THE PAST, PRESENT & FUTURE OF ENTERPRISE SECURITY

THE ‘GOLDEN AGE’ OF ATTACK AUTOMATION
Marcello Salvati

- @byt3bl33d3r
- https://github.com/byt3bl33d3r
- Lead researcher @coalfirelabs

- Years of experience building open source security tools
Enterprise Security

It’s big. It’s a thing. It’s a problem. It’s complicated.
Challenges

- Huge networks
  - A lot of times ‘inherited’ from acquisitions
  - Lack of visibility, inventory, patch management, documentation

- Security vs. business continuity
  - Limited budgets for security
  - Non-effective communication
  - Often investing in products, not people
  - Legacy system(s), application(s)

We can be here all week talking about this...
The typical corporate network
Realistically....
The Past

Pre-PowerShell Era
Lack of tooling and tradecraft...

... especially for very large networks

- Usually, most post-exploitation tools were just wrappers
- In dire need of automated situational awareness
- Implants usually all touched disk
The Game Changers

- **Mimikatz**
  - https://github.com/gentilkiwi/mimikatz

- **SMBExec**
  - https://github.com/brav0hax/smbexec

- **Responder**
  - https://github.com/lgandx/Responder
Icing on the cake

- PowerShell... omfg
  - Defcon 18
  - David Kennedy, Josh Kelly
The Present
PowerShell Era
PowerShell, PowerShell, PowerShell...

- Built into every Windows OS by default
- Extremely powerful as it allows full dynamic access to .NET
- PowerShell < V4.0 had no protections in place for in-memory script execution
- Has built in features that can be abused by attackers

Needless to say, this was the dream (or nightmare) ...
The Game Changers V2.0

- Powerview & PowerSploit
  - https://github.com/PowerShellMafia/PowerSploit

- Empire
  - https://github.com/EmpireProject/Empire

- BloodHound/Sharphound
  - https://github.com/BloodHoundAD/BloodHound
  - https://github.com/BloodHoundAD/Sharphound
Big networks & limited time? Not an issue!

- CrackMapExec
  - https://github.com/byt3bl33d3r/CrackMapExec

Own an entire subnet in minutes!
EMPIRE

278 modules currently loaded
1 listeners currently active
0 agents currently active
Why not automate the entire process?

- DeathStar
  - https://github.com/byt3bl33d3r/DeathStar

- GoFetch
  - https://github.com/GoFetchAD/GoFetch

Need to automate getting a foothold?

- IceBreaker
  - https://github.com/DanMcInerney/icebreaker
This sounds familiar...


Stuff that I'd like to see added

There is so much more that could be done with DeathStar: more domain privilege escalation techniques could be added, more lateral movement methods, the logic could be fine tuned a bit more, we could do some post-exploitation and SPN shenanigans etc.. The current release is definitely a rough first draft.

The game changer would be SMB Named Pipe pivoting. Once that’s in Empire this will truly ‘walk and talk’ like a worm.

Conclusion

DeathStar demonstrates that automating obtaining Domain Admin rights in an Active Directory environment is a clear possibility using existing open-source toolsets. I expect to see many more tools that do something like this in the near future (I personally know two people who are working on their own versions/implementations which is awesome, and I encourage more people to do so)

One final point I’d like everyone to reflect on: I put this together in 3-4 days. Imagine what a bunch of much more smarter people than me could do/have already done with more time and resources (cough cough nation states cough cough). That’s something that I think is particularly interesting.
THREE WEEKS LATER
Called it?

NotPetya Summary

- Initial infection in Ukraine accomplished by exploiting vulnerability in M.E.Doc software
- Infected systems then attempt to propagate the infection to other systems

- To infect other systems inside the organization, the malware steals credentials and propagates with built-in Windows tools WMI and PSEEXEC:
  PSEEXEC code snippet: C:\Windows\dllhost.dat \IP ADDRESS -accepteula -s -d
  C:\Windows\System32\rundll32.exe "C:\Windows\perfc.dat",#1 10
  "USERNAME:PASSWORD"
  WMI code snippet: C:\Windows\system32\wbem\wmic.exe /node:"IP ADDRESS"
  /user:"USERNAME" /password:"PASSWORD" process call create "C:\Windows
  \System32\rundll32.exe \"C:\Windows\perfc.dat\" #1 XX
  "USERNAME:PASSWORD"

- To infect additional systems outside the organization, the malware attempts to exploit the EternalBlue vulnerability

The Very Near Future (arguably the present)

C#/.NET
The attacker’s creed

WE NEED TO GO

DEEPER
The Power in PowerShell...

...comes from dynamically calling .NET!

Can we do this without going through PowerShell?
A perfect example

- DotNetToJScript
  - https://github.com/tyranid/DotNetToJScript
Something may be in the works 😊
C#/.NET!

- Quick Retooling in .Net for Red Teams
  - Circle City Con 2018
  - @Op_Nomad
  - https://github.com/dsnezhkov/typhoon
Let’s talk mitigation

(A.K.A things you can do right after this talk to harden your network)
Start with the basics

Don’t have an account lockout policy, segmentation, host isolation and inventory?
SMB Signing

One of the most overlooked and underrated AD security settings...
SMB Signing

- Following key needs to be set EVERYWHERE:
  - HKLM\System\CurrentControlSet\Services\LanManServer\Parameters\RequireSecuritySignature

- Test in lab before deploying to all systems!

Difficulty: EASY PEASY
Breaks Stuff: MAYBE
Situational Awareness

- Most of this functionality is considered a feature not a bug and is still there mainly for backwards compatibility reasons (a.k.a. Microsoft’s Curse)

- There are some TechNet PS scripts which allow you to harden session enumeration and SAMR remote access (shoutout to @ItaiGrady <3):
  - https://gallery.technet.microsoft.com/Net-Cease-Blocking-Net-1e8dcb5b

- If anyone has any pro-tips on how to mitigate AD information gathering on the cheap would love to hear it :)

Difficulty: HARD
Breaks Stuff: MAYBE
Domain Privesc

By far, the most common way I’ve found to escalate privileges is to look for passwords in SYSVOL & GPP
Domain Privesc

- Install KB2962486 on every computer used to manage GPOs which prevents new credentials from being placed in Group Policy Preferences.
  - https://support.microsoft.com/en-us/kb/2962486
- Delete existing GPP xml files in SYSVOL containing passwords.
- Don’t put passwords in files that are accessible by all authenticated users.

Difficulty: EASY\MODERATE
Breaks Stuff: NO
Cleartext Passwords in Memory

- This attack can’t be performed on Windows 2012R2+ and Windows 8.1+.
- On older systems KB2871997 should be installed EVERYWHERE
- [https://support.microsoft.com/en-us/kb/2871997](https://support.microsoft.com/en-us/kb/2871997)
- The following registry should be set EVERYWHERE and monitored:
  HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest\UseLogonCredential: Value 0 (REG_DWORD)
- Your Administrators should have a separate workstation for their administrative activities!

**Difficulty:** EASY  
**Breaks Stuff:** NO/MAYBE
Local Administrator Accounts

Here’s a good example of what **NOT** to do:
Local Administrator Accounts

- Microsoft LAPS:
  - https://adsecurity.org/?p=1790

Difficulty: MODERATE
Breaks Stuff: NO
Conclusion
Thanks!

ANY QUESTIONS?

You can find me at:
@byt3bl33d3r
byt3bl33d3r@pm.me