CSI: Evidence Indicators for Targeted Ransomware Attacks – Part II

mcafee.com/blogs/other-blogs/mcafee-labs/csi-evidence-indicators-for-targeted-ransomware-attacks-part-ii/

February 20, 2020

In our first article we discussed the growing pattern of targeted ransomware attacks where the first infection stage is often an info-stealer kind of malware used to gain credentials/access to determine if the target would be valuable for a ransomware attack. In this second part we will pick up where we left off: the attacker has a foothold on the network by controlling an infected host or has a valid account to access a remote service.

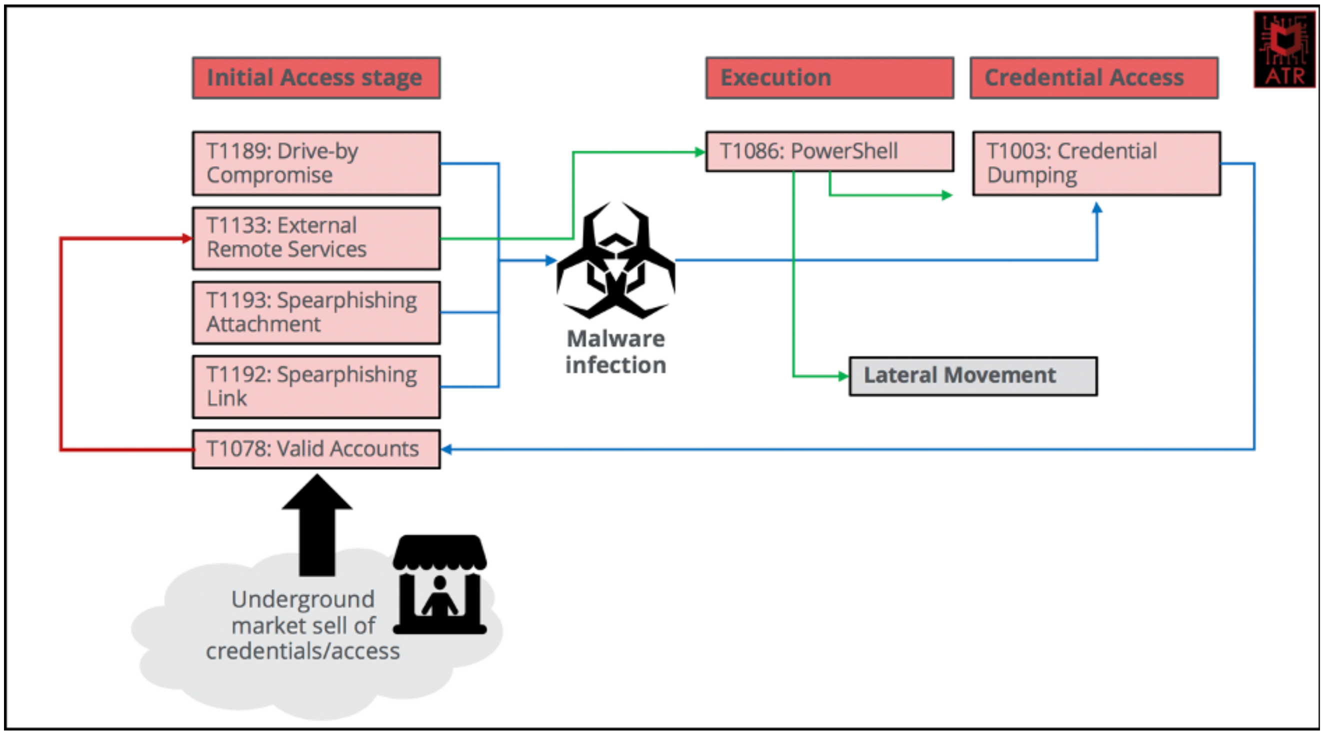

Initial Access stage
Execution
Credential Access
T1189: Drive-by Compromise
T1133: External Remote Services
T1193: Spearphishing Attachment
T1192: Spearphishing Link
T1078: Valid Accounts
T1086: PowerShell
T1003: Credential Dumping
Malware infection
Lateral Movement
Underground market sell of credentials/access
ATR

Figure 1 Adversary has a beachhead

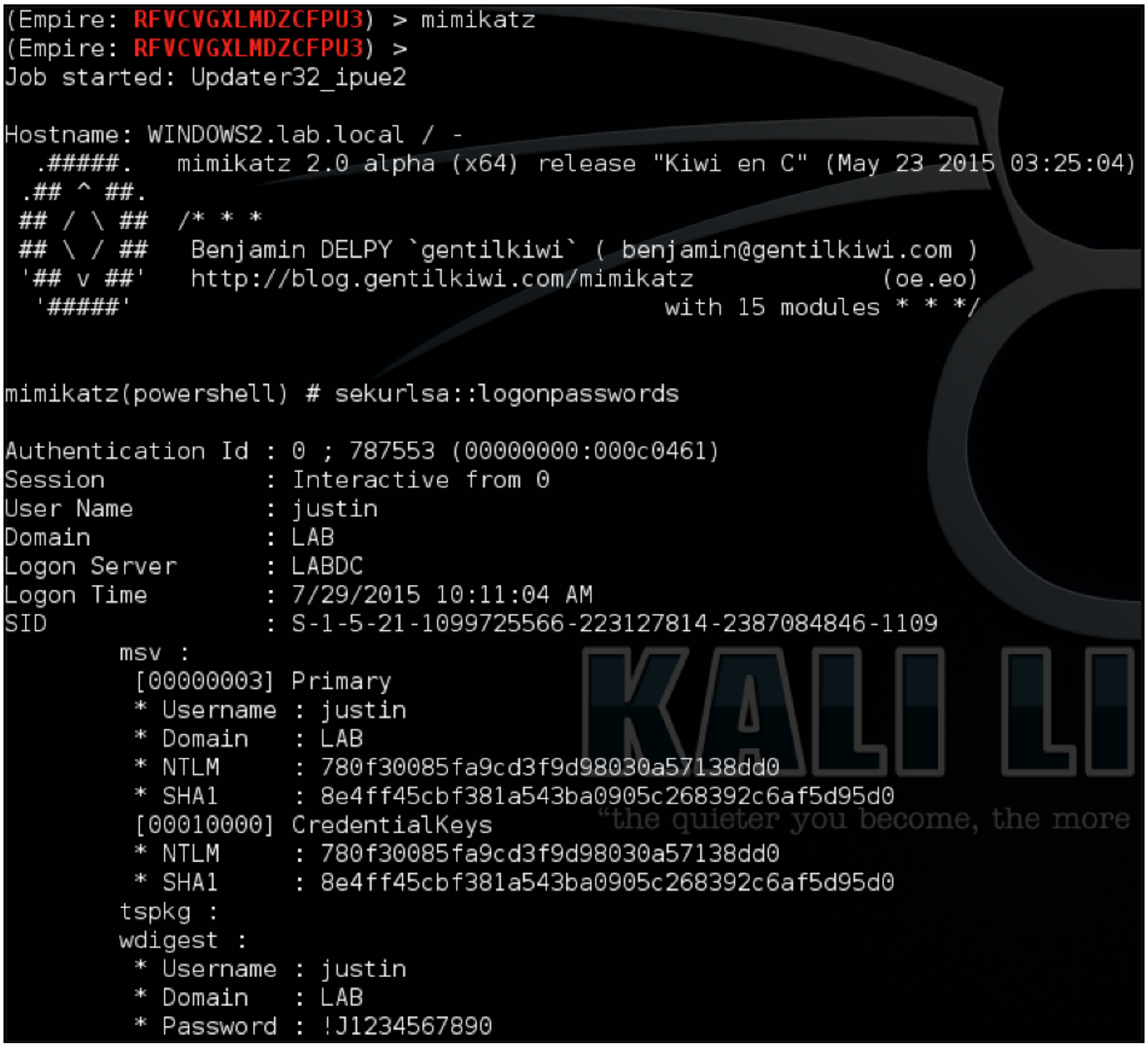
With either a valid account or having access towards a system in a company, the first two things you want to figure out are:

1. What kind of rights do I have from this machine?
2. Where the heck am I in this network?

One of the first commands you would observe as a responder is "whoami/all". The output of this command will give the details of the account the attacker has on the machine with regards to group/privileges. A great way to detect suspicious activity in your network is to setup a detection rule for the "whoami" command and assign it to the assets in use by executives or holders of key positions in the company. There might always be a techie executive in the company but most of them will never use command or use a command-line.

In the context of the targeted ransomware attacks, the attacker preferably wants to have local-admin/domain-admin and or system rights. Those will be the keys to the kingdom and open all gates.

In the next step, mostly observe some variant from Mimikatz or other password-dumping tools that will dump the credentials from a machine. Either the tool Mimikatz is compiled in various formats or it is part of a remote PowerShell toolkit like Empire:



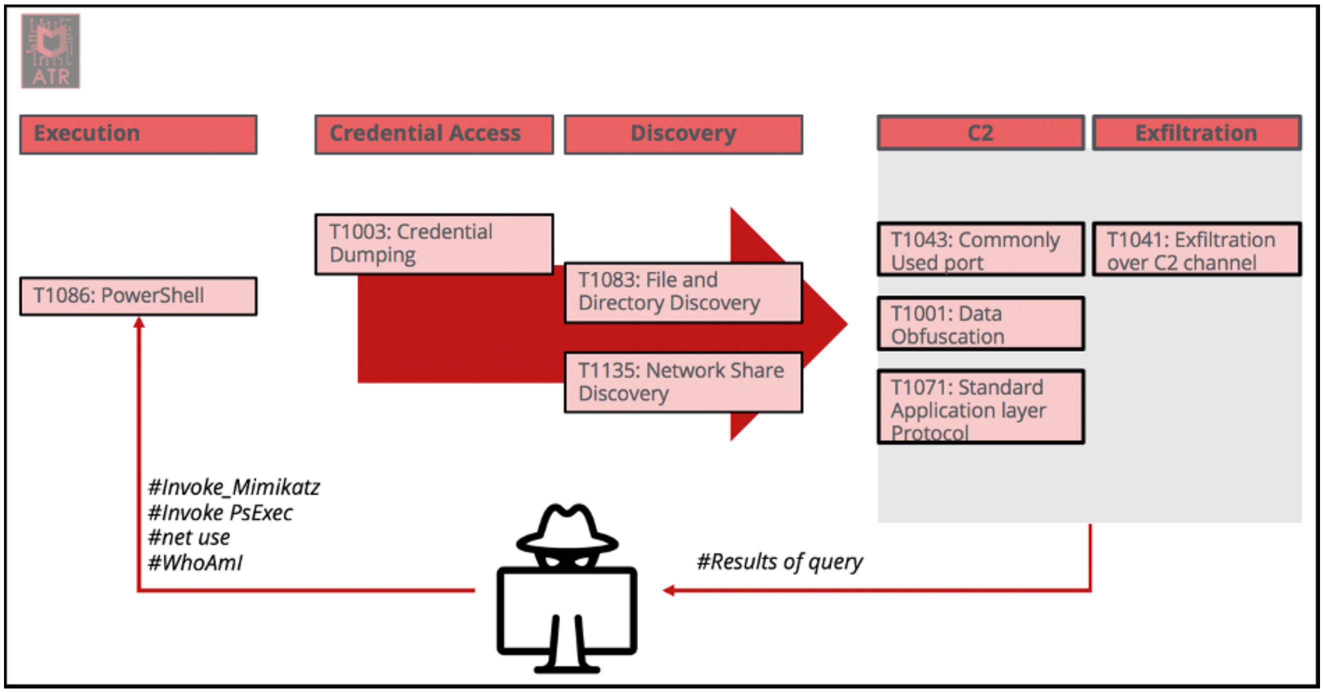
```
(Empire: RFVCVGXLMDZCFPU3) > mimikatz
(Empire: RFVCVGXLMDZCFPU3) >
Job started: Updater32_ipue2

Hostname: WINDOWS2.lab.local / -
  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (May 23 2015 03:25:04)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                    with 15 modules * * */


mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 787553 (00000000:000c0461)
Session           : Interactive from 0
User Name         : justin
Domain            : LAB
Logon Server      : LABDC
Logon Time        : 7/29/2015 10:11:04 AM
SID               : S-1-5-21-1099725566-223127814-2387084846-1109
        msv :
         [00000003] Primary
         * Username : justin
         * Domain   : LAB
         * NTLM     : 780f30085fa9cd3f9d98030a57138dd0
         * SHA1     : 8e4ff45cbf381a543ba0905c268392c6af5d95d0
         [00010000] CredentialKeys
         * NTLM     : 780f30085fa9cd3f9d98030a57138dd0
         * SHA1     : 8e4ff45cbf381a543ba0905c268392c6af5d95d0
        tspkg :
        wdigest :
         * Username : justin
         * Domain   : LAB
         * Password : !J1234567890
```

Figure 2 Empire Mimikatz (source powershellempire.com)

We began this article series with the fact that every touch of a system will leave digital footprints behind, in this example with Empire, the following is happening:

- Attacker has a connection to a system through usage of a 'launch script'
- Attacker is using a C2 (command and control) to interact
- Attacker is using PowerShell to execute commands

From a MITRE ATT&CK Techniques perspective we would look at this overview of techniques and interaction:

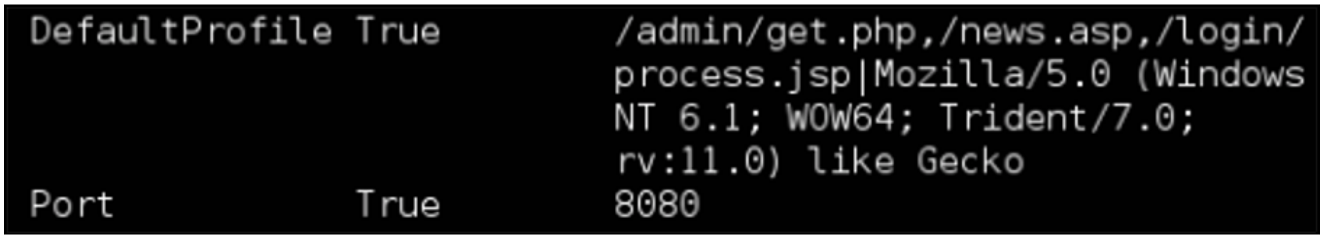

ATR
Execution
Credential Access
Discovery
C2
Exfiltration
T1086: PowerShell
T1003: Credential Dumping
T1083: File and Directory Discovery
T1135: Network Share Discovery
T1043: Commonly Used port
T1001: Data Obfuscation
T1071: Standard Application layer Protocol
T1041: Exfiltration over C2 channel
#Invoke_Mimikatz
#Invoke PsExec
#net use
#WhoAmI
#Results of query
Figure 3 Attacker using Empire
Which stages and evidence sources will create visibility and early warning indicators?

For PowerShell usage there are several digital evidence locations available. Besides traces in memory, there can be traces discovered in the Windows Event logs, the Registry, on the file-system like the Prefetch directory and there's even a PowerShell command history file available if a version of PowerShell v5 and above is used. That location can be found at:

C:\Users\<username>\AppData\Roaming\Microsoft\Windows PowerShell\PSReadline\ConsoleHost_history.txt

Empire by design is encrypting its communications, makes infrequent and randomized connections and, last but not least, is mirroring HTTP activity to stay hidden in the 'normal' traffic. If we look at the order of volatility from our first article, network traffic will change from seconds to minutes, giving us a very short window unless we do full packet capture and inspection of our network traffic.

The network activity might be challenging to notice, however there are possible indicators that can help possible Empire traffic. To setup the C2 traffic, the attacker needs to configure a 'listener' that contains all the settings for the C2 traffic interaction.



```
DefaultProfile True        /admin/get.php,/news.asp,/login/
                           process.jsp|Mozilla/5.0 (Windows
                           NT 6.1; WOW64; Trident/7.0;
                           rv:11.0) like Gecko
Port           True        8080
```

Figure 4 Empire Listener setup

In the partial setup screen in Figure 4, we see three URI's being configured that will be frequently requested and combined with an HTTP Header. A combination detection of the three URI's polled within x timeframe combined with the HTTP header could be a very good network-indicator to detect Empire C2 traffic. As a responder, when you detect these indicators, look back at Figure 3 and start hunting backwards. You discovered the C2 activity, but it means the hosts who are interacting with the C2 have PowerShell activity and time to secure your evidence. This is how I like to apply the MITRE ATT&CK model: understand what you just discovered, and the implications then hunt for evidence or anticipate the attacker's next step and act.

Keep in mind this is a default setup and can be changed by the attacker. Experience suggests that cyber-crime motivated actors are in a hurry for the cash and use a default installation of tools, whereas nation-state cyber-espionage operations are customizing since they want to operate longer in a victim's network.

Now back to our scenario. The attacker has acquired the right privileges and has laterally moved through the network to have an idea how large it is and have possibly identified critical assets — they love shared folders with a lot of data in them.

The attacker will next prepare a custom version of the ransomware and could have tested it on underground AV-testing services to make sure that it will be fully undetectable (FUD). Depending on skills and capabilities, the ransomware will either be uploaded to the victim's network or using scripts distributed and executed over the network. We have observed cases where a series of .bat scripts were used and where the payload was downloaded from Pastebin and executed by PowerShell on hosts – enough variety and options. Once executed, the first calls will come into the Helpdesk reporting the ransom notes.

A lot of the above techniques are available within post exploitation frameworks, of which there are several options that attackers can choose from. With any of these, the tool is only a conduit and we should always focus on detecting techniques rather than implementation specifics. Available options come in the form of both underground toolkits and penetration test tools designed to build awareness around the attacks. Unfortunately, those designed as security tools may also be used for nefarious purposes. This 'chicken and egg' situation is often the cause of heated debate since on one hand the tools can streamline attack scenarios, while on the other, without having them available to defenders, adequate testing of precautions is impossible.

Empire was mentioned above. This excellent PowerShell-based post-exploitation framework is no longer maintained or supported by its authors. Although forks exist, traction has slowed but its use is still detected. Cobalt Strike, a commercial adversary simulation platform regularly used by red teams to test infrastructural security measures and detection capacity, is increasingly being adopted by criminal actors." Although its license is strictly controlled, pirated and cracked trial versions are available in the criminal underworld. Cobalt Strike is updated regularly to include the latest techniques and tools such as Mimikatz, while allowing a lot of flexibility for addition of new and unique attack and bypass techniques. This can increase the likelihood of success, even on the most recent operating system versions.

Threat groups distributing GoGalocker, MegaCortex and Maze ransomware have been observed utilizing Cobalt Strike. At this stage in our scenario, with a foothold on the network, Cobalt Strike provides many options which can be used to complete their objective. These include T1075 Pass the Hash, T1097 Pass the Ticket, T1105 Remote File Copy, T1021 Remote Services and the old reliable: T1077 Windows Admin Shares.

As detailed in part 1, this evidence remains on a system for differing time spans. By studying previous attacks, we can see a general mode of operation and a sequence in which these techniques are executed. This can help guide us when choosing the timing and methods of evidence collection.

Detection of these attacks is not an easy task. Exploitation frameworks are often open source, in which case the attacker can modify code to manipulate IOC's (indicators of compromise). Alternatively, and as is the case for our example, Cobalt Strike, the framework will provide specific configuration around the final form of binary droppers, network traffic format, timing and specific parameters, etc. Therefore, generic signatures will only detect the most basic of attacks and, as previously mentioned, focusing on techniques and behaviors rather than tools becomes critical. Creating a Cobalt Strike generic detection may miss the point of its existence – education around malicious techniques and behaviors. However, we can learn a lot about various indicator categories that can be monitored for functional outliers and unusual behaviors and, where the tool is merged into malware strains, the markers for that specific instance can often become specific IOC's.

In a lot of cases, understanding normal system behavior is crucial for uncovering outliers and having a baseline system for comparison can aid the analysis.

A few examples of these include:

- Memory
    - The holy grail. Binary droppers may be modified to mask their true intentions, however in memory they can be visible in a decoded format.
    - Search for markers of functionality such as suspicious system calls or reads/writes of sensitive system locations (e.g. lsass memory).
- Running Processes
    - Similarly, running processes can be very revealing.
    - Although process 'migration' is often carried out, review for abnormal process hierarchies.
    - Does the process carry out functionality that is not expected? E.g. does notepad make http requests to an external domain?

- Network Traffic
    - Domains should match known-good.
    - Spikes or regular timing patterns in traffic type or to specific servers.
    - Variables used in unusual manner, e.g. http headers including encoded binary data.
- Disk
    - Binary and configuration files available for reverse engineering.
    - This static analysis is not as powerful as dynamic, running software.
    - Debugging a malicious binary may help but beware of sandbox/debugger detection.
- Backup / Log Files
    - Never fully trust log files as attackers can easily falsify. External sources, e.g. syslog going to an external db, can improve the reliability of log data if available.
    - Search for specific markers for malicious activity. Some useful items to include: command line logs, PowerShell script block logging, specific logs for network applications such as web servers, service installation, share accessed – particularly C$, process launch, account login.

As we highlighted in the two articles, targeted ransomware attacks have increased massively over the past 8 months. Targeting MSPs to hit many at the same time, victims running critical operations, public services, etc. Many of them are all using a similar blueprint as we tried to highlight.

Learn from the articles, identify which technology can give you that visibility, what digital evidence sources do you have, and can you detect fast enough to preserve and respond? If the ‘initial access stage’ is passed, where can you pick up in your line of defense and stop the threat?

Stay tuned for part 3, where we discuss technical controls that are available to help your organization to react to these early warning signs within an acceptable timeframe.

Christiaan Beek Lead Scientist & Sr. Principal Engineer
Christiaan Beek is the Lead Scientist & Sr. Principal Engineer of the Enterprise Office of the CTO. He is leading the strategic threat intelligence research with a focus on inventing...