

8ème édition de la
Soirée du Test
Logiciel
Sophia -
Antipolis



13 octobre 2025
15h à 22h30



Amadeus, Biot



Soirée du
test logiciel



Sophia Antipolis

**Au-delà des tests classiques :Automatiser l'audit de sécurité logiciel
depuis le binaire avec MOABI**

Sécurisation des produits et de la
supply chain



Nicolas Gaume



MOABI

Société familiale créée en 2022, rentable (produit industriel chez plusieurs clients) et 100% souveraine. 15 ans de R&D pour concevoir la plateforme MOABI.

Aucune levée de fonds ni aucun investisseur extérieur.





Soirée du test logiciel

Conférence





Ce que vous allez découvrir aujourd'hui

1. Les limites des tests classiques de sécurité
2. Pourquoi la supply chain est devenue le maillon faible
3. L'avantage décisif de partir du binaire
4. Comment l'automatisation devient obligatoire (CRA/NIS2)
5. Exemples concrets et démonstrations

Objectif : Comprendre comment automatiser la sécurité produit





Pentests

- Audit manuel par des experts
- Recherche active de vulnérabilités exploitables

Tests de sécurité applicative (SAST/DAST)

- SAST : analyse du code source (si disponible)
- DAST : tests dynamiques sur l'application déployée

Les limites

Limite #1 : Le snapshot

Limite #2 : Besoin du code source

Limite #3 : Périmètre limité

Limite #4 : Scale impossible





Votre code
✓ Testé et maîtrisé

Code Fournisseur
x Pas de visibilité

Bibliothèques Open Source
x Non vérifiées

Legacy
x 10 - 15 ans ?

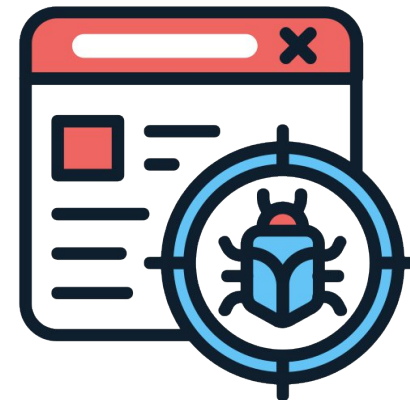
Dépendances transitives
x Inconnues

Paradoxe moderne :

→ Vous dépendez de code que vous n'avez jamais vu

→ Vous intégrez ce que vous ne pouvez pas vérifier

→ Vous livrez ce que vous ne maîtrisez pas





Cas #1 : SolarWinds (décembre 2020)

Cible : Pipeline de build du logiciel Orion

Attaque : Backdoor SUNBURST injectée pendant la compilation

Impact : 18 000 organisations compromises

Détection : 6 mois après déploiement

Binaire non signé

→ Les clients installaient eux-mêmes le malware en "mettant à jour"

Cas #2 : XZ Utils (mars 2024)

Cible : Bibliothèque de compression xz/liblzma

Attaque : Backdoor insérée dans le processus de build (absente du code source Git)

Impact potentiel : Accès root sur millions de serveurs Linux

Détection : Par hasard (500ms de latence SSH remarquée)

→ Code source propre ✓ | Binaire compromis ✗

Cas #3 : Aéroports européens (septembre 2025)

Cible : Prestataire IT gérant les services aéroportuaires

Attaque : Compromission du fournisseur de services

Impact : Perturbations massives dans plusieurs aéroports européens

Effet domino : Un fournisseur compromis → des dizaines d'aéroports paralysés

→ Le fournisseur était le point d'entrée, pas la cible finale



Code source

```
#include <stdio.h>
int main(void)
{
    printf("Bonjour!")
}
```

Code binaire

```
0110101 0110100
0100110 0101010
1010101 0110100
0101001 0110101
```



Soirée du
test logiciel

Pourquoi partir du binaire ?

Conférence





CRA / (UE) 2024/2847
Produits numériques

- Audit cyber des produits
- Sécurisation supply chain
- Security by design
- Analyse de risques
- Documentation 'cyber'
- Analyse de risques
- Période d'assistance (durée de vie du produit)
- Suivi vulnérabilités
- Gouvernance & contrôles

NIS 2 / (UE) 2022/2555
Opérateurs d'importance vitale

- Sécurisation supply chain
- Analyse de risques cyber
- Gestion des incidents
- Continuité d'activité
- Cryptographie renforcée
- Gouvernance et contrôles

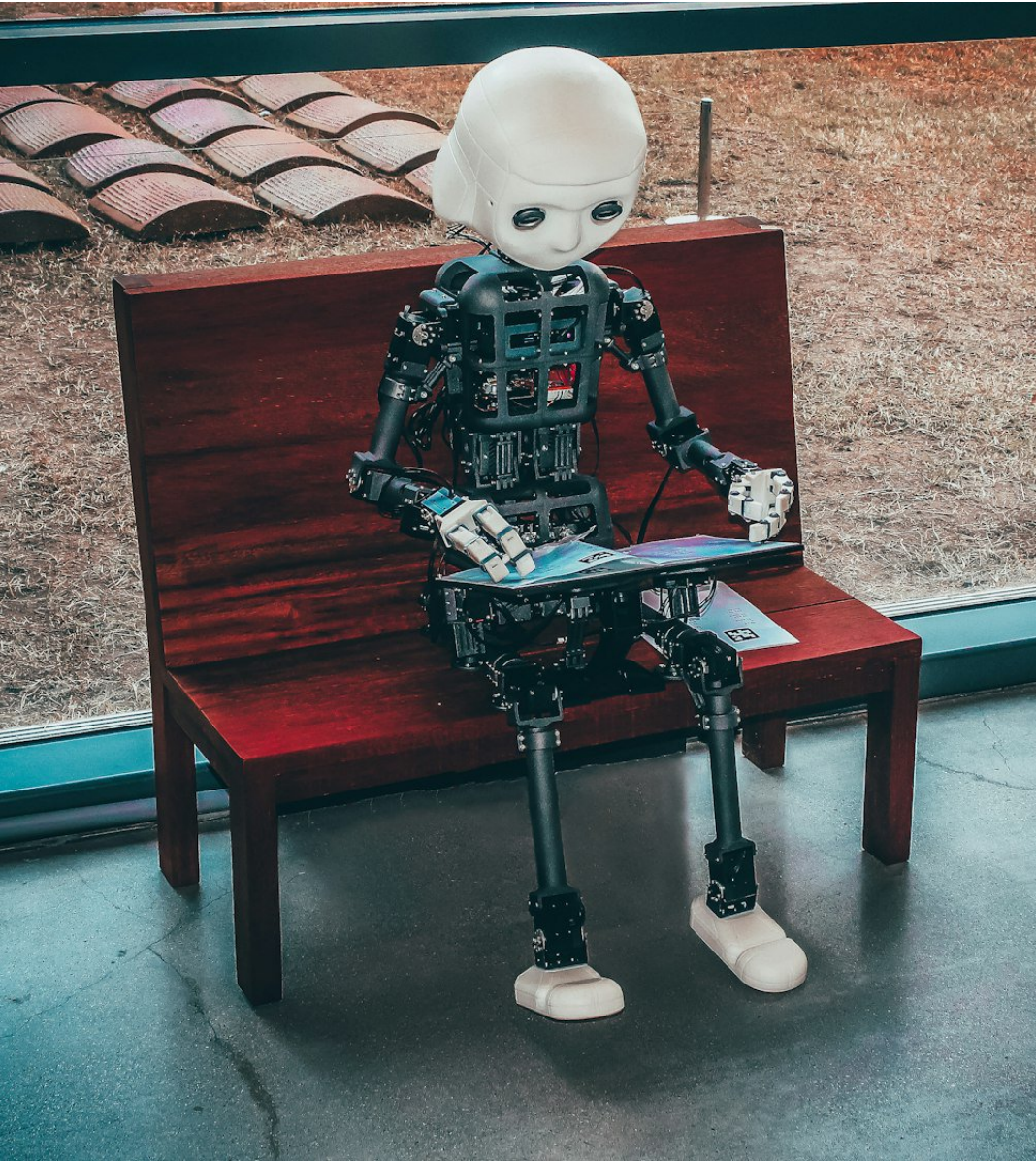
La cybersécurité devient un standard industriel.
À appliquer. À documenter. À démontrer.
L'automatisation rend cela possible.



Soirée du
test logiciel

L'automatisation : un impératif de scale

Conférence



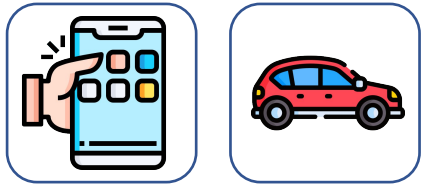
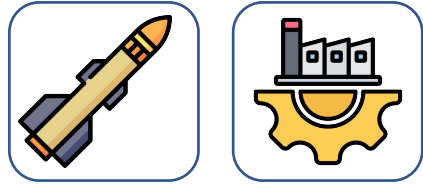
- Répétable avec les mêmes paramètres
- Focus remédiation, pas recherche de vuln.
- Gain de temps (5 000+)
- Éviter les tâches répétitives
- Surveillance continue (alertes)
- Scaling



Soirée du
test logiciel

Produit : Plateforme de Reverse Engineering automatique

Conférence



Logiciels / Binaires



Vulnérabilités
(8métriques)



SBOM



CVE
monitoring



Analyse de
risques



Plans de
remédiation



Licences



Odays



Vulnerability
Exploitability
eXchange (VEX)



MITRE
ATT&Ck

Plateforme pour la sécurité produit, le MCO et le MCS



POLICY	PRIORITY	FINDINGS
FAIL	P2	
PASS		
FAIL	P1	8 configuration issues detected
FAIL	P2	13 unique cves2 unique advisories
FAIL	P2	3 hardening problems
FAIL	P2	11 cryptographic problems
PASS	-	1 legacy problems
FAIL	P2	63 compliance problems
PASS	-	



Soirée du
test logiciel

Banque (Apk)

Conférence

POLICY	PRIORITY	FINDINGS
FAIL	P2	
PASS		
PASS	-	
FAIL	P2	💀 8 unique public vulnerabilities
PASS	-	🛡️ 1 hardening problems
PASS	-	🔑 1 cryptographic problems
FAIL	P2	👤 2 legacy problems
FAIL	P2	📅 39 compliance problems
PASS	-	



CREDENTIALS	
TYPE	DATA
🔥 Unsafe configuration	PermitEmptyPasswords yes
🔥 Unsafe configuration	PermitRootLogin yes
✖ HOW TO FIX?	

TYPE	DATA
🔥 Password hash	root:\$1\$1D0/1R8M\$f8J00u4W8GBZzdNzCgSW40:17059:0:99999:7:::
✖ HOW TO FIX?	



CWE-120 - BUFFER OVERFLOW

SCORE : 9



	Impact: 9	Confidence: 9	Risk: 10
Type	CWE-120 - BUFFER OVERFLOW		
Address	00171ca0		
Function	00171990		
Description	Vulnerability when calling function memcpy() : * function argument 3 has the following properties: user controled, remote control, arithmetic operation, upper function argument 2, upper function argument 6, stack variable at offset 0x24, read only data, return value from function, . * source has the following properties: not user controled, . * destination has the following properties: user controled, remote control, arithmetic operation, upper function argument 2, upper function argument 5, upper function argument 6, stack variable at offset 0x2c, read only data, data section, return value from function, global memory, writable memory. buffer overflow in third argument		
Backtrace	#00 <171ca0> void * __cdecl memcpy(__out_bcount_full_opt(_Size) void * _Dst, __in_bcount_opt(_Size) const void * _Src, __in size_t _Size) at: ./libopenvpn.so:0x171ca0 #01 <171990> PEM_read_bio_ex() at: ./libopenvpn.so:0x171990		



CWE-190 - INTEGER OVERFLOW OR WRAPAROUND

SCORE : 6



	Impact: 4	Confidence: 6	Risk: 10
Type	CWE-190 - INTEGER OVERFLOW OR WRAPAROUND		
Address	000ec0d7		
Function	000ebfc0		
Description	Vulnerability when calling function memset() : * function argument 3 has the following properties: user controled, arithmetic overflow, arithmetic operation, read only data, return value from function, heap memory, global memory, function pointer, memory not initialized, integer overflow in third argument		
Backtrace	#00 <ec0d7> void *memset(void *s, int c, size_t n) at: ./libopenvpn.so:0xec0d7 #01 <ebfc0> event_set_init() at: ./libopenvpn.so:0xebfc0		



```
1105c sprintf(r15@0x11016_0 + function_argument1 + 0x0 + 0x4 + function_argument1, rbp@0x11058_1, 6)
110bc sprintf(rbp@0x11084_1, r15@0x11016_0 + function_argument1 + 0x0 + 0x4 + function_argument1 + 0x1, 6, rsp + 0x8)
110d4 sprintf(sprintf(rbp@0x11084_1, r15@0x11016_0 + function_argument1 + 0x0 + 0x4 + function_argument1 + 0x1, 6, rsp + 0x8) + 0x6 +
r15@0x11016_0 + function_argument1 + 0x0 + 0x4 + function_argument1, r15@0x11016_0 + function_argument1 + 0x0 + 0x4 + function_argument1,
6)
110fc sprintf(r15@0x11016_0 + function_argument1 + 0x0 + 0x4 + function_argument1 + 0x6 + r15@0x11016_0 + function_argument1 + 0x0 + 0x4 +
function_argument1, rbp@0x11084_1, 6)
11146 sprintf(rsp + 0xc + 0x1 + 0x18 + rsp@0x11141_1, rsp + 0x4, 6, function_argument1 + 0x0 + 0x0)
1118c sprintf(rsp + 0x10 + 0x2 + function_argument1 + 0x0, function_argument1 + 0x0 + 0x18 + rsp@0x11185_1, 6)
111a6 function_11004(function_argument1, rsp + 0x4)
}
```

```
static int function_111d0() {
11212 function_111d0(1, function_argument1 + 0x1c + 0x4, function_argument3, 0x1 + 0x1c + 0x4)
1121c function_111d0()
11238 fread(0x1 + 0xc, function_argument1 + 0x1c + 0x4, function_argument3, 0x1 + 0x14)
11242 fread(0x1 + 0x10, fread(0x1 + 0xc, function_argument1 + 0x1c + 0x4, function_argument3, 0x1 + 0x14), 0, 0) // VULNERABILITY
}
```



Soirée du
test logiciel

CVE Monitoring

Conférence

SCAN DATE	UNIQUE SCAN CVEs	NEW CVEs	Last Modified
14 February 2025	117	1519	3 October 2025
13 February 2025	114	1422	3 October 2025
22 July 2025	422	1176	3 October 2025



✂ HOW TO FIX?

⚠ P2 PRIORITY



Critical Security Alert: This software is not compiled with an optimal set of compiler flags implementing all the available defense in depth mechanisms, resulting in structural weaknesses in the application at runtime. This is a direct threat to the integrity of your Information System, of high priority.



Immediate Action Required: Compilers have made considerable progress over the past 20 years to implement generic counter measures against many critical attack classes and exploitation vectors. Ensuring that those hardenings are activated at compile time participates to the defense in depth of software at run time and should be implemented to comply with industry best practices. You may read more about the lack of hardening class of vulnerability [here](#) and more about modern compiler flags [here](#)



Update Process: Fixing hardening problems requires access to the source code and involves setting the appropriate compiler flags at build time. The recommended set of CFLAGS for a C compiler such as GCC, g++ or clang is:

```
CFLAGS="-fPIC -pie -fPIE -fstack-protector-all -O2 -D_FORTIFY_SOURCE=2 -WI,-z,now "
```



Additional Information: For extra hardening, you may consider adding the following CFLAGS as well, to enable [stack clashing protection](#), [call-used register wiping](#) and [Control-flow Enforcement Technology](#) (CET, Intel processors specific) respectively:

```
CFLAGS_EXTRA="-fstack-clash-protection -fzero-call-used-regs=used-gpr -fcf-protection"
```

You may read more about the recommended MOABI compiler flags in the remediation section of this report, or by refering to the manual of your specific compiler.

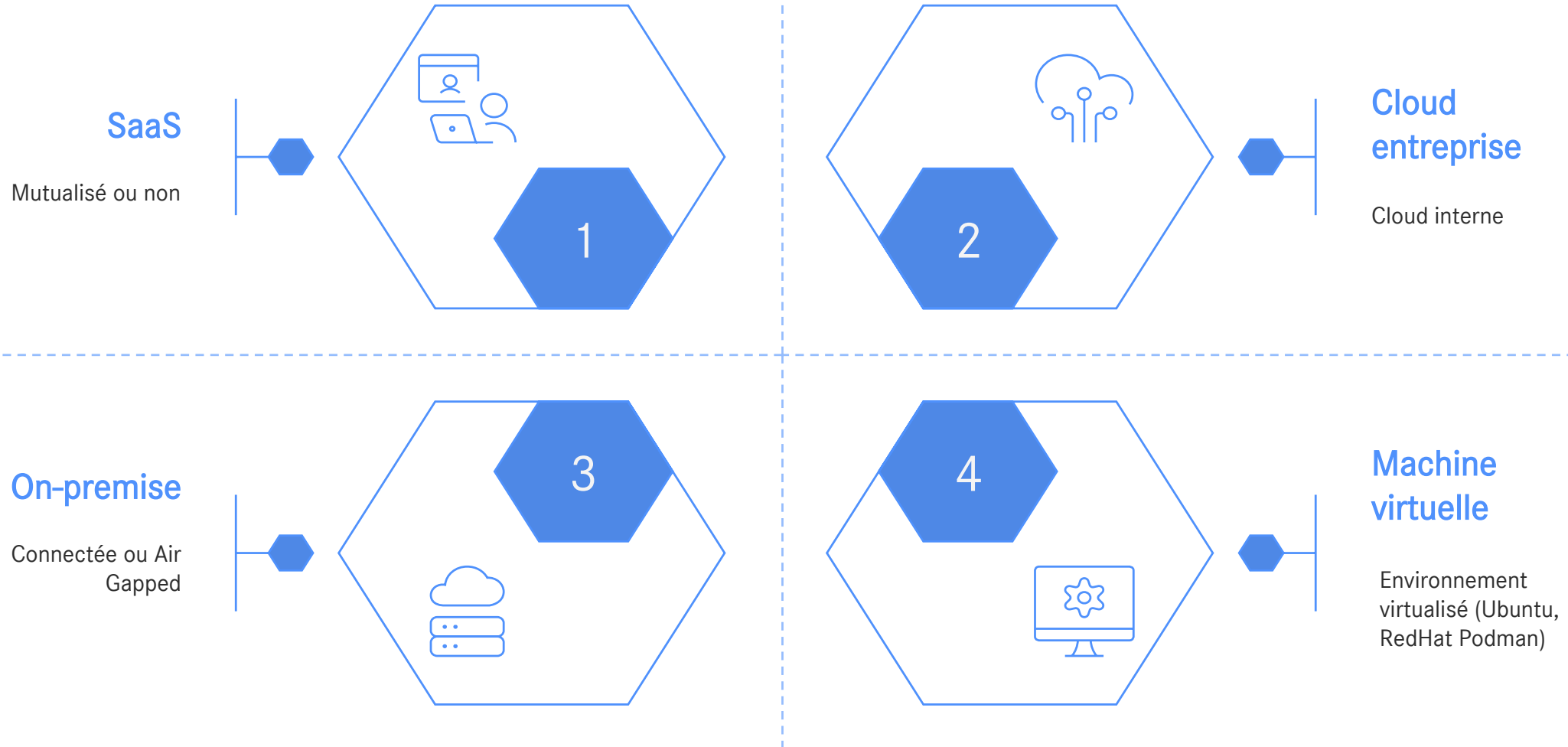
Please refer to the [table](#) above for a list of missing hardening features.

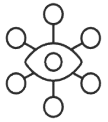


Soirée du
test logiciel

Déploiement

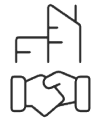
Conférence





**Direction
Générale**

**Conformité
Juridique
Avantage
Vision 360°**



**Équipes
Commerciales**

**Preuves tangibles
Arguments de vente
Proactivité cyber
Confiance**



**Équipes
Techniques**

**Automatisation
Focus innovation
Mesurer la cyber
Impact augmenté**



**Direction
Financière**

**ROI rapide
Coûts maîtrisés
Sanctions évitées
Vente cyber**



Vos Clients

**Transparence totale
Preuves objectives
Réactivité
Partage de données**

**MOABI ne crée pas seulement de la conformité et des rapports.
MOABI crée de la valeur alignée pour toute l'organisation.**

8ème édition de la
Soirée du Test
Logiciel
Sophia -
Antipolis



13 octobre 2025
15h à 22h30



Amadeus, Biot



Soirée du
test logiciel

✓ Sophia Antipolis

SCAN ME



Merci de votre attention !

Pour mon contact ou des démos, il
faut venir me voir :)

Votre avis nous intéresse