NDR, XDR
  https://en.wikipedia.org/wiki/Network_detection_and_response
  https://en.wikipedia.org/wiki/Extended_detection_and_response

Degrade

> Closely related to disrupting and denying an adversary, **degrade focuses on marginal reduction** of an adversary's resources while they're actively being used. An easily understandable **example is throttling an adversary's bandwidth during exfiltration**, causing a large file to upload over an extremely slow time frame. This degradation of access attempts to frustrate adversaries, ideally driving them to attempt to access the data in a different way and expose additional infrastructure, tools, or TTPs.

- If you mostly receive data over some connection, could you implement throttling before attacks start?

Active defense conclusion, Implementing multiple tools before they are needed is recommended

Deceive

> Easily the most advanced of available techniques, the deceive active defense action is based on the counter-intelligence concept of **deliberately feeding adversaries false information** with the hopes they'll treat it as truth and make decisions based on it. This ranges from planting false documents with incorrect values to hosting honeypot systems or even networks.

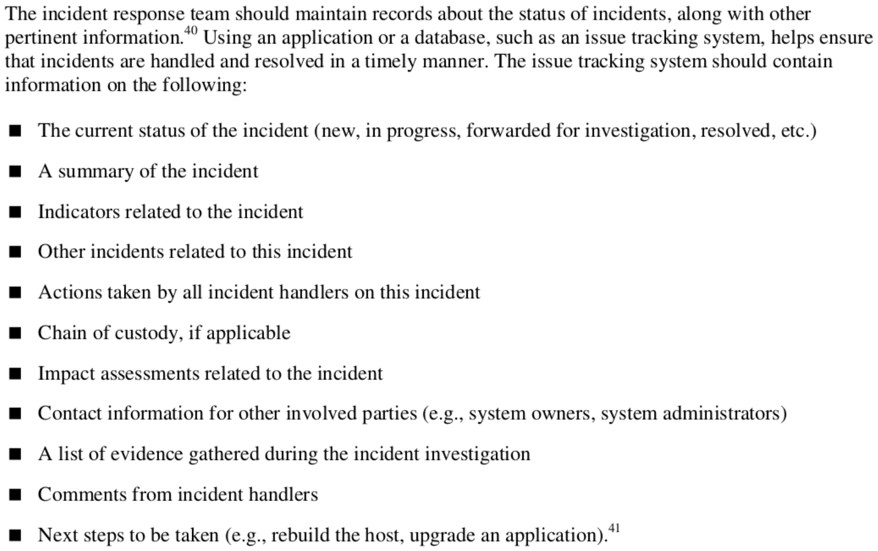
- The Cuckoos Egg book has a few examples where the defender tried getting the attacker to spend more time and resources

Destroy

> Destroy actions do **actual harm, whether kinetic or virtual, to an adversary's tools, infrastructure, or operators**. In most cases, this is the purview of law enforcement, intelligence, or military operators that have the legal authority to commit such acts (in the US, these are Title 10 and Title 50 organizations). For a **commercial or private organization to do so is not only generally accepted to be illegal but also dangerous.**

- See the LockBit example, currently in the news – search and you will quickly find it

Checklists



The incident response team should maintain records about the status of incidents, along with other pertinent information.[40] Using an application or a database, such as an issue tracking system, helps ensure that incidents are handled and resolved in a timely manner. The issue tracking system should contain information on the following:

■ The current status of the incident (new, in progress, forwarded for investigation, resolved, etc.)

■ A summary of the incident

■ Indicators related to the incident

■ Other incidents related to this incident

■ Actions taken by all incident handlers on this incident

■ Chain of custody, if applicable

■ Impact assessments related to the incident

■ Contact information for other involved parties (e.g., system owners, system administrators)

■ A list of evidence gathered during the incident investigation

■ Comments from incident handlers

■ Next steps to be taken (e.g., rebuild the host, upgrade an application).[41]

- Checklists inspired by NIST SP800-61r2 – why, who, what, where, when, ...

Summary Where are we in the course!

Questions to ask:

- What is a tool? Are programs the only tools, can a checklist become a tool
- Which things are generic, which things need to be decided in a specific environment

For Next Time

Think about the subjects from this time, write down questions

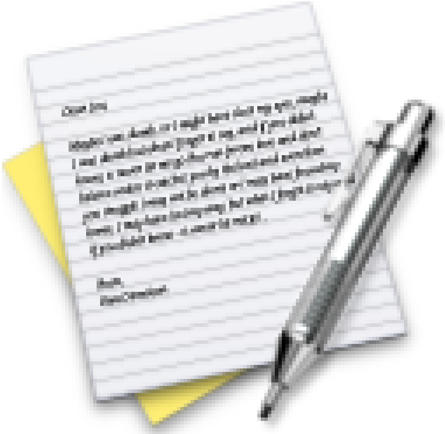
Check the plan for chapters to read in the books

Visit web sites and download papers if needed

Retry the exercises to get more confident using the tools

Buy the books! Create your VMs

Exercise



Now lets do the exercise

ℹ Use Ansible to install programs 10-60min

which is number **20** in the exercise PDF.