YOU « TRY » TO DETECT MIMIKATZ ;)

I saw your credentials!

I saw your credentials!

I saw your credentials!
again!

again & again!
Whoami

THEY TOLD ME I COULD BE ANYTHING I WANTED

SO I BECAME A DOMAIN CONTROLLER

mimikatz
DCSync
DCShadow

Vincent LE TOUX
@mysmartlogon
Does this remind something to you?

<Insert vendor>

<Insert vendor>

STOPS MIMIKATZ
Busylight stops mimikatz!

Busylight

STOPS MIMIKATZ

Demo
COMMON MISTAKE: MIMIKATZ IS NOT JUST ABOUT CREDENTIAL COLLECTION
No excuse: ATT&CK from Mitre

<table>
<thead>
<tr>
<th>Tactic</th>
<th>Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Persistence</td>
<td>Security Support Provider</td>
</tr>
<tr>
<td>Privilege Escalation</td>
<td>SID-History Injection</td>
</tr>
<tr>
<td>Defense Evasion</td>
<td>DCShadow</td>
</tr>
<tr>
<td>Credential Access</td>
<td>Manipulation</td>
</tr>
<tr>
<td>Credential Access</td>
<td>Credential Dumping</td>
</tr>
<tr>
<td>Credential Access</td>
<td>Credentials in Files</td>
</tr>
<tr>
<td>Credential Access</td>
<td>Private Keys</td>
</tr>
<tr>
<td>Lateral Movement</td>
<td>Pass the Hash</td>
</tr>
<tr>
<td>Lateral Movement</td>
<td>Pass the Ticket</td>
</tr>
</tbody>
</table>

Golden ticket

[https://mitre.github.io/attack-navigator/enterprise/](https://mitre.github.io/attack-navigator/enterprise/)
3 main areas

- Local LSASS hacking
  - SEKURLSA::LogonPasswords
- Remote AD hacking
  - LSADUMP::DCSync, kerberos::golden
- MISC
  - CRYPTO::Certificates

If you want to stop mimikatz, you have to stop every techniques!

AN EXAMPLE: UNDERSTANDING THE GOLDEN TICKET ATTACK DISCLOSURE
A reminder about the golden ticket attack

Presented at BlackHat USA 2014

The reactions in the security community 1 year later.
Nothing found in US CERT databases

Is that because the "golden tickets" attack is not a vulnerability?

No analysis was done?
Thanks to wikileaks for more insight

Don’t mix BlackHat with RSA!

Root cause:
Wrong information flow in the infosec community
TRYING TO DETECT MIMIKATZ
Buy an Antivirus (or not) 1/2?

1) Mimikatz is not a «virus»

2017

+13 AV
Only +4 detection?

2019

https://www.virustotal.com/#/file/b985bca0eaf044c321f1d4274ec1cf9660e5d90553c557b3769f0bce744fa3ae/detection
Buy an Antivirus (or not) 2/2 ?

2) If it worked 100% of time, we won’t have this discussion ;-) 

Example with Windows Defender on my computer:

- The first official version of mimikatz (the one shown in the previous slide) compiled in 2013

- Analysis performed March, 6th 2019

Root cause: Signature instead of « Behavior » detection
Time to Do It Yourself?

Let’s start with the basics and progress

- Idea: you cannot win the « tour de France » if you do not know how to ride a bike

- Same with mimikatz
DETECT: THE CISO WAY
Let’s try the CISO way

Pick a framework
• France: ANSSI
• Germany: BSI
• USA: STIG

Complete with watch
• CERT alerts
• Conferences follow-up

Connect a big BOX
• Rely on your vendor rules
• And start handling alerts
Example of frameworks
What about the watch?

Follow your national CERT (CERT-FR, CERT-Bund, US-CERT, …)

If you have to follow only one person on twitter:
@PyroTek3 – Sean Metcalf is the author of www.adsecurity.org and retweet any AD focused topics

So many interesting AD leaders:
@gentilkiwi – Mimikatz’s author for new features ;-)  
Specter ops team: @harmj0y, @tifkin_, @wald0, @cptjesus, @enigma0x3, ..
@DirectoryRanger – linked with ERNW (Troopers)
List of persons to follow: https://adsecurity.org/?page_id=4031

Don’t follow @NerdPyle since he doesn’t talk AD anymore ;-}
A BOX ? What about a SIEM ?

A Siem « process » ALL events you are sending to it
And you « detect » mimikatz !

Wait …
Frameworks & Watch vs Reality

**Good point:** frameworks are explicit (no unlimited list of problems to fix)

**Twitter** is the best source of data

**But:**
- Based on the assumption you have no history (few domains, ...)
- Not all attacks are covered by CERT alerts
- Heterogeneous coverage between framework
- Basic security problem not covered
SIEM vs Reality

What you think: «new attacks automatically covered»

What you have:
- An increase of 30% of your EPS
- Brute force attack detected
- Logs collected (which logs?)

What you don’t have:
- DCSync, Golden ticket, ... Detection

In short no mimikatz detection
And compliance?

Compliance reports from a AD security vendor:
It does not detect mimikatz...
In summary

- Frameworks are structured but do not cover all attacks
- Watch covers advanced topics but not the basic one
- SIEM process logs but are they the right logs and what about the rules?
LETS GET TECHNICAL: ZOOMING ON CREDENTIAL THEFT
Evolution of LSASS security posture

Windows 7:
Mimikatz is a post compromission tool
This is not a vulnerability

Windows 8.1:
Prohibit storage of sensitive passwords
(“Restricted Admin mode for Remote Desktop Connection”, “LSA Protection”, “Protected Users security group”)

Then:
More and more protection such as virtualisation
New ways to prevent mimikatz

Mimikatz requires the « debug privilege » - Just remove it!

psst: run mimikatz as system ;-)
## Status of LSA protection

<table>
<thead>
<tr>
<th>Protection mechanism</th>
<th>Applicable Windows version, edition</th>
<th>Requirement</th>
<th>Bypassed by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prevent credentials to be sent on a remote server</td>
<td>Windows 7 patched</td>
<td>None</td>
<td>Allow authentication by « pass-the-hash » &amp; « pass-the-ticket » via CredSSP</td>
</tr>
<tr>
<td>Force Kerberos only SSP</td>
<td>Windows 7 patched</td>
<td>None</td>
<td>Kerberos ticket stolen</td>
</tr>
<tr>
<td>Restrict access to LSA process on the OS</td>
<td>Windows 7 patched</td>
<td>Requires LSA signature of ALL third party components using EV certificate</td>
<td>!processprotect /process:lsass.exe /remove</td>
</tr>
<tr>
<td>Isolate secrets from OS on Hypervisor</td>
<td>Windows 10 Enterprise only</td>
<td>Secure boot (TPM) &amp; HyperV (Not VMWare)</td>
<td>Capture credentials before being stored</td>
</tr>
</tbody>
</table>

The most effective protection is difficult to implement when dealing with legacy
But there is no place such as LSASS.exe

Methods to read LSASS.exe memory

- Genuine Debug access
  - Dll injection
  - Memory copy
  - Requires Debug Privilege

- Genuine access to passwords
  - Security Package
  - Authentication package
  - Password filters
  - («ProjectSauron»)
  - («Calais database»)

- Genuine memory access
  - Smart Cards driver
  - Sub Package (*)

Lessons learned: removing «debug privilege» is not enough

(*) https://docs.microsoft.com/en-us/windows/desktop/secauthn/subauthentication-packages
In fact, LSASS is only a «gold mine»

LSASS.exe

Golden flakes still in the river
Demo 3 – driver + SSPI

I don't want one position, I want all positions!
ZOOMING ON ACTIVE DIRECTORY
How it works: 1/2

In short: the golden ticket factory
How it works: 2/2

1) Retrieve the credentials to open the first « safe »
2) Then abuse it to get other credentials to open other safes

Quickest way to propagate to other domains
The root causes

- It is not about credential / authentication but about AD secret management
- It is about network segmentation
- It is about having unknown trust relationship with other domains

Is a technical project the solution?
Demo 4: And ... trust are not a strict border
HOW TO « DETECT » MIMIKATZ ?
Rule #1: accept you can’t

You don’t need mimikatz to be mimikatzed

Attacks implemented in other tools. Example:

- Credential dump: Quarks PwDump
- DCSync: secretsdump.py from Impacket
- Kerberos, DPAPI: GhostPack
- DCSync, Golden ticket: MakeMeEnterpriseAdmin

- New mimikatz: kekeo!
Rule #2: apply the author recommendations

Do you know @gentilkiwi published yara rules?

Same for DCSync Detection?
Check out (and adapt)
https://gist.github.com/gentilkiwi/dcc132457408cf11ad2061340dcb53c2
Rule #3: Know your scope!

I’m still surprised to see companies that:

- Do not know how much AD they have
- Cannot list open shares (with passwords) or local admins
- Have still some MS17-010 unpatched

My gift to the community: https://www.pingcastle.com
CONCLUSION
Mimikatz is a brand

You cannot fight an image

And for techies

- You can (sometimes) detect mimikatz as a whole application

When I see 'solutions' trying to detect/stop #mimikatz by identifying DLL loading list/order...
You, of course, are aware real malwares using embedded versions are built with #mimikatz without tons of modules?

- But maybe you should understand the attack behind rather than looking for a tool...

http://github.com/gentilkiwi/mimikatz
http://github.com/vletoux/pingcastle
@mysmartlogon