

FalconForce

SENTRY DETECT

— Enhancing threat detection with bespoke detection content

Olaf Hartong

Detection Engineer and Security Researcher

- Purple teaming, Threat hunting
- Microsoft Security MVP


Former documentary photographer

Father of 2 boys

“I like **warm hugs**”

 @olafhartong

 olafhartong.nl

 github.com/olafhartong

 Olaf falconforce.nl

 olafhartong.nl / falconforce.nl









Henri Hambartsumyan

Offensive defender

- Red teaming & detection engineering
- Uses offensive skills to improve detection

Loves evading defensive software
Father of 2 boys

 @Oxffhh
 github.com/Oxffhh
 henri@falconforce.nl
 falconforce.nl

About FalconForce

- Founded in 2020 by a group of highly experienced cyber security professionals
- ♡ Vision to make cyber security more purple
- Applied research is at the core of everything we do
- Focus on highly technical project in corporates with a high security maturity

Goal of this webinar

1

Explore why custom detection is needed

2

Understand what makes detection engineering so hard

3

Learn how to approach detection engineering

4

Discover managed detection engineering



Why do we need custom detection content?



Do we need more detection content?



Built-in detections are sufficient against low-skill attackers



Built-in detections need to work for everyone, worldwide!



Attackers test their detections against built-in content



Custom detections are built and tuned for your environment



Why does my security product have blindspots?

Limited research effort

Security vendors often focus on detecting public tradecraft, rarely on new or modified techniques

Limited tuning possibilities

Built-in tuning tools only support basic tuning options. Complex, correlation based tuning not supported.

Cloud complexity

Requires understanding on how YOU use your cloud

Lacking correlation

Detection relies on single-vendor data with minimal cross-product correlation

Needs to work for everyone

Detection logic lacks tailoring, aiming for global reliability

Lacks detailed understanding

Detection needs deep environmental and infrastructure insights



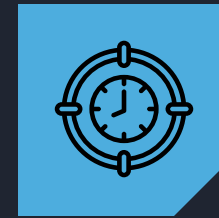
Why is detection engineering hard?



Hands-on
offensive
skills



Realistic
lab
environment



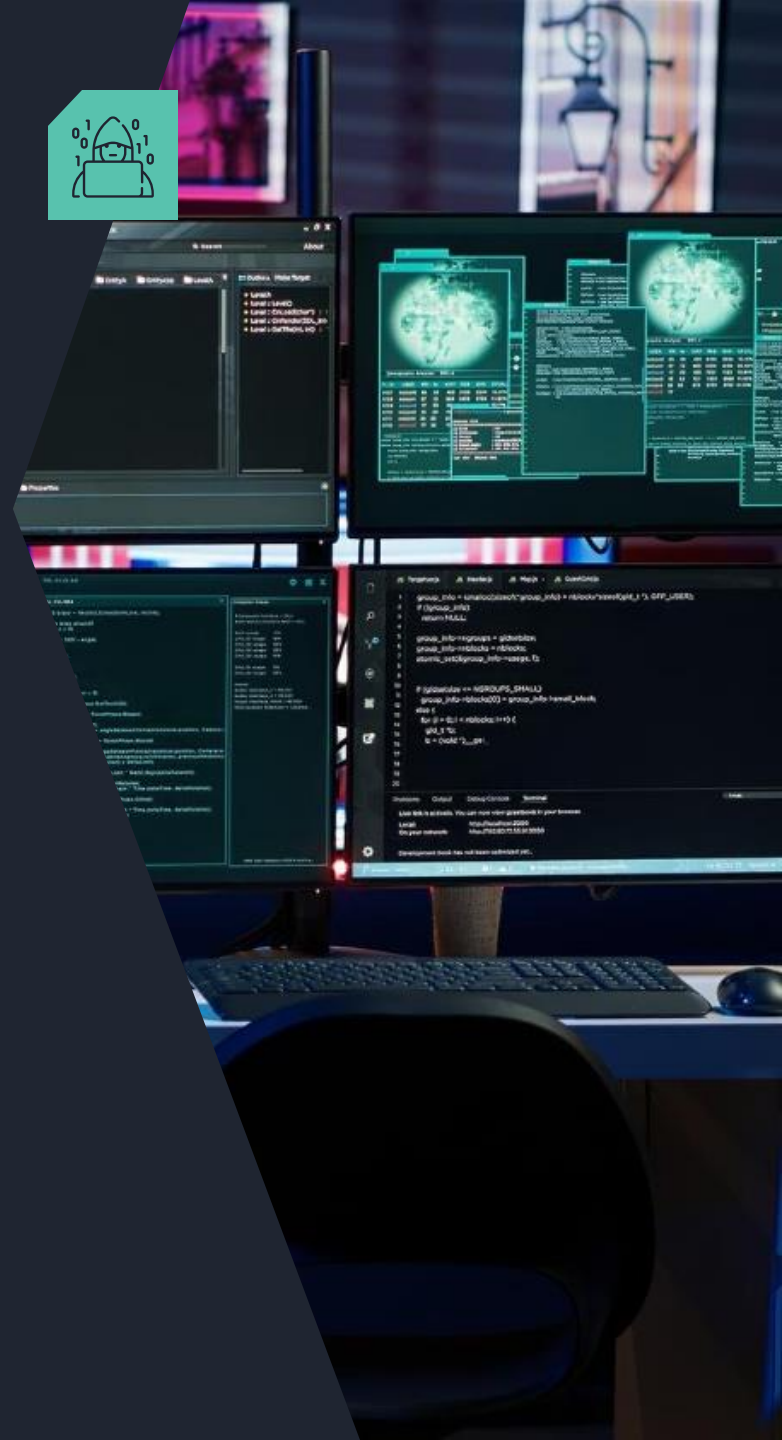
Time
and
focus



Hands-on offensive skills



- Executing attacks for telemetry requires strong offensive security skills
- They must also possess blue team skills to create production-ready detections
- Splitting roles is possible but sacrifices efficiency and finesse



Realistic lab environment



- Simulating attack techniques requires a realistic lab, as most TTPs can't run in production
- Maintaining a realistic lab for detection engineers demands significant effort
- Organizations vary in environments, including Linux, Mac, AWS, and GCP
- Labs need regular cleanup due to disruptions from attack techniques and untrusted tools

- AD domain
- Workstations
- Servers hosting
- Security tooling
- Sentinel/SIEM
- Intune setup (+enrollment)
- Entra ID setup
- Azure resources for testing



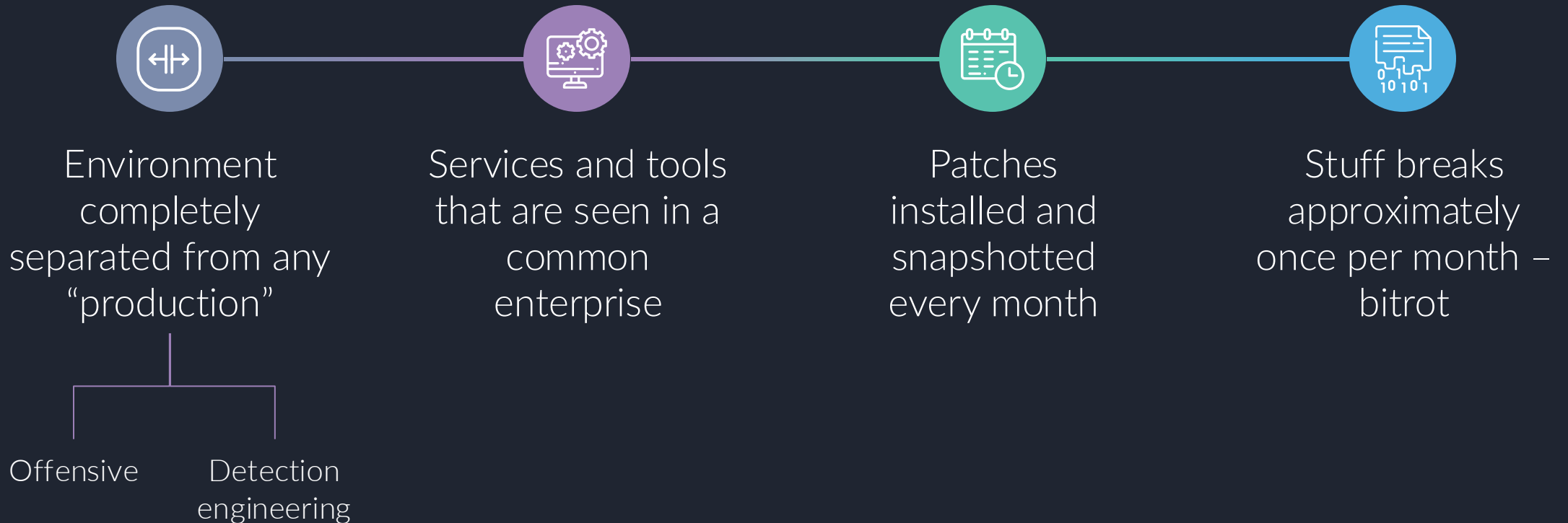
Time and focus



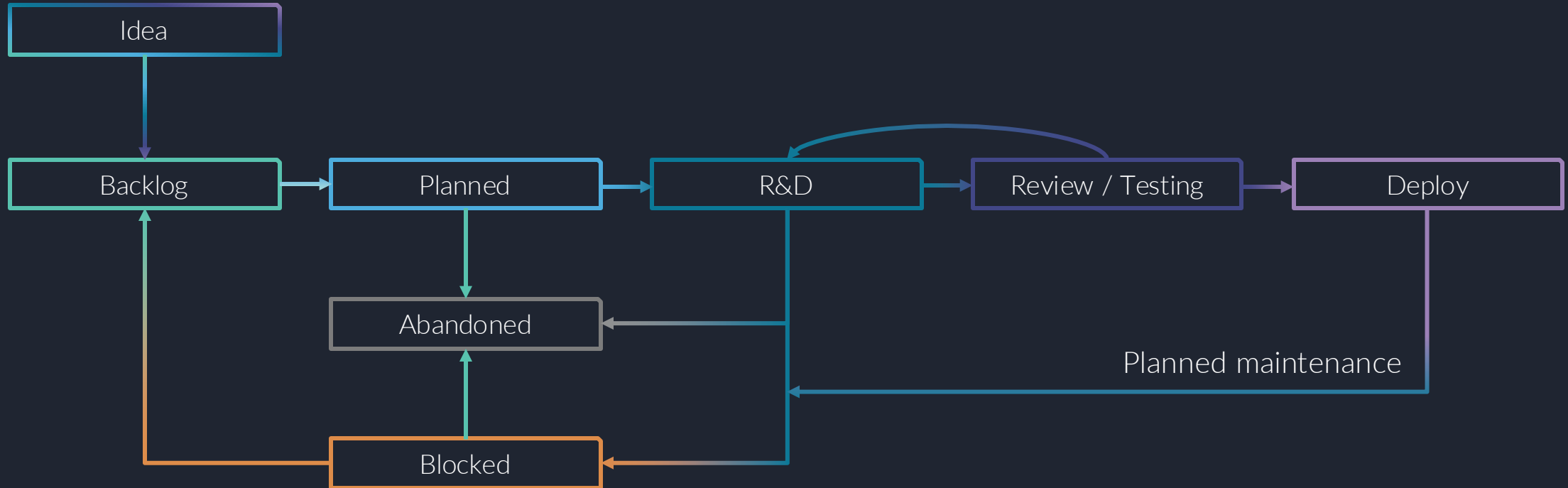
- Proper detection engineering requires 1-2 full-time engineers
- Creating a production-ready detection takes 1-3 days, including engineering, testing, and documentation
- Detection engineering needs dedicated time and focus, not just during downtime



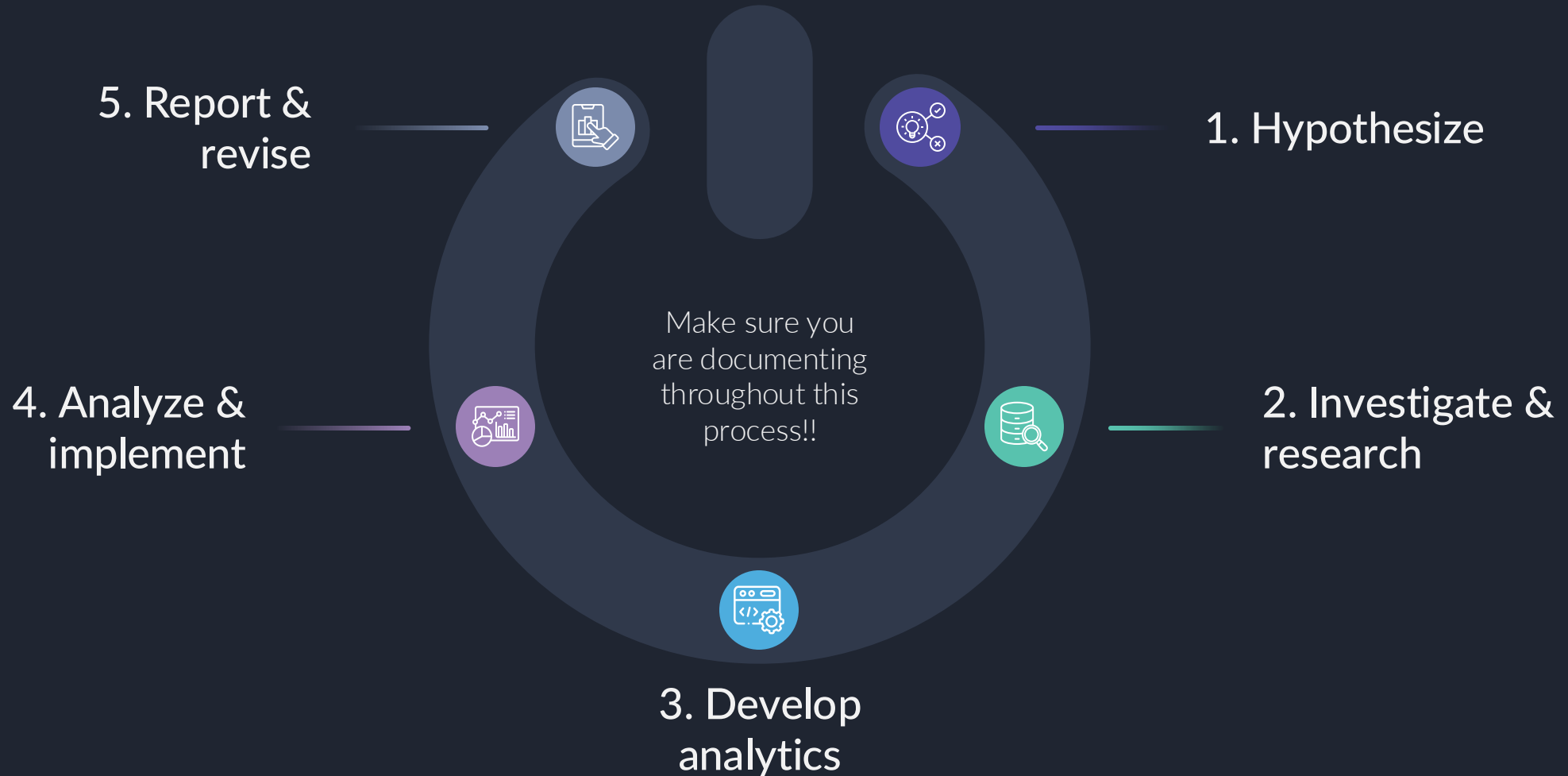
How we approach detection engineering



Our (simplified) process flow



FalconForce Detection engineering cycle



Automation

○ People make mistakes – you can use automation to catch mistakes as early as possible

○ Git is the single source of truth. Everything else is derived from the version on git.

Automation use cases

- ❑ Syntax validation of the produced output by detection engineer
- ❑ Semantic validation of “standards” (e.g., naming convention, time conventions, function usage, etc.)
- ❑ Automatic deployment to your security tools
- ❑ Generating (and export) nice looking documentation files (Confluence/Markdown/ Word/...)
- ❑ Generating MITRE ATTACK mappings for reporting
- ❑ Performing generic maintenance tasks (e.g., are the URLs in my documentation still active?)

More on automation in the next webinar!



Testing




Recent examples

- 1 out of 14 DCs stopped sending security event logs
- Microsoft has accidentally excluded XLL loads from the DeviceImageLoadEvents table.
- GraphAPI logs started to behave erratic, only a small percentage of logs makes it into Sentinel – rest is lost.
- The log format has changed in Azure, the same action now triggers slightly different looking telemetry.

- Detection break silently all the time: you **need** to test detections!
- World around us has changed (intentionally)
- Something in the IT stack is broken (accidentally)



Minimal Tools


-  Lab as discussed before
-  Clean VMs to work from
-  A solid editor for working with YAML files (VSCode)





Why Managed Detection Engineering?




What we offer

✓  High-fidelity MS stack detections, detecting attacker behavior instead of tools

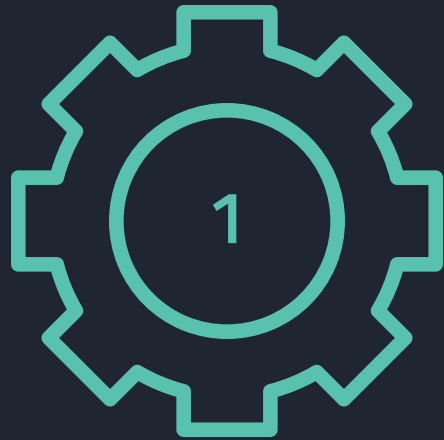
✓  Built and tested by purple teamers that have hands-on offensive security expertise

✓  Implemented and maintained for you

✓  Low number of false-positives



Sentry Detect – Managed Detection Engineering



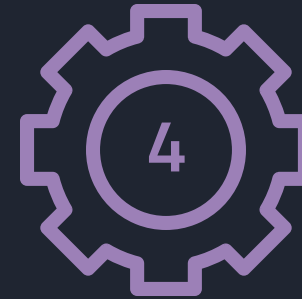
You choose detections from our portal - we give recommendations based on your environment



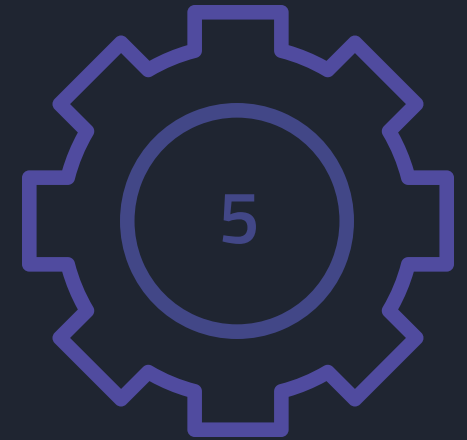
We validate if the detections are feasible in your environment



We tune detections and stage detections & documentation for deployment



We discuss tuning results together



We (or you) push from staging to production



Sentry Detect – collaboration models



Booster

- 🎯 Go fast
- € One-time fee

✓ **High fidelity custom detections**



Pro

- 🎯 Regular new detections
- € Subscription

✓ High fidelity custom detections

✓ **Detection-as-code pipelines**

✓ **Maintenance**



Enterprise

- 🎯 Deploy and tune together
- € Subscription

✓ High fidelity custom detections

✓ Detection-as-code pipelines

✓ Maintenance

✓ **Risk-based scoring dashboard**



Detection library – Live demo

Depth and breadth of library

Categorization

MITRE mapping

High-value detections

FalconForce

Search...

- Different From Requester - Windows
- 0156 - ADCS Abuse Recently Issued Certificate Exchanged for Kerberos Ticket - Windows
- 0159 - Disable MFA - AWS
- 0160 - Large number of unauthorized requests - AWS
- 0165 - ASR Bypass Executable Content Advanced - Windows
- 0174 - Untrusted Executable Launched from ISO - Windows
- 0176 - PasswordManager Credential Theft - Windows
- 0178 - SQL Server suspicious childprocess - Windows
- 0202 - Pentest Logins - Windows
- 0203 - Metasploit Logins - Windows
- 0204 - DNS Dump From LDAP - Windows
- 0206 - Creation Of Files Commonly Used By Exploit Tools - Windows
- 0208 - WMI Security Product Discovery - Windows
- 0209 - Oracle Suspicious Command Execution - Win
- 0210 - Usage of Self Managed Remote Access Software - Windows
- 0217 - DLL Planting in Default System Path - Windows**
- 0222 - Suspicious Named Pipes - Windows
- 0224 - ADEplorer Usage - Windows
- 0225 - AV Detection on Server or DC - Windows
- 0229 - LSASS dumping edr bypass - Windows
- 0230 - LSASS patching - Windows
- 0231 - Impacket Pass The Hash - Windows
- 0232 - Unexpected Process Accessing KeePass File - Windows
- 0243 - Dumping MSOL Password - Windows
- 0244 - PowerShell Azure API Usage by Non Admin Account - Azure
- 0247 - Sensitive Azure Resource Accessed using Device Token - Azure
- 0254 - Users added as exclusion to MFA required policy - Azure
- 0255 - Uncommon Azure shell activity by a user - Azure
- 0267 - Failed Logins from same Source IP for Users from Multiple Countries - Azure
- 0284 - RDP Login on Domain Controller - Windows
- 0296 - TGT requested with suspicious tools - Windows
- 0343 - LSASS Memory Copy Created via Fork or Snapshot - Windows
- 0376 - TGS requested with suspicious tools - Windows
- 0388 - EvilWinRM Usage - Windows
- 0401 - WindowsDivert Driver Usage - Windows
- 0527 - Password spraying against AD - Windows
- 0544 - Script Interpreter Loading DotNet Assembly From Memory - Windows

FalconForce.blue > FalconForce Detection Repository > 0217 - DLL Planting in Default System Path - Windows

This use-case is classified as **high-value**. High-value use-cases have proven to be highly effective in detecting confirmed malicious behavior in most environments.

Metadata

ID	OS
0xFF-0217-DLL_Planting_In_Default_System_Path-Win	WindowsEndpoint, WindowsServer

ATT&CK Tags

Tactic	Technique	Subtechnique	Technique Name
TA0003 - Persistence	T1574	001	Hijack Execution Flow - DLL Search Order Hijacking
TA0004 - Privilege Escalation	T1574	001	Hijack Execution Flow - DLL Search Order Hijacking
TA0005 - Defense Evasion	T1574	001	Hijack Execution Flow - DLL Search Order Hijacking
TA0003 - Persistence	T1574	002	Hijack Execution Flow - DLL Side-Loading
TA0004 - Privilege Escalation	T1574	002	Hijack Execution Flow - DLL Side-Loading
TA0005 - Defense Evasion	T1574	002	Hijack Execution Flow - DLL Side-Loading

Utilized Data Sources

Log Provider	Event ID	Event Name	ATT&CK Data Source	ATT&CK Data Component
MicrosoftThreatProtection	ImageLoaded		Module	Module Load

Detected attack

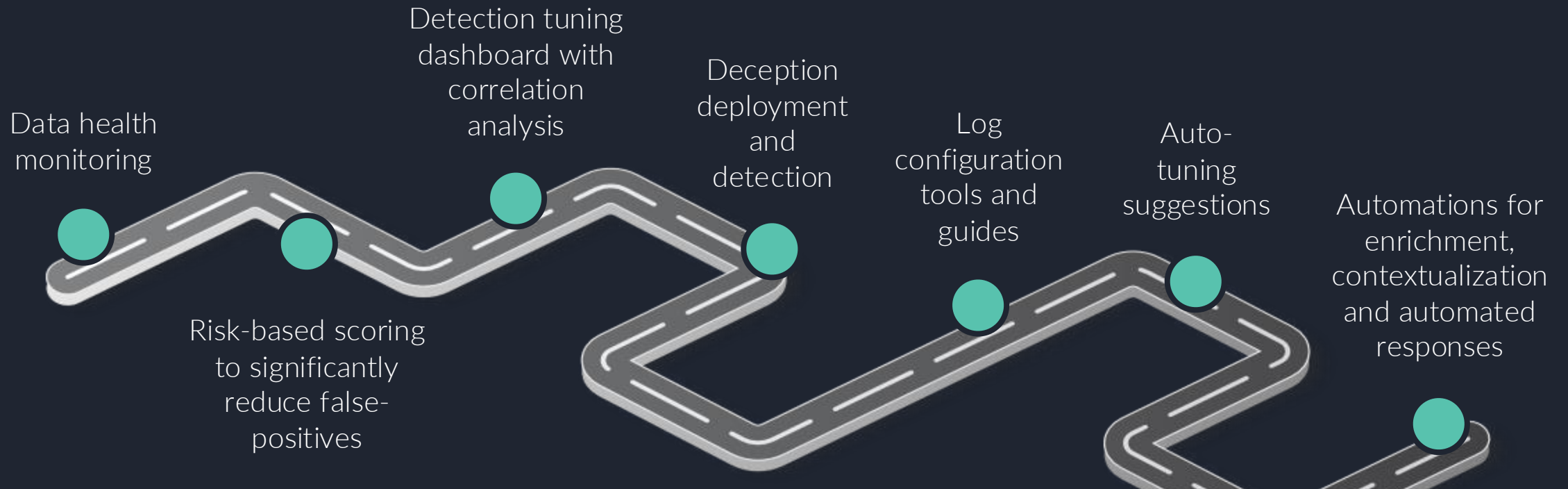
Attackers can use various techniques for DLL hijacking, allowing them to execute a malicious DLL as part of an existing program. One method to achieve DLL hijacking is to place DLLs in the system path. By default, programs will search this path in case of DLLs being referenced that do not exist on disk.

Version History

Version	Date	Impact	Notes
1.6	2024-06-28	minor	Modified the usage of FileProfile to exclude results if the call to the FileProfile API has failed.
1.5	2023-01-03	minor	Lowered the case of hashes that are fed to the FileProfile function due to case sensitivity.
1.4	2022-11-01	minor	Use default_global_prevalence variable to allow customizing handling of empty GlobalPrevalence
1.3	2022-10-11	minor	Added missing pre-filter
1.2	2022-05-20	minor	Updated the response plan.
1.1	2022-02-22	minor	Use ingestion_time for event selection and include de-duplication logic.
1.0	2021-11-30	major	Initial version.



Roadmap



On top of only detection content, we're working on providing other high-quality content for your SOC!

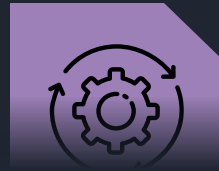


Keep an eye out for the following webinar

Risk-based
scoring



Automation



SOAR





Thank you! Questions?



info@falconforce.nl



<https://falconforce.nl>



[@falconforceteam](https://twitter.com/falconforceteam)



<https://linkedin.com/company/falconforce>

Relevant links

FalconFriday – a repository of free detections

<https://github.com/FalconForceTeam/FalconFriday/>

Deploying Detections at Scale — Part 0x01

<https://falconforce.nl/deploying-detections-at-scale-part-0x01/>

FalconForge – A basic version of our deployment tool

<https://github.com/FalconForceTeam/FalconForge>

The slides will be available soon

<https://falconforce.nl/>



Appendix

Sentry Detect models

Sentry Detect – Managed Detection Engineering

	Booster	Pro	Enterprise
	Usage of custom detection content		
<i>Portal access to content</i>	✓	✓	✓
<i>Deployment automation</i>	✓	✓	✓
<i>Content available</i>	All detections	All detections	All detections
<i>Allowed deployment</i>	Up to 50 detections	5 detections/month	Unlimited, fair use
<i>Tuning and major versions</i>	Up to 50 service credits Used for new detections + tuning	5 service credits/month Used for new detections or major version updates + tuning	10 service credits/month Used for new detections or major version updates + tuning
<i>FalconForce support</i>	✗	Minor version updates Ongoing detection tuning	Minor version updates Ongoing detection tuning
<i>Custom development</i>	✗	✓	✓
<i>Self-deployment & tuning</i>	✗	✗	✓
	Go fast	Regular new detections	Deploy and tune together



Sentry Detect – Managed Detection Engineering

	Booster	Pro	Enterprise
Pipelines-as-code			
<i>Implementation of pipelines</i>	Optional	✓	✓
<i>Usage of pipelines</i>	Optional	✓	✓
<i>Updates and support</i>	✗	✓	✓
Risk-based scoring dashboard			
<i>Risk-based scoring engine & dashboard</i>	✗	✗	✓
<i>Updates on engine</i>	✗	✗	✓
Fee and licensing			
<i>Fee structure</i>	One-time fee	Subscription	Subscription
<i>License model</i>	License to keep	License to use or to keep	License to use or to keep

