Red Team Tools and Techniques

Red For Detection

About Me

- Information Security Engineer With Fortune 100 Finance Institution
    - Over 20 years in IT with the last 17 focused on Infosec
    - Focused primarily on Defense and Prevention
    - Always thinking of new ways for detection
    - Passion for learning
- SANS Certified Instructor
    - Teach Mainly Blue Team Courses
        - SEC301, SEC401, SEC501, SEC511, SEC555
    - Have Taken Many Red team course (504, 560, 542)
- Love independent Research

Red For Blue

- Even if you will never perform a Pen-Test knowing how to “hack” is a crucial skill
  - Makes you more well Rounded
  - Gives better understanding or attacker techniques
  - Can help you better understand your environment
  - Hacking is kind of Fun
- More Valuable in My opinion
  - Detection is the number on reason to know Red Team techniques
  - Testing new and existing attacks and identifying how that changes your system will improve your detection capabilities

Obligatory Pop Culture Reference

Known Attacks

Nation States

Cyber Criminals

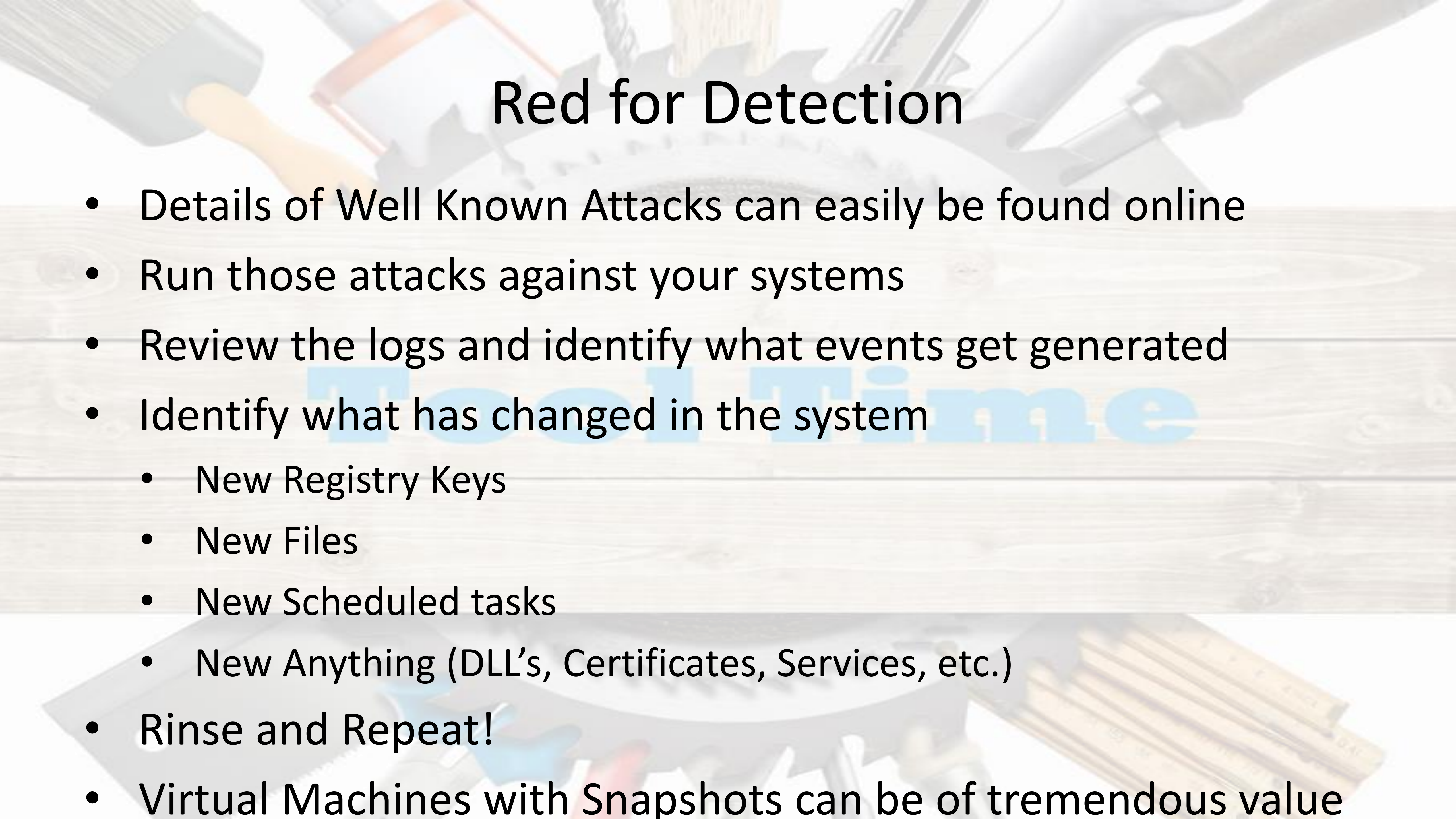
Evil Bad Actor

Insider Threats

Malware

Know Yourself

Zero Days



Red for Detection

- Details of Well Known Attacks can easily be found online
- Run those attacks against your systems
- Review the logs and identify what events get generated
- Identify what has changed in the system
    - New Registry Keys
    - New Files
    - New Scheduled tasks
    - New Anything (DLL’s, Certificates, Services, etc.)
- Rinse and Repeat!
- Virtual Machines with Snapshots can be of tremendous value

How do I know what/how to test?

- Often the most difficult and most fun part of the process
- Will require some trial and error
- Play with different attack techniques
- Useful Resources
  - SANS Classes……..Shameless plug!
  - Twitter……yes I said Twitter
  - Red Canary Atomic Red Team
  - MITRE ATT&CK
  - Previous Penetration Tests
  - Books on hacking

SANS and Offensive Security Training

GIAC WEB APPLICATION PENETRATION TESTER
GWAPT

GIAC CERTIFIED FORENSIC ANALYST
GCFA

GIAC CERTIFIED INCIDENT HANDLER
GCIH

GIAC CERTIFIED DETECTION ANALYST
GCDA

GIAC CERTIFIED PENETRATION TESTER
GPEN

GIAC CONTINUOUS MONITORING CERTIFICATION
GMON

OFFENSIVE®
security
OSWE

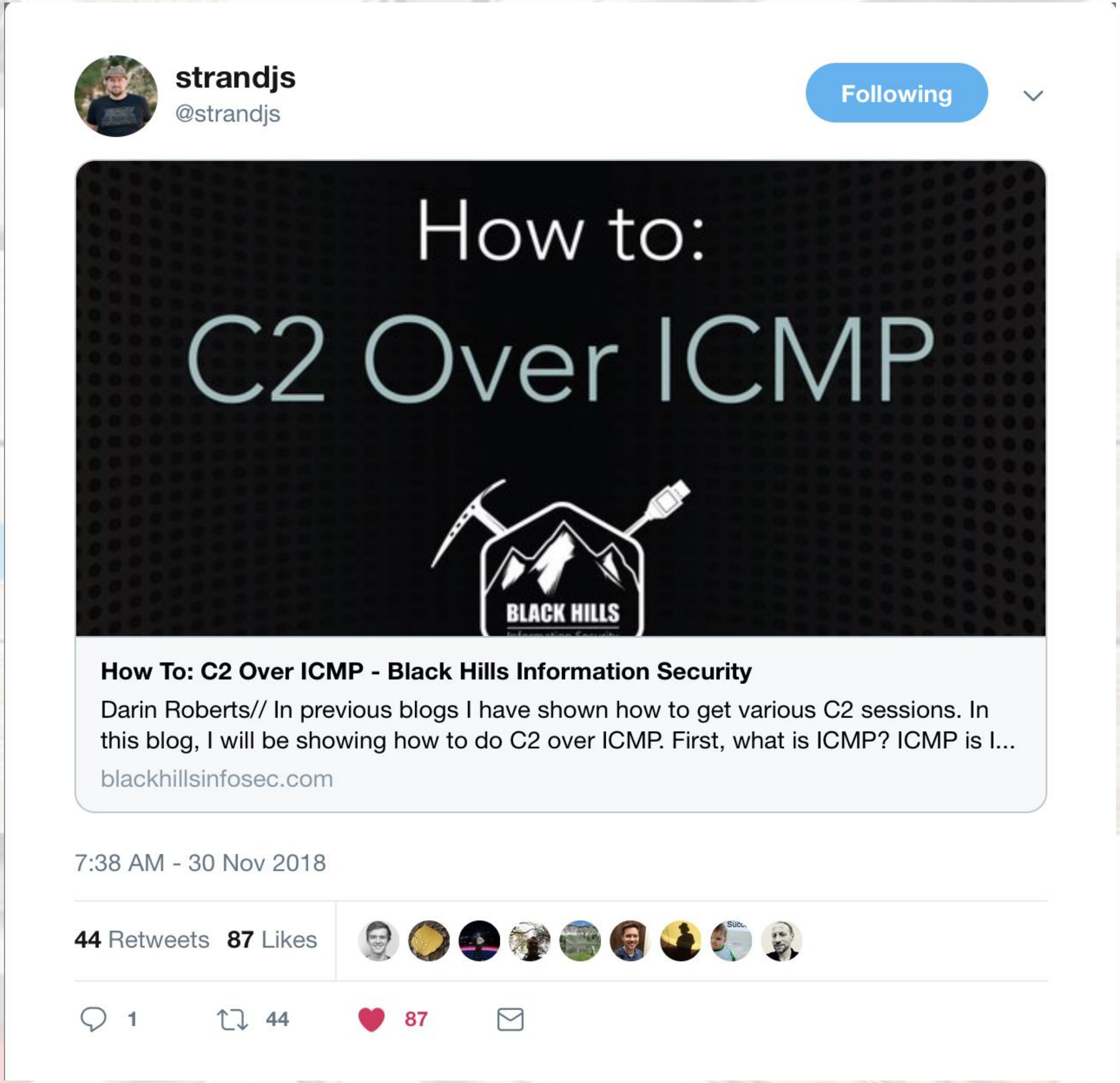
GIAC PYTHON CODER
GPYC

OFFENSIVE®
security
OSCP

Twitter

People to Follow:

- @strandjs
- @Jsnover
- @Hacksforpancakes
- @Malwarejake
- @Trustedsec
- @Hackingdave
- @Mubix
- @binnishah
- @deviantollam (physical Security)
- @jaysonstreet
- @enigma0x3
- @mattifestation
- People They Follow!



strandjs
@strandjs

How To: C2 Over ICMP - Black Hills Information Security

Darin Roberts// In previous blogs I have shown how to get various C2 sessions. In this blog, I will be showing how to do C2 over ICMP. First, what is ICMP? ICMP is I...

blackhillsinfosec.com

7:38 AM - 30 Nov 2018

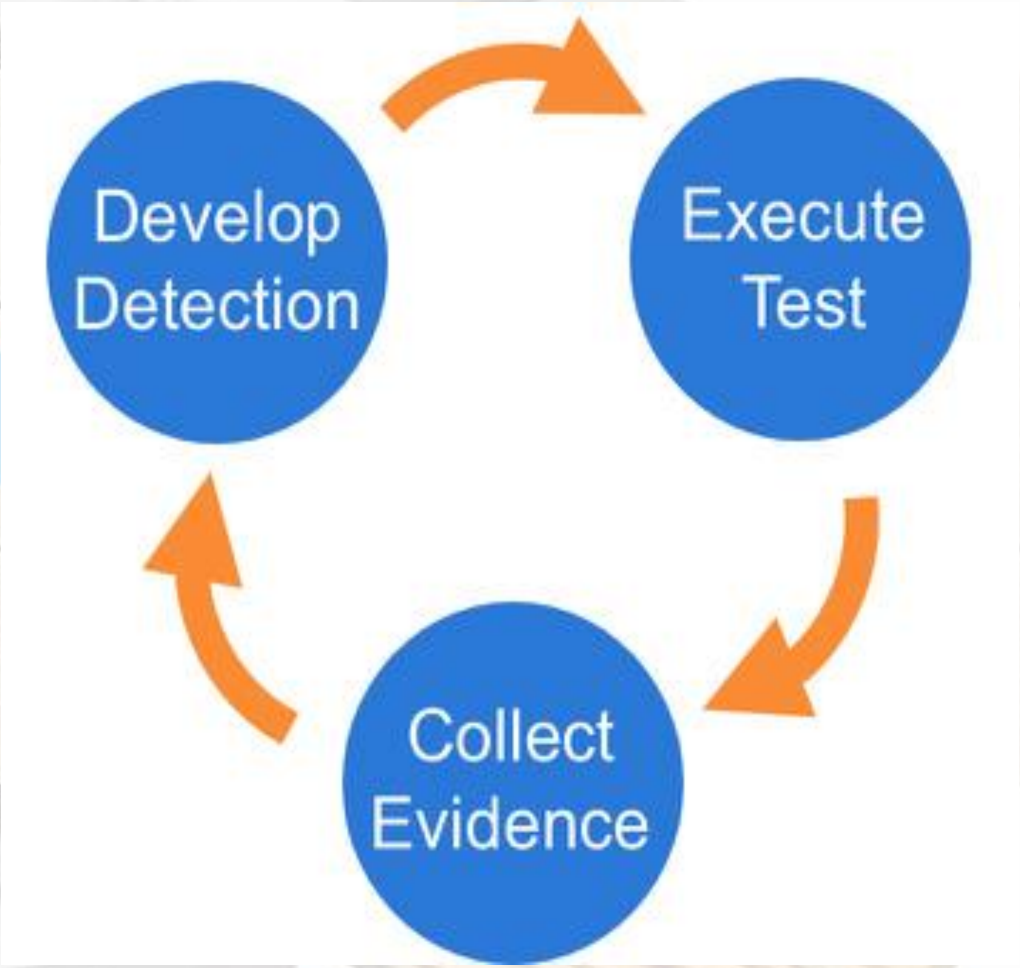
44 Retweets 87 Likes

Atomic Red ---- https://atomicredteam.io

- A Library of simple Tests anyone can execute to test your controls and detection
- Recommended Approach
  - Select a Test ([GitHub link](#))
  - Execute. Test
  - Collect Evidence
  - Develop Detection
  - Measure Progress
- Some Tests Mapped directly to MITRE ATT&CK
- Tests for Windows, Mac, and Linux

Atomic Red Cont.

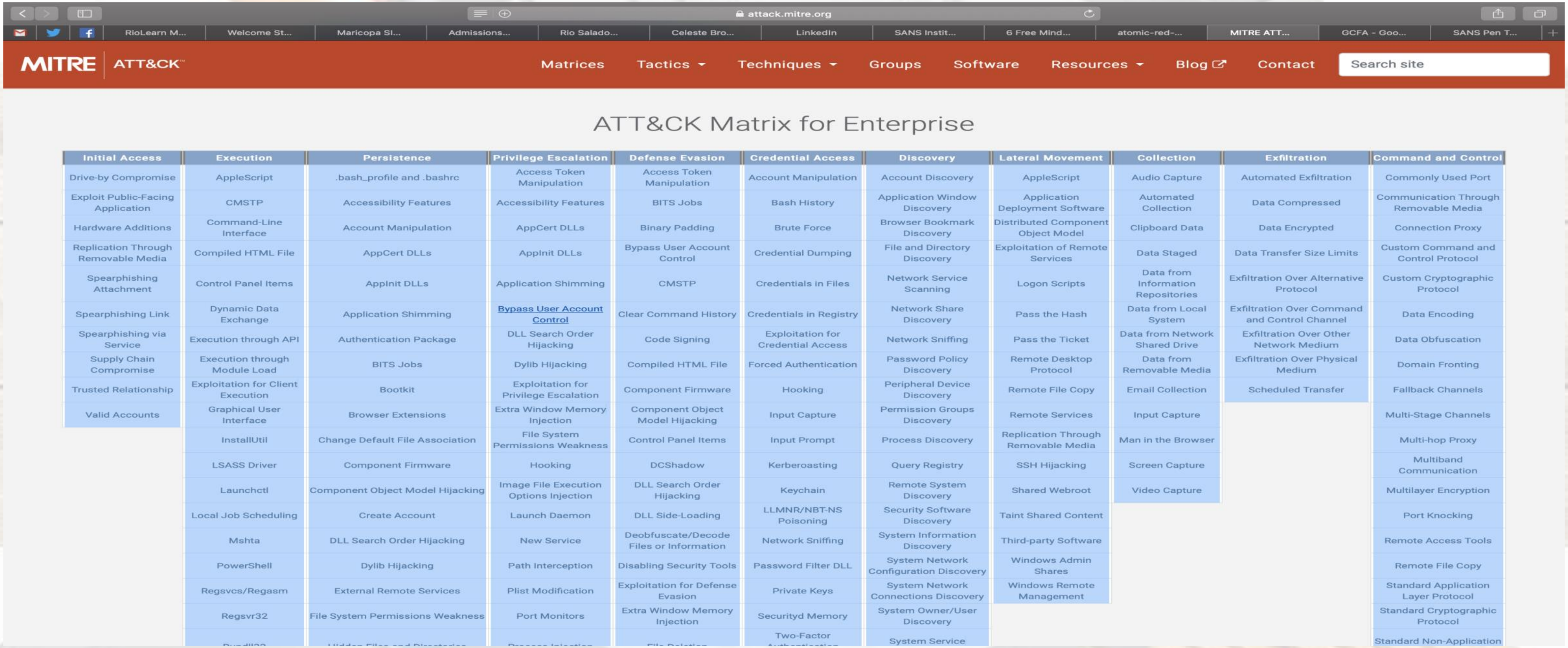
- Tests laid out for
  - Persistence
  - Defense-Evasion
  - Privilege escalation
  - Discovery
  - Credential access
  - Execution
  - Lateral Movement
  - Collection
  - Exfiltration
  - Command and Control
  - Initial Access



Atomic Red Example

- T1050 New Service – Installs A Local Service
  - Atomic Test #1: Service Installation [windows]
- **Run it with command_prompt!**
- sc.exe create #{service_name} binPath= #{binary_path}
- sc.exe start #{service_name}
- sc.exe stop #{service_name}
- sc.exe delete #{service_name}
  - Atomic Test #2 - Service Installation PowerShell Installs A Local Service using PowerShell
- New-Service -Name "#{service_name}" -BinaryPathName "#{binary_path}"
- Start-Service -Name "#{service_name}"
- Stop-Service -Name "#{service_name}"
- (Get-WmiObject Win32_Service -filter "name='#{service_name}'").Delete()

MITRE ATT&CK FRAMEWORK



ATT&CK Matrix for Enterprise

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command and Control
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Automated Exfiltration	Commonly Used Port
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	BITS Jobs	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Data Compressed	Communication Through Removable Media
Hardware Additions	Command-Line Interface	Account Manipulation	AppCert DLLs	Binary Padding	Brute Force	Browser Bookmark Discovery	Distributed Component Object Model	Clipboard Data	Data Encrypted	Connection Proxy
Replication Through Removable Media	Compiled HTML File	AppCert DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	File and Directory Discovery	Exploitation of Remote Services	Data Staged	Data Transfer Size Limits	Custom Command and Control Protocol
Spearphishing Attachment	Control Panel Items	AppInit DLLs	Application Shimming	CMSTP	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Information Repositories	Exfiltration Over Alternative Protocol	Custom Cryptographic Protocol
Spearphishing Link	Dynamic Data Exchange	Application Shimming	Bypass User Account Control	Clear Command History	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Local System	Exfiltration Over Command and Control Channel	Data Encoding
Spearphishing via Service	Execution through API	Authentication Package	DLL Search Order Hijacking	Code Signing	Exploitation for Credential Access	Network Sniffing	Pass the Ticket	Data from Network Shared Drive	Exfiltration Over Other Network Medium	Data Obfuscation
Supply Chain Compromise	Execution through Module Load	BITS Jobs	Dylib Hijacking	Compiled HTML File	Forced Authentication	Password Policy Discovery	Remote Desktop Protocol	Data from Removable Media	Exfiltration Over Physical Medium	Domain Fronting
Trusted Relationship	Exploitation for Client Execution	Bootkit	Exploitation for Privilege Escalation	Component Firmware	Hooking	Peripheral Device Discovery	Remote File Copy	Email Collection	Scheduled Transfer	Fallback Channels
Valid Accounts	Graphical User Interface	Browser Extensions	Extra Window Memory Injection	Component Object Model Hijacking	Input Capture	Permission Groups Discovery	Remote Services	Input Capture		Multi-Stage Channels
	InstallUtil	Change Default File Association	File System Permissions Weakness	Control Panel Items	Input Prompt	Process Discovery	Replication Through Removable Media	Man in the Browser		Multi-hop Proxy
	LSASS Driver	Component Firmware	Hooking	DCShadow	Kerberoasting	Query Registry	SSH Hijacking	Screen Capture		Multiband Communication
	Launchctl	Component Object Model Hijacking	Image File Execution Options Injection	DLL Search Order Hijacking	Keychain	Remote System Discovery	Shared Webroot	Video Capture		Multilayer Encryption
	Local Job Scheduling	Create Account	Launch Daemon	DLL Side-Loading	LLMNR/NBT-NS Poisoning	Security Software Discovery	Taint Shared Content			Port Knocking
	Mshta	DLL Search Order Hijacking	New Service	Deobfuscate/Decode Files or Information	Network Sniffing	System Information Discovery	Third-party Software			Remote Access Tools
	PowerShell	Dylib Hijacking	Path Interception	Disabling Security Tools	Password Filter DLL	System Network Configuration Discovery	Windows Admin Shares			Remote File Copy
	Regsvcs/Regasm	External Remote Services	Plist Modification	Exploitation for Defense Evasion	Private Keys	System Network Connections Discovery	Windows Remote Management			Standard Application Layer Protocol
	Regsvr32	File System Permissions Weakness	Port Monitors	Extra Window Memory Injection	Securityd Memory	System Owner/User Discovery				Standard Cryptographic Protocol
	Rundll32	Hidden Files and Directories	Process Injection	File Deletion	Two-Factor Authentication	System Service				Standard Non-Application

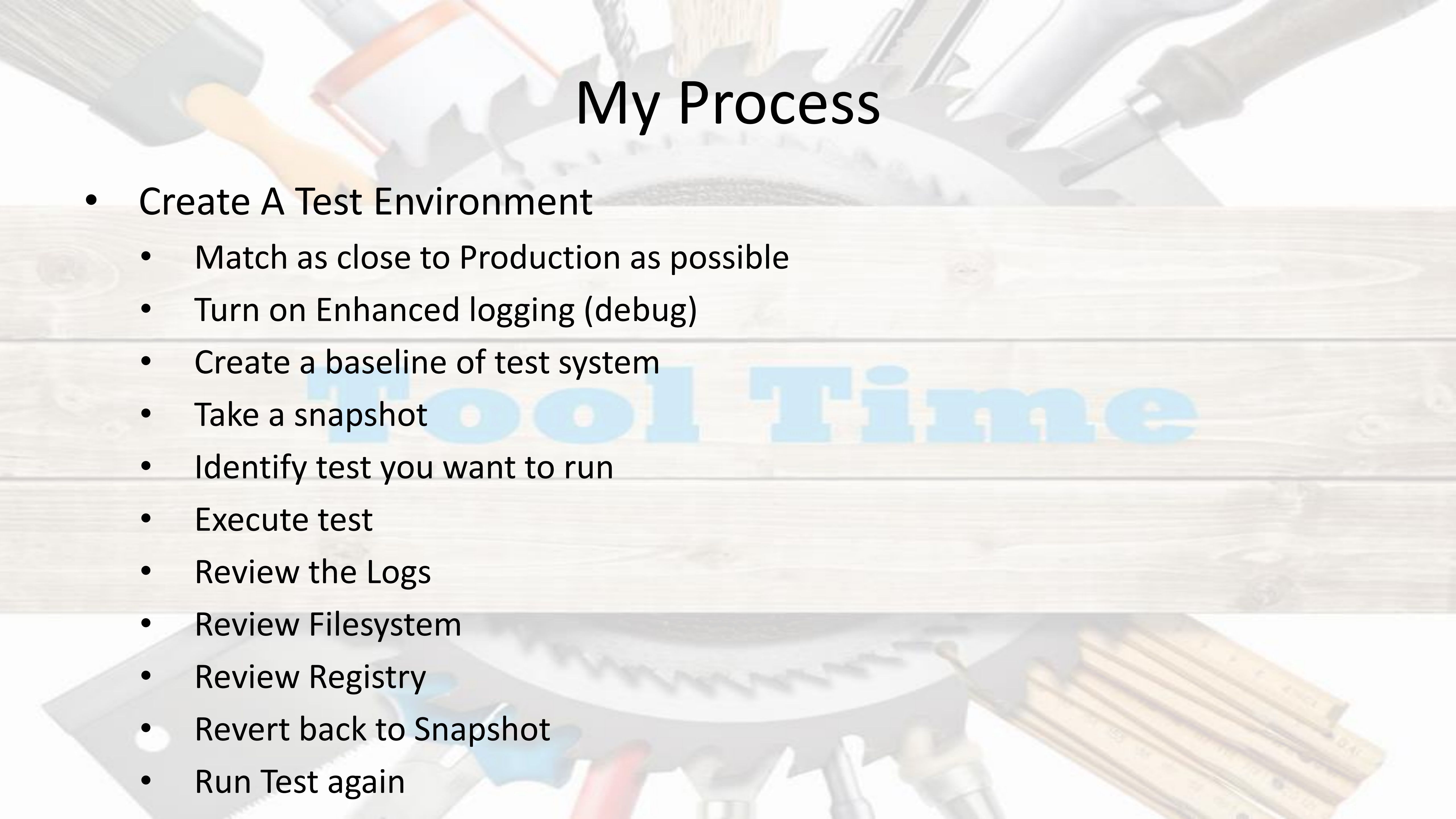
Books and Online Resources

- **Good books** (focus on setting up a lab and attacking it)
- *Penetration Testing: A Hands-On Introduction to Hacking*
- *The Hacker Playbook 2: Practical Guide to Penetration Testing*
- *Metasploit: The Penetration Tester’s Guide*
- **Online Resources**
- Cybrary.it
- SANS Penetration Testing Blog
- YOUTUBE
- VulnHub
- Offensive Security Red Team Experiments

Additional Online Resources

- https://www.kali.org/category/tutorials/
- Metasploit Unleashed
- https://www.pentesteracademy.com
- http://overthewire.org/wargames/
- https://www.amanhardikar.com/mindmaps/Practice.html
- Cobalt Strike

My Process



- Create A Test Environment
  - Match as close to Production as possible
  - Turn on Enhanced logging (debug)
  - Create a baseline of test system
  - Take a snapshot
  - Identify test you want to run
  - Execute test
  - Review the Logs
  - Review Filesystem
  - Review Registry
  - Revert back to Snapshot
  - Run Test again

Analysis Time

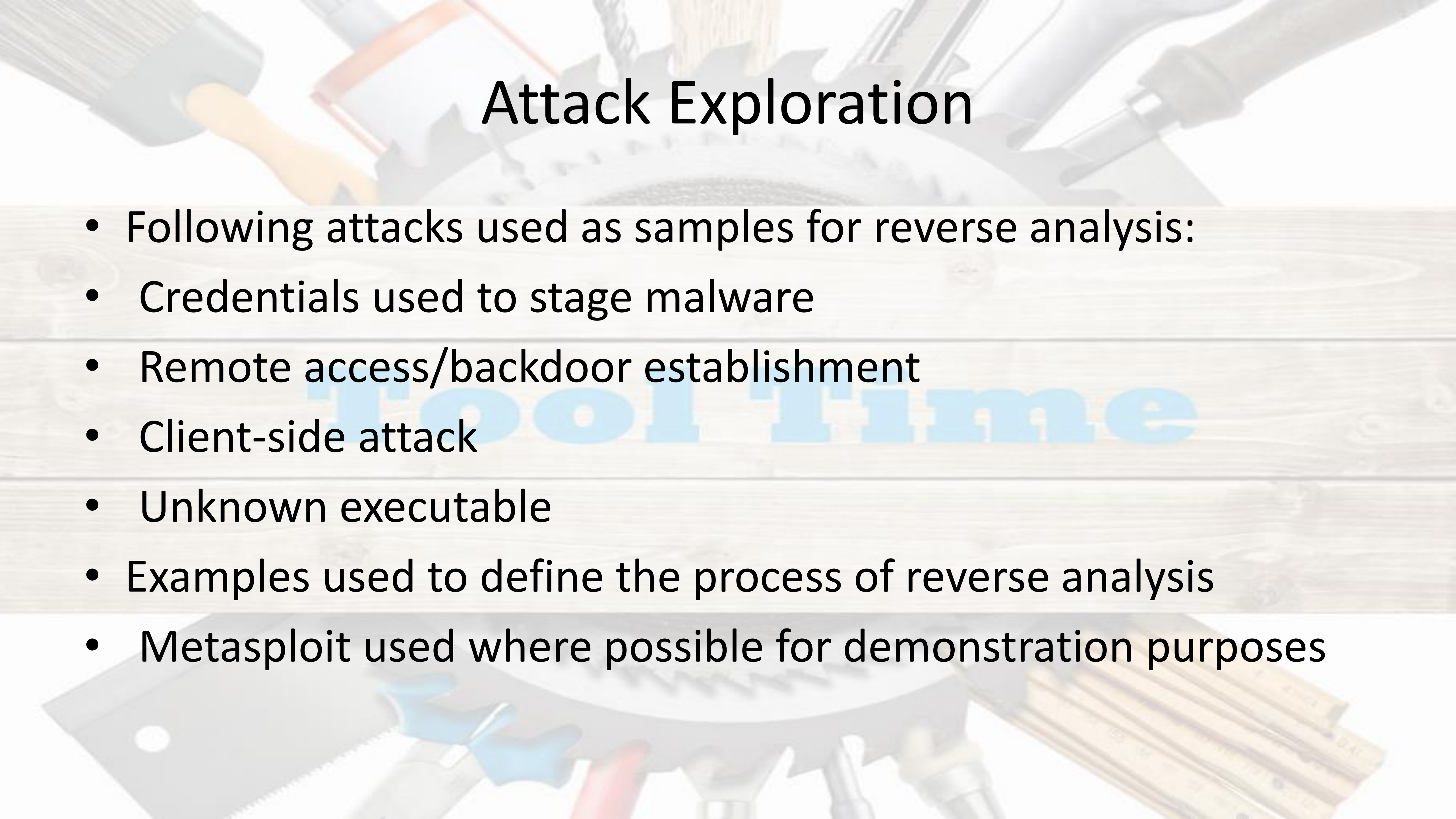
- Next section covers a few example attacks
- Used to walk through the process of reverse analysis
- Possible to identify more events of interest
- Examples explain the attack and then results of analysis
- Specific detects found may not match every environment

"Never theorize before you have data. Invariably, you end up twisting facts to suit theories instead of theories to suit facts."

~ Sir Arthur Conan Doyle

as Sherlock Homes

Attack Exploration



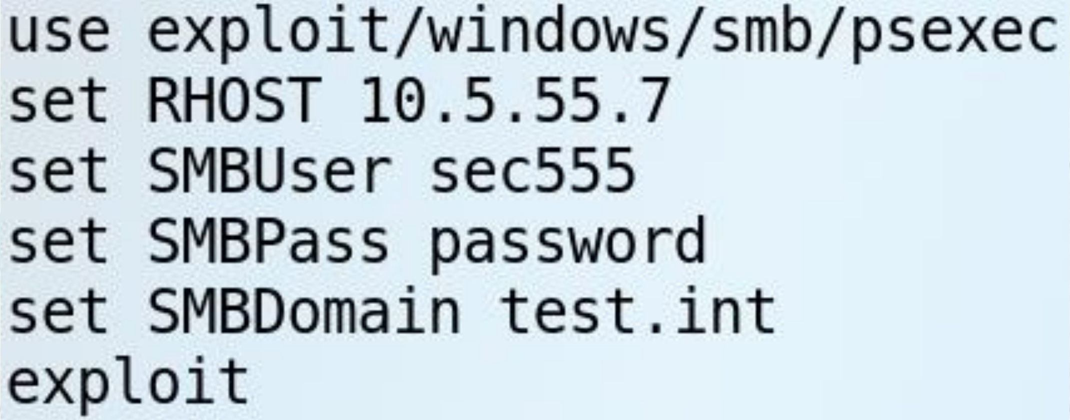
- Following attacks used as samples for reverse analysis:
- Credentials used to stage malware
- Remote access/backdoor establishment
- Client-side attack
- Unknown executable
- Examples used to define the process of reverse analysis
- Metasploit used where possible for demonstration purposes

psexec

- One of the most common Metasploit modules is psexec
- Uses legitimate credentials to log in to systems
- Again, the attack is not an exploit, it is a login …
- But how can you catch a normal login using credentials?
- Point of exercise is things may not be "normal"
- Test performed with two accounts: jhenderson and sec555
- jhenderson used only with traditional login methods
- sec555 used with Metasploit psexec module

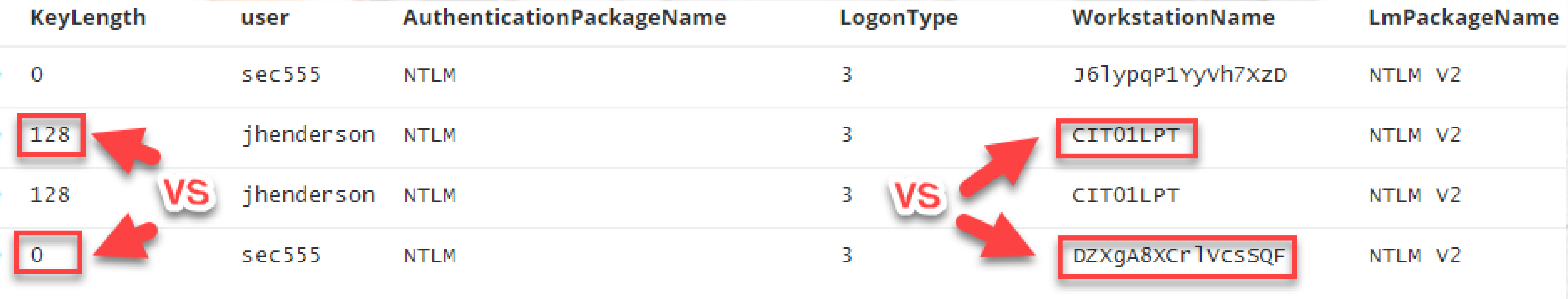
Logon Methods

- jhenderson account used with following methods:
- Remote Desktop login
- Local console login
- File share access
- PowerShell remote access
- sec555 account used with psexec
- Different user account used to further distinguish between logons



```
use exploit/windows/smb/psexec
set RHOST 10.5.55.7
set SMBUser sec555
set SMBPass password
set SMBDomain test.int
exploit
```

Normal vs. psexec

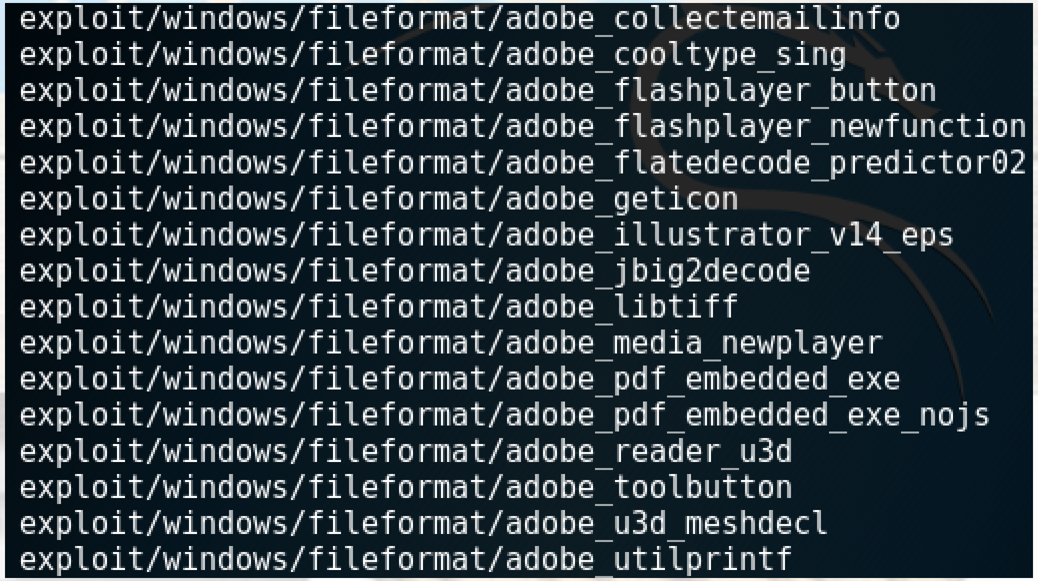


KeyLength	user	AuthenticationPackageName	LogonType	WorkstationName	LmPackageName
0	sec555	NTLM	3	J6lypqP1Yyvh7XzD	NTLM V2
128	jhenderson	NTLM	3	CIT01LPT	NTLM V2
128	jhenderson	NTLM	3	CIT01LPT	NTLM V2
0	sec555	NTLM	3	DZXgA8XCrlVcsSQF	NTLM V2

- Logs show two possible discrepancies
- KeyLength used is 0 for psexec and 128 for regular login
- Workstation name is clearly random for psexec
- Verification requires comparing key lengths for NTLM v2 over a longer period of time

Evil Files

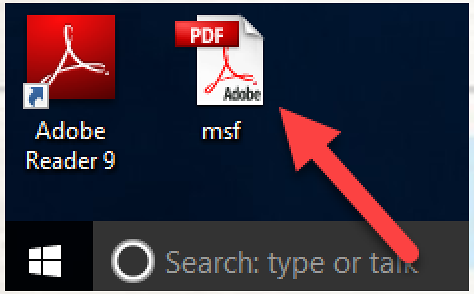
- Most common attack targets today are client-side
- Involves PDFs, Word Documents, Java, Flash
- Metasploit has multiple examples available to test with:
- Auto-generate evil files
- Sets up listener
- Waits for the user to open file
- Evil PDF examples ->
- What happens when opened?

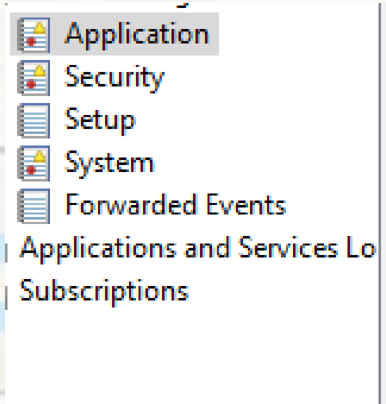


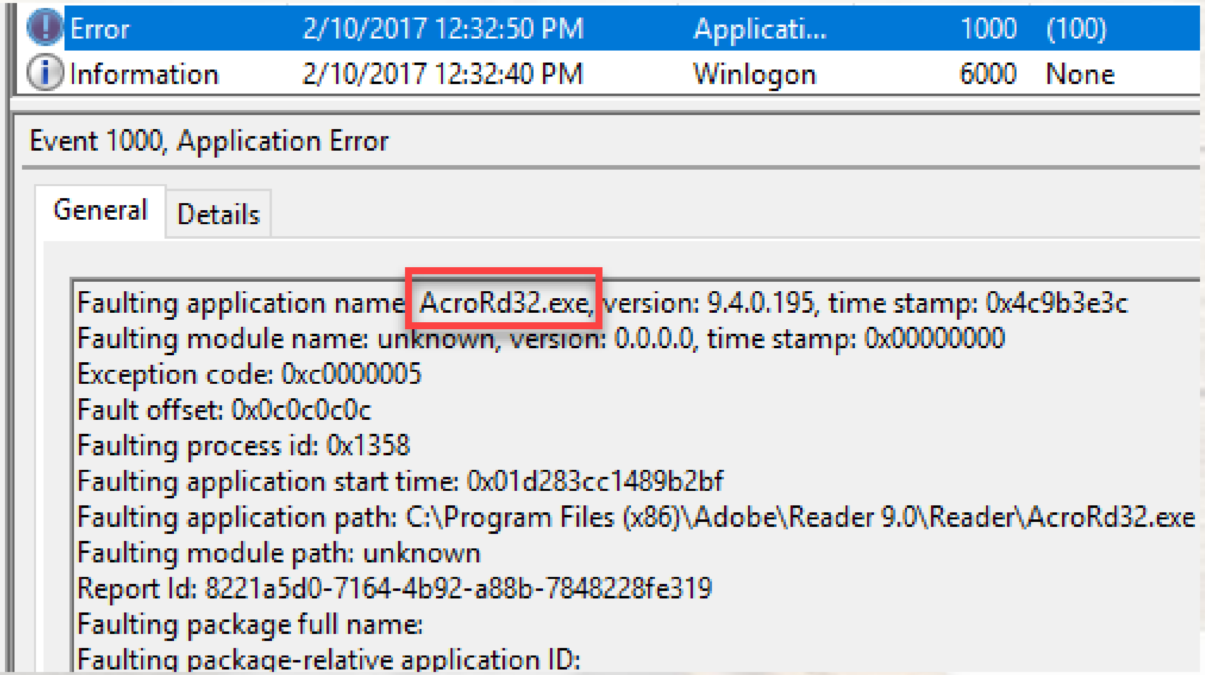
```
exploit/windows/fileformat/adobe_collectemailinfo
exploit/windows/fileformat/adobe_cooltype_sing
exploit/windows/fileformat/adobe_flashplayer_button
exploit/windows/fileformat/adobe_flashplayer_newfunction
exploit/windows/fileformat/adobe_flatedecode_predictor02
exploit/windows/fileformat/adobe_geticon
exploit/windows/fileformat/adobe_illustrator_v14_eps
exploit/windows/fileformat/adobe_jbig2decode
exploit/windows/fileformat/adobe_libtiff
exploit/windows/fileformat/adobe_media_newplayer
exploit/windows/fileformat/adobe_pdf_embedded_exe
exploit/windows/fileformat/adobe_pdf_embedded_exe_nojs
exploit/windows/fileformat/adobe_reader_u3d
exploit/windows/fileformat/adobe_toolbutton
exploit/windows/fileformat/adobe_u3d_meshdecl
exploit/windows/fileformat/adobe_utilprintf
```

Application Crash Example

- User opens evil PDF, app crashes, and system is compromised

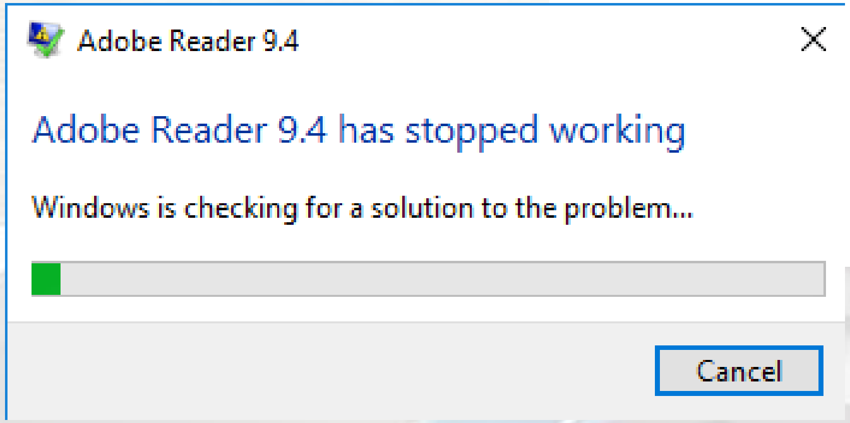





```
Error        2/10/2017 12:32:50 PM   Applicati...   1000   (100)
Information  2/10/2017 12:32:40 PM   Winlogon       6000   None
```

Event 1000, Application Error

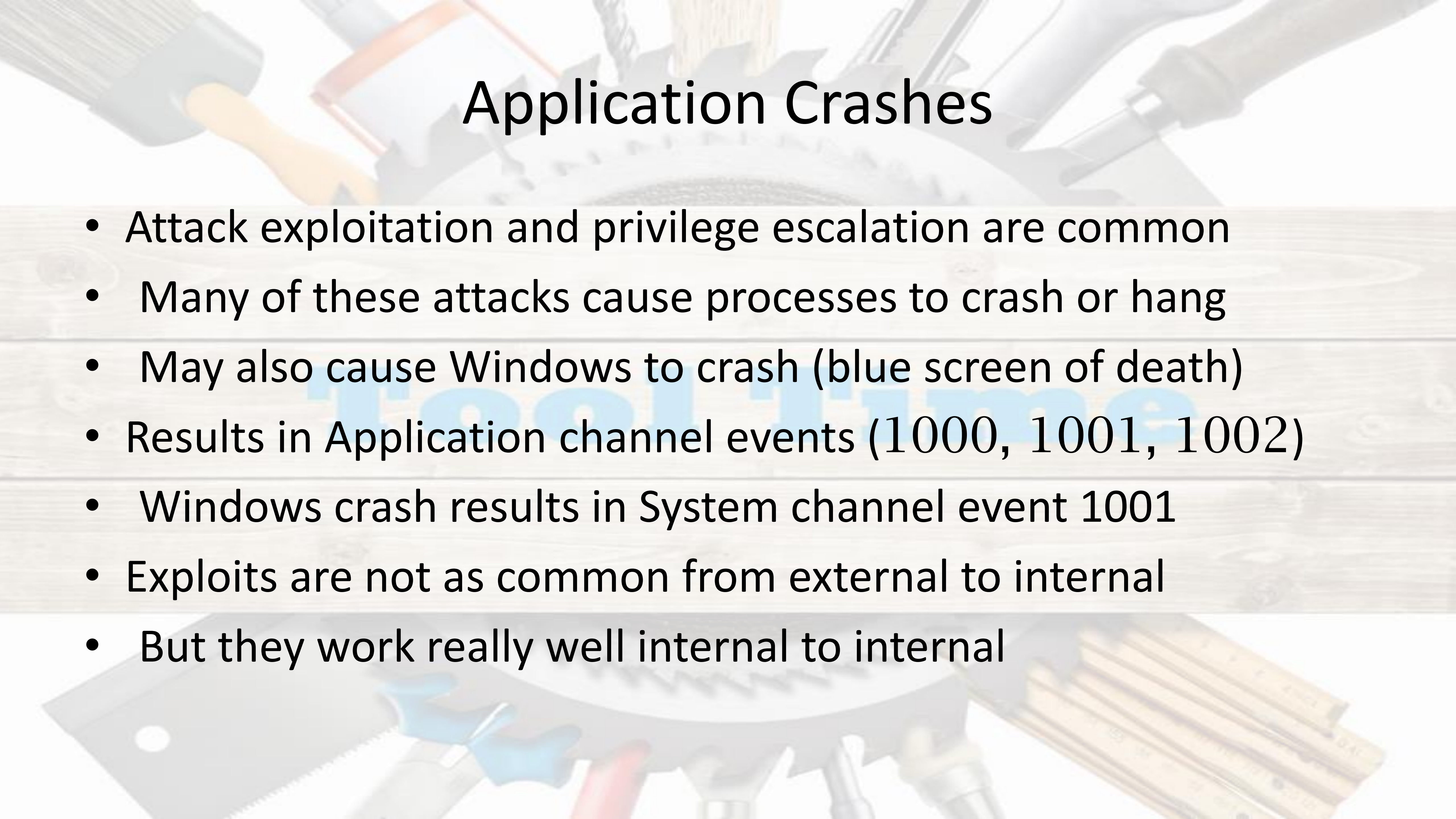
```
Faulting application name: AcroRd32.exe, version: 9.4.0.195, time stamp: 0x4c9b3e3c
Faulting module name: unknown, version: 0.0.0.0, time stamp: 0x00000000
Exception code: 0xc0000005
Fault offset: 0x0c0c0c0c
Faulting process id: 0x1358
Faulting application start time: 0x01d283cc1489b2bf
Faulting application path: C:\Program Files (x86)\Adobe\Reader 9.0\Reader\AcroRd32.exe
Faulting module path: unknown
Report Id: 8221a5d0-7164-4b92-a88b-7848228fe319
Faulting package full name:
Faulting package-relative application ID:
```


Adobe Reader 9.4

Adobe Reader 9.4 has stopped working

Windows is checking for a solution to the problem...

Application Crashes



- Attack exploitation and privilege escalation are common
- Many of these attacks cause processes to crash or hang
- May also cause Windows to crash (blue screen of death)
- Results in Application channel events (1000, 1001, 1002)
- Windows crash results in System channel event 1001
- Exploits are not as common from external to internal
- But they work really well internal to internal

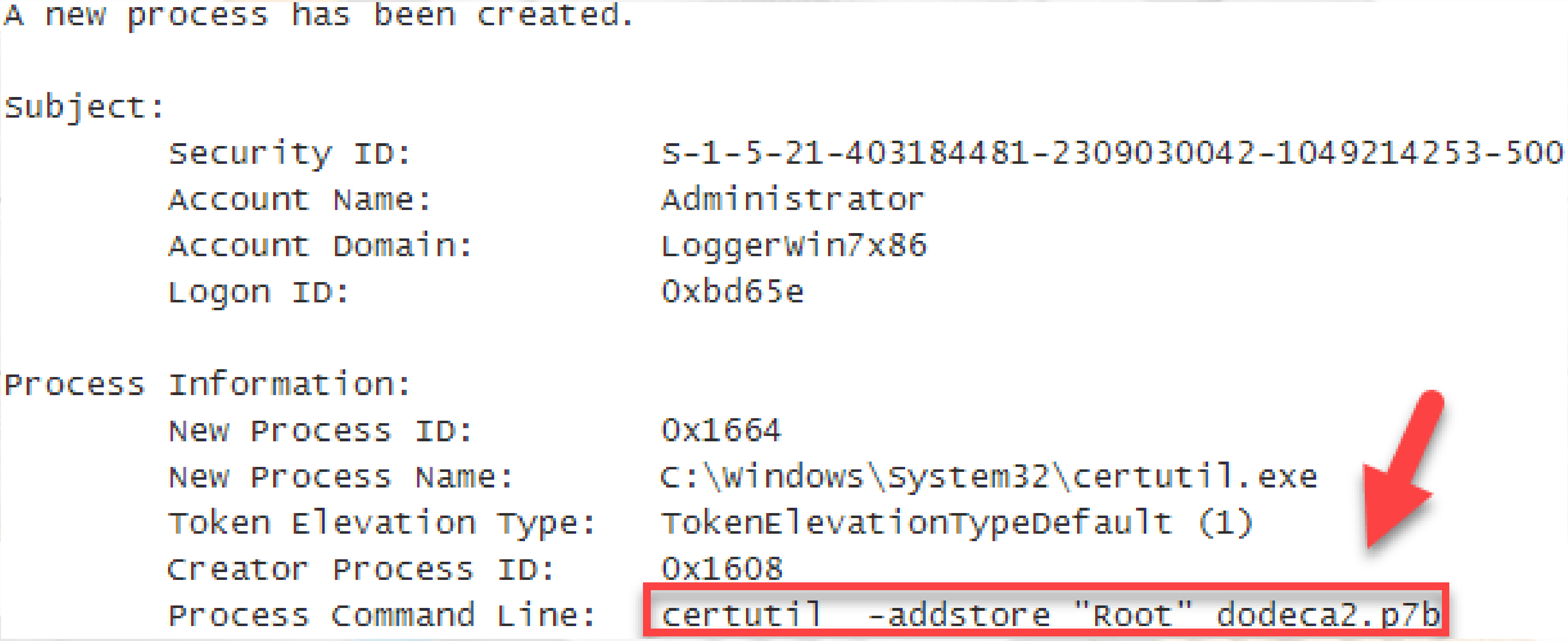
Malware Analysis

- Reverse analysis works with malware analysis at scale
- Use known or unknown samples
- The depth of analysis != malware analysis or forensics
- The purpose is to build tactical alerts and know thyself
- SIEM is used to perform high-level analysis
- Creates alert capabilities by finding things outside norm
- High-level analysis at scale yet can produce value

Unknown Specimen

- Example: PandorasBox.exe
- Not sure what it does yet user wants to click it
- Ran through Cuckoo Sandbox with logs going to SIEM
- Certificate installation via certutil is discovered
- May be legitimate but likely malicious
- The analysis makes you stop and think:
  - Should CA installation events be monitored?
  - Or is this likely to generate false positives?

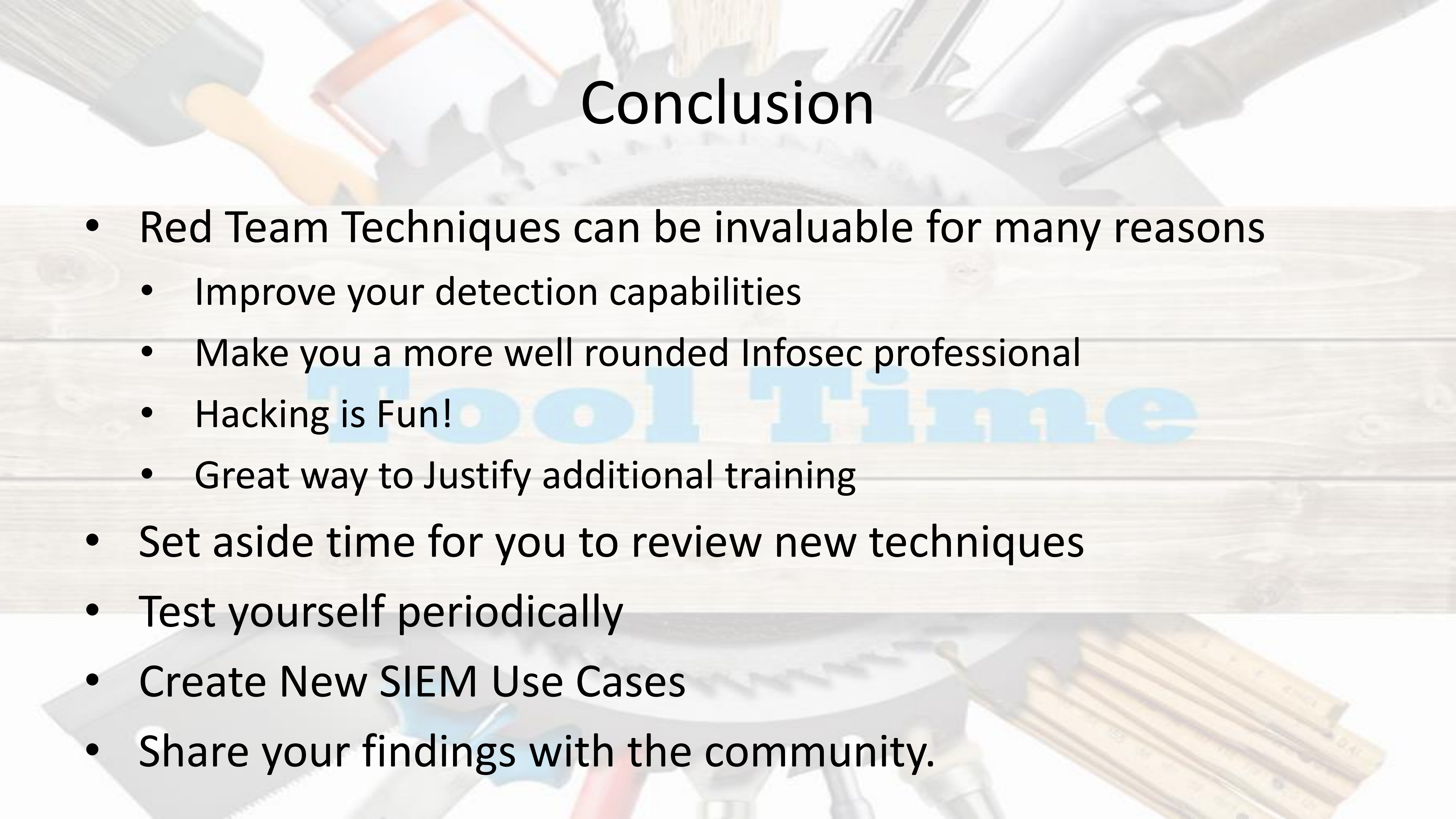
Event ID 4688



```
A new process has been created.

Subject:
        Security ID:            S-1-5-21-403184481-2309030042-1049214253-500
        Account Name:           Administrator
        Account Domain:         LoggerWin7x86
        Logon ID:               0xbd65e

Process Information:
        New Process ID:         0x1664
        New Process Name:       C:\Windows\System32\certutil.exe
        Token Elevation Type:   TokenElevationTypeDefault (1)
        Creator Process ID:     0x1608
        Process Command Line:   certutil  -addstore "Root" dodeca2.p7b
```



Conclusion

- Red Team Techniques can be invaluable for many reasons
  - Improve your detection capabilities
  - Make you a more well rounded Infosec professional
  - Hacking is Fun!
  - Great way to Justify additional training
- Set aside time for you to review new techniques
- Test yourself periodically
- Create New SIEM Use Cases
- Share your findings with the community.