CYBER
THREAT
ANALYSIS

Recorded Future®

By Insikt Group®

CTA-2021-0308

Introduction to Sigma Rules and Detection of Credential Harvesting

Recorded Future®

Recorded Future's Insikt Group created detections to run with SIEM software and incident response guides for 4 popular credential harvesting tools. Sources included the Recorded Future® Platform, Malpedia, PolySwarm, reverse engineering and open-source intelligence (OSINT) enrichments. The target audience for this research includes security practitioners, network defenders, and threat intelligence professionals who are interested in protecting organizations from credential harvesting tools.

Executive Summary

The use of credential harvesting tools is a common and powerful way for threat actors to gain additional access to your infrastructure. Details of a recent Ryuk incident show a 15-step procedure for victim compromise, 2 of which include the use of the credential harvesting tools Mimikatz and LaZagne. These tools were used to move laterally throughout the victim's environment and compromise other hosts on the network.

This article details our research regarding Sigma based detection rules for Mimikatz, LaZagne, T-Rat 2.0, and Osno Stealer. Additionally, we provide an initial incident priority level and a high-level response procedure to help security operations teams respond to credential harvesting incidents.

The Sigma rules provided by the open-source Sigma project and the custom rules developed by Recorded Future (available to existing clients only) offer a powerful capability to detect and respond to credential harvesting using existing SIEM solutions. When combined with properly configured host-based logging, using tools such as Sysmon, Sigma rules can elevate the ability of an organization to detect and respond to threats with increased accuracy and efficiency.

Sigma is a standardized rule syntax which can be converted into many different SIEM-supported syntax formats .The Recorded Future Platform allows clients to access and download Sigma rules developed by Insikt Group for use in their organizations.

Key Judgments

- Most credential harvesting tools are high risk since they enable additional tactics, techniques, and procedures (TTPs) such as lateral movement and privilege escalation; commonly, credential harvesting tools are used as a second-stage tool and indicate the host is already compromised.
- Successful detection and response to credential harvesting activity may prevent intrusions from successfully completing their objectives.
- Sigma rules are an effective way to share detections among multiple platforms. Using Recorded Future priority levels and response procedures with Sigma rules provides an easy-to-implement detection and response capability for cybersecurity teams.

Table of Contents

Background

The strategy of "defense in depth" has provided organizations a powerful blueprint for security architectures for almost 20 years. Rather than rely on a single product or solution as a security panacea, this strategy recommends layering multiple products and solutions, each with their own view of the state of an organization's network. There are numerous products and solutions an organization can use to protect the layers of their infrastructure, such as next-generation firewalls, endpoint detection and response, email protection, and intrusion detection systems (IDS). In combination with tools and technology, threat intelligence adds crucial context to security alerts. Every organization has its own combination of products, solutions, threat intelligence, processes, and models. Regardless of differences in deployments, most organizations rely on a security information and event management (SIEM) tool as a central hub for all of the components of their security architecture. The SIEM acts as the centralized aggregation and correlation device for all the different technologies used in an organization, for this reason, security analysts rely heavily on the SIEM for investigating alerts. Figure 1 shows a common data flow that Recorded Future Professional Services builds out for our customers and shows how the SIEM is the central component of all the cybersecurity technologies.

Like any technology, SIEMs have to be deployed and configured correctly to be effective. There are 3 key stages to maximize the effectiveness of a SIEM: configuring appropriate logging, aggregating and ingesting log data, and performing analytics like correlation and detection. The logs collected into your SIEM give you visibility into your infrastructure; these will be logs such as VPN logs, Windows event logs, IDS logs, and firewall logs. While it would be optimal to collect every log generated in their environment, monetary and resource costs as well as technical restraints can prohibit that approach. Instead, a strategic approach can be used to identify logs that provide an organization with the visibility needed but without the noise of irrelevant events. This will be different for each organization as it will be based on the type of threats a security team is tasked to detect and investigate.

Configuring and aggregating logs to gain visibility into your infrastructure is just the first step. Just as important is the use of analytics to enable correlation and detection. This is the next stage that correlates all the events collected from various sources and applies logic to signal or alert of suspicious activity. It would not be feasible for security analysts to review every event generated in a SIEM, making correlation and detection just as important as visibility, and ensuring analysts have eyes on only the important events. However, consideration must be taken when creating detection and correlation logic. Configuring a SIEM with every open-source detection available will have an adverse effect, as it could subject analysts to "alert fatigue" and may detract from identifying the important alerts they should be investigating. Rules that are too specific will do the opposite, and may cause analysts to miss something important.

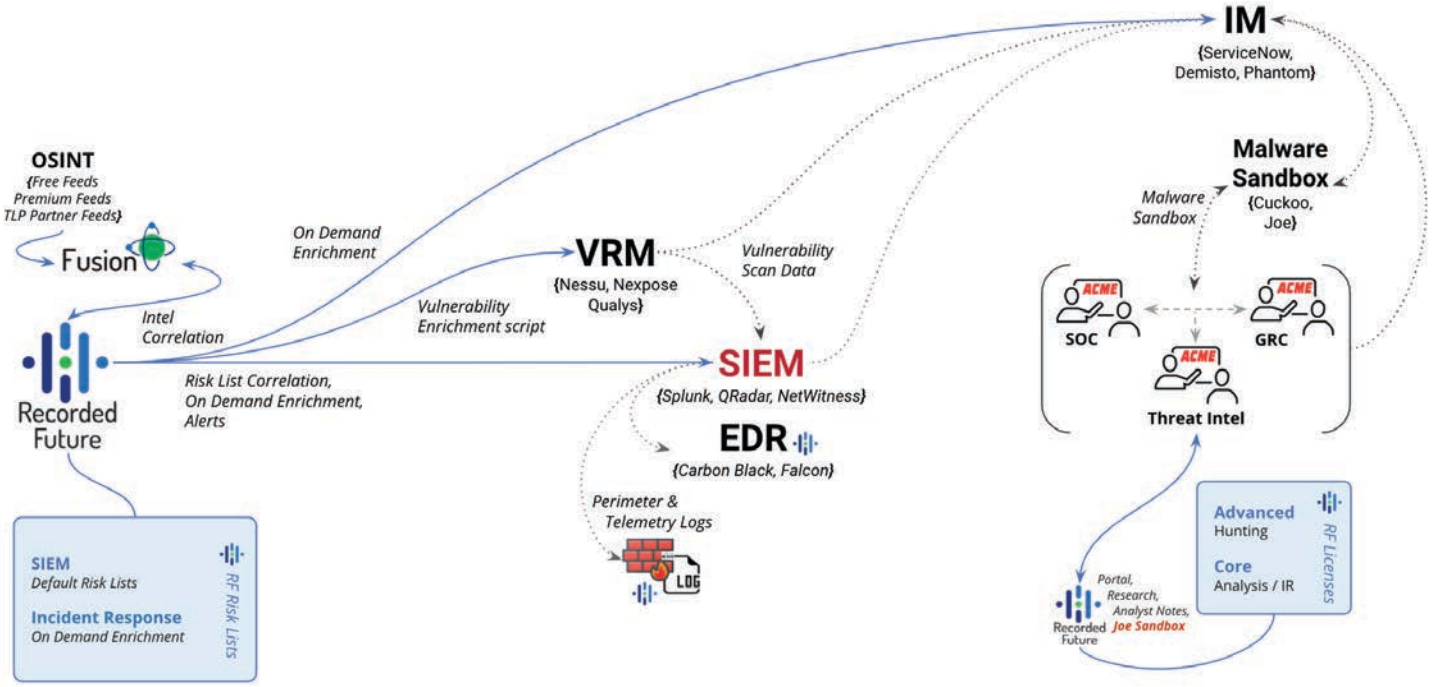

Figure 1: Recorded Future Professional Services recommended security data flow

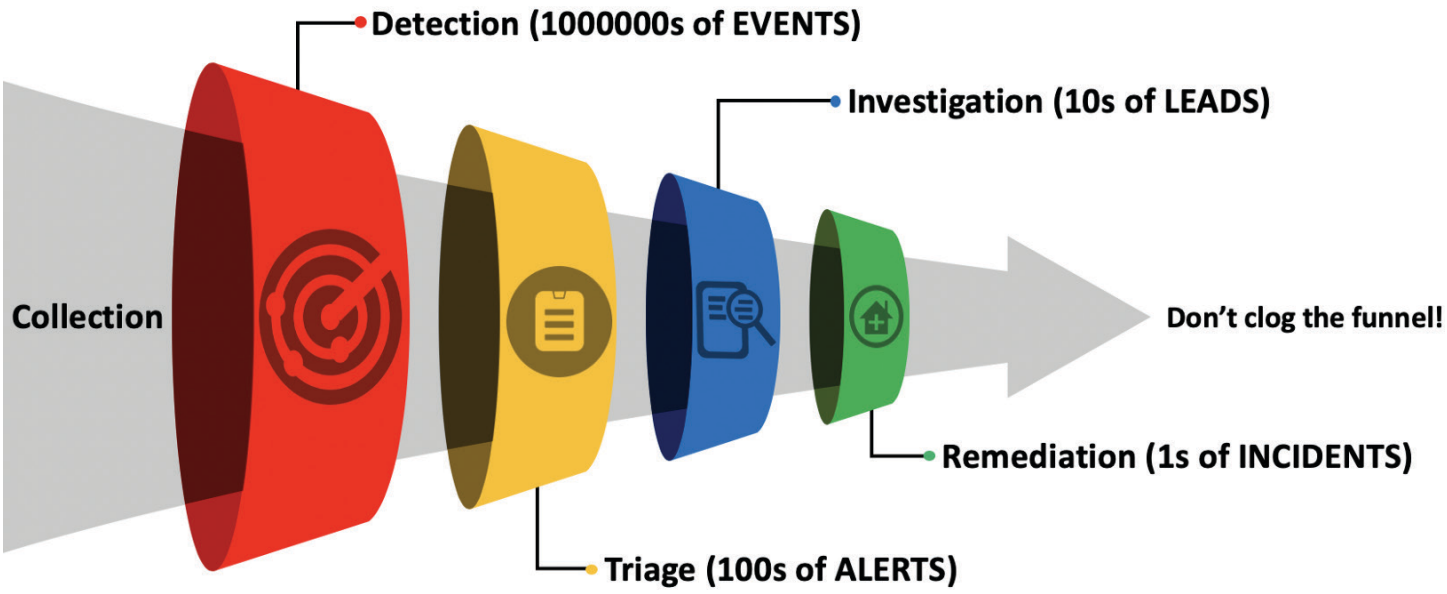

Figure 2: Funnel of Fidelity (Source: SpecterOps)

This approach aligns with the "Funnel of Fidelity" concept presented by SpecterOps. This concept includes 5 stages: Collection, Detection, Triage, Investigation, and Remediation. The idea of the Funnel of Fidelity is that each stage filters out noise in such a way that there is no impact to detection and response to known and unknown threats.

To align with the Funnel of Fidelity concept, organizations can define use cases for detecting and responding to specific threats. Recorded Future has designed a "Use Case Framework" to aid in the development of use cases. The Use Case Framework is built of 11 key components which are required to design, develop, and maintain an effective threat detection.

As shown in Figure 3, the Use Case Framework provides a solution for combining architecture, engineering, detection, and response components to create an end-to-end response to cyber threats.

In addition to creating the actual detection logic for threat detection, the Sigma rules provided can assist in defining the Data, Priority, Logic, and Rule stages of the Use Case Framework, simplifying the creation of use cases.

Sigma Rules Overview

Sigma is a generic format for signatures (detection mechanisms), which are deployed as queries and saved as alerts in SIEM solutions. Sigma rules contain logic to detect computer processes, commands, and operations (this contrasts with correlations against IOCs contained in network traffic events).

As detailed here, Sigma aims to provide platform-agnostic detections that can be shared more widely and then later converted to individual platforms for actionable detections.

As a generic format, Sigma is not deployed as written. First, a given Sigma rule is translated into a SIEM-specific query language using the open source translator sigmac or the online utility uncoder.io (free and claims to collect no user data). As of this writing, Sigma can be translated into 40 different query formats, including the most common SIEM query languages such as Splunk, Azure Log Analytics (for use with Sentinel), and Elasticsearch.

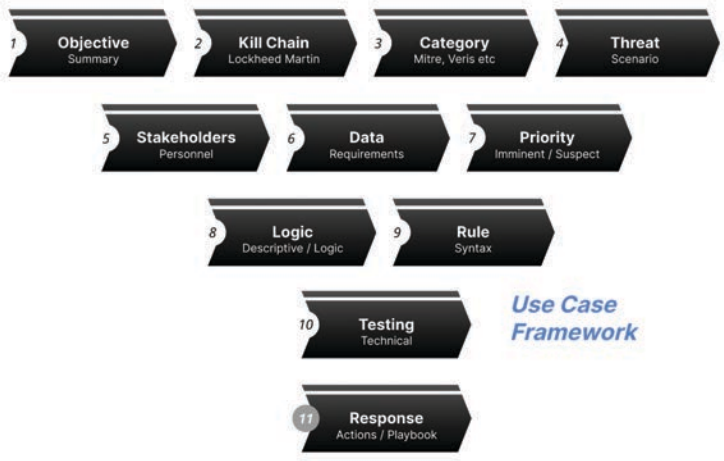

Figure 3: Recorded Future Use Case Framework[1]

[1] For additional details regarding the Use Case Framework and methodology, contact Professional Services at Recorded Future.

A client's ability to deploy Sigma rules will depend on the type of logs they ingest into their SIEM. For example, Sigma rules that look at specific Windows Event IDs for the detection need to have the associated Window Event logs collected into the SIEM to be successful. Additionally, certain Sigma rules are only designed to detect events from Windows and not Unix operating systems. Or some rules may be deployed but are most often only valuable after tuning out of activity considered "normal" in a given environment (see "Using Endpoint Logs for Security" here and full post here). Before deployment of a Sigma rule, it is important to review that you have the visibility needed for detection and a defined way to respond and remediate the threat.

The full specification for Sigma rules can be found here. The specification defines all of the different components of a Sigma rule. Figure 4 shows the general schema of a Sigma rule. The 2 main components of any Sigma rule are the "logsource" and "detection" sections, as these sections detail the sources and logic required for successful deployment and detection.

```
title
id [optional]
related [optional]
   - type {type-identifier}
     id {rule-id}
status [optional]
description [optional]
author [optional]
references [optional]
logsource
   category [optional]
   product [optional]
   service [optional]
   definition [optional]
   ...
detection
   {search-identifier} [optional]
      {string-list} [optional]
      {field: value} [optional]
   ...
   timeframe [optional]
   condition
fields [optional]
falsepositives [optional]
level [optional]
tags [optional]
...
[arbitrary custom fields]
```

Figure 4: Sigma rule YAML schema (Source: Sigma)

Log Sources

The "logsource" component defines the log sources required for a Sigma rule to work. This can detail the required log sources, platform and application. This is classified with 3 attributes: "Category", "Product", and "Service".

Category is used to group the log files to certain technologies such as firewalls, IDS, and web server logs. Product is more specific than category and would be used to define items such as Windows Event Logs. Service can be used to define only a subset of certain Product logs — for instance, instead of all Windows Security Event logs, maybe only Security Event logs are needed.

There is an additional attribute, "definition", that is not automatically used when converting a Sigma rule to a output format, but can provide additional and helpful information about the log sources needed.

All of the information provided by the logsource section informs the translation layer as to what fields and attributes are mapped to SIEM-specific indices and fields. The mapping from the generic Sigma attributes to SIEM-specific indices and fields is also fully configurable. If none of the default mappings found here match the configuration in the SIEM perfectly, it is possible to create a new configuration to more accurately map.

Having effective logging enabled is crucial to successfully using Sigma rules. In the public Sigma rules repository, the most common product is "windows" and the most common category is "process creation". This is no accident, as process monitoring enables detection of 234 unique MITRE ATT&CK techniques — more than any other data source identified by MITRE ATT&CK.

One of the simplest and cheapest ways to enable process monitoring is Sysmon. Sysmon provides fast and efficient collection of a large number of process-related activities, including process creation, remote process access, file access, and registry modification. With the ability to log so much, it is important to configure Sysmon itself to maximize the signal to noise ratio in logs of Windows host behavior. There are 2 Sysmon configuration files that are effective with minimal tuning required by an organization: Sysmon Modular and Swift on Security's Sysmon Config.

Detection Logic

The detection component provides search-identifiers that represent searches on defined log data. As per the specification, the search-identifiers are case-insensitive, can include wildcard characters, regular expressions. Lists of search-identifiers can be built and linked with a logical OR, as per the example in Figure 5. The highlighted search-identifier list is read logically as "Command Line Contains net1 stop OR net stop OR cmd /c choice /t 10 /d y OR vssadmin.exe OR wmic.exe shadowcopy delete OR taskkill /f /im".

```
detection:
    selection_1:
        ParentImage|endswith: '\explorer.exe'
    selection_2:
        - CommandLine|contains:
            - net1 stop
            - net stop
            - cmd /c choice /t 10 /d y
            - vssadmin.exe
            - wmic.exe shadowcopy delete
            - taskkill /f /im

    condition: selection_1 and selection_2
```

Figure 5: Sigma sample rule using search-identifier list

The logic included in the selections as well as in the condition statement is generally used to detect single log entries. In the example in Figure 5, only process creation log entries that satisfy both selection_1 and selection_2 will trigger alerts. 1 alert will be generated per log entry that matches the condition. Sigma's most supported use case is searching for single log entries, but the condition statement can be used for some limited aggregations and correlations such as count and near. The rule Failed Logins with Different Accounts from Single Source System uses count to find a certain number of failed login attempts from the same system. The rule APT29 Google Update Service Install uses near to alert when log entries matching 2 different selection conditions occur in close temporal proximity to each other. However, these aggregations do not have support in some SIEM-specific languages. Using the aggregation and correlation logic makes Sigma rules less portable.

The detection component of Sigma provides a lot of flexibility. With that great flexibility comes great responsibility. Detection logic can vary in how resilient it is to threat actor adaptation. There is a trade-off in choosing how resilient or brittle to make a detection. Creating resilient detections that make it difficult for threat actors to adapt causes the detections to be broad and possibly prone to false positives (alerting on benign activity). Creating detections that are more precise can provide more context to an alert, but are prone to false negatives (not alerting on malicious activity) as adversaries adapt. Finding a balance between the two can be done by considering the cost the detections impose on an adversary.

A good model to understand the cost imposed on an adversary who needs to avoid specific detection logic is David J. Bianco's Pyramid of Pain seen in Figure 6. The Pyramid of Pain ranks the types of detections that can be used in terms of how much it would cost (in time, effort, or money) an adversary to circumvent. At the bottom of the Pyramid of Pain are hash values of malicious files. These are the easiest for adversaries to circumvent; in the simplest case, padding the end of a file would result in changing a hash value. For each higher level on the Pyramid of Pain, the types of detection logic become increasingly costly for an adversary to overcome. At the highest level, Tactics, Techniques, and Procedures (TTPs) correspond to detection logic that captures fundamental artifacts of high-level techniques. For instance, this could be detection of Word documents with malicious macros. Appropriate detection for such TTPs can impose a high cost on adversaries who use those TTPs.

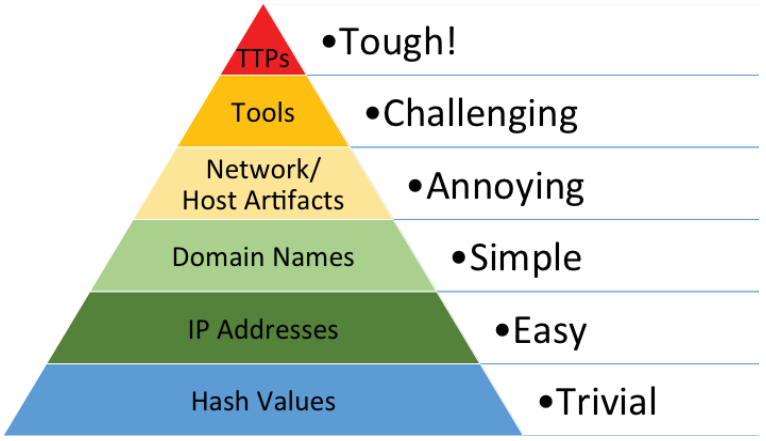

Figure 6: Pyramid of pain model for understanding cost imposed on adversaries by detection types (Source: David J. Bianco)

Since Sigma rules are flexible and have access to a variety of logs, detections could be written in a multitude of ways for any given threat. Choosing appropriate detections is critical to good Sigma rules. For instance, in the public Sigma rules repository, there are 17 different rules related to detecting the credential stealer Mimikatz. Full discussion of the detection of Mimikatz can be found later in this report, but examining 2 of the rules helps illustrate the pros and cons of targeting different levels of the pyramid.

On the lowest level of the pyramid is the rule Malicious PowerShell Commandlet Names. This rule detects use of specific file names such as "Invoke-Mimikatz.ps1" and "Invoke-Mimikittenz.ps1." For an adversary trying to circumvent this detection, it would be trivial to change the name of the "Invoke-Mimikatz.ps1" file.

At the highest level of the pyramid is the rule LSASS Access from Non System Account. This rule detects non-privileged processes that attempt to access the LSASS process. This is an artifact of a critical step in executing Mimikatz to collect credentials from a system. It would be non-trivial for attackers to modify Mimikatz to avoid triggering this detection.

While at first glance, it might seem that the second rule is strictly better than the first, it does lack some context and precision found in the first rule. The behavior that will be

detected by the rule LSASS Access from Non System Account is not limited to Mimikatz. In fact, some security products will trigger this rule.[1] Triaging an alert based on LSASS Access from Non System Account will require additional effort to understand what happened. On the other hand, since the rule Malicious PowerShell Commandlet Names describes specific post-exploitation PowerShell scripts, when that rule generates an alert there is already much more context known about what caused that alert to trigger.

The tradeoff between these levels can be thought of as being on a spectrum between broad threat hunting detection logic and alert detection logic .Threat Hunting detections make it difficult for adversaries to adapt and as a result have a low false negative rate ,but are prone to high false positives .Alert detections have a low false positive rate and provide more context to an alert ,but may in time have a high false negative rate as adversaries can easily adapt to the detection logic.

A mix of both types of rules is ideal .Taking a strategic approach to developing and implementing your use cases using the Use Case Framework will help to balance your alert detection versus threat hunting detections .The Objective and Threat factors in the Use Case Framework will help aid on how broad or precise to have your detections .For example ,Insikt Group's objective for the Credential Harvesting Sigma rules is to detect specific tools used for credential harvesting .This objective yields more of a precise detection as opposed to an objective such as" Identify Credential Harvesting Activity".

Sigma Rule Testing

The deployment of new Sigma rules requires a high level of trust and understanding of how a rule will impact the alert workload of an organization and a minimum of false positives or false negatives .To develop that trust ,Recorded Future follows a thorough testing and analysis process for all Sigma rules that we publish .As discussed in the previous section ,all rules will have some balance of false positives and false negatives ,but through testing we can ensure that false positives that exist are predictable and that the true positive rate is high .Even in the cases of broader detection logic ,where false positives are inevitable ,our testing process ensures that there are no irrelevant logs highlighted by our rules.

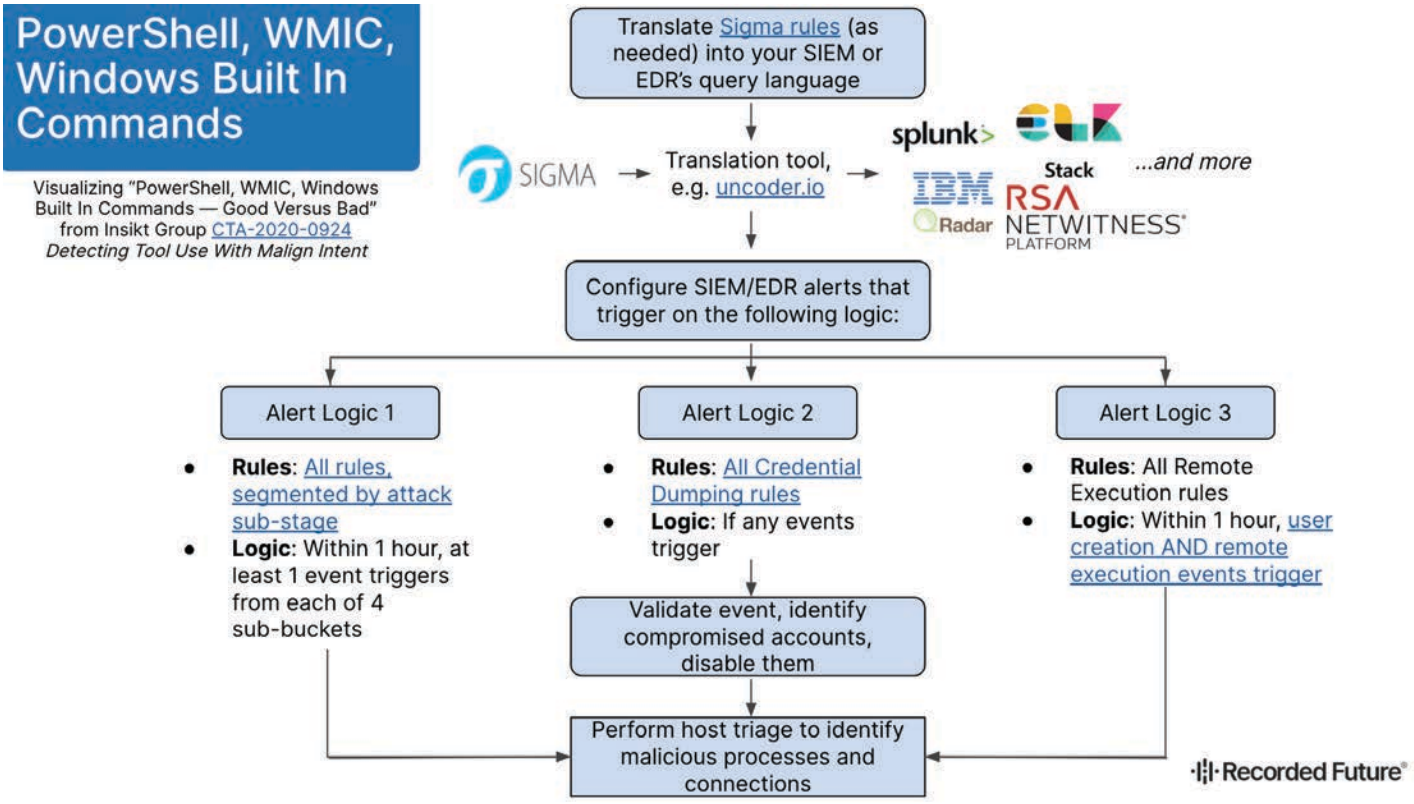

Figure 7: Sigma rule detection flow

[1] According to the Threat Hunter Playbook, Microsoft's Monitoring Agent "pmfexe.exe" commonly accesses the LSASS process.

Testing Process

Rules written by Recorded Future are tested first against logs collected during analysis of specific threats .We validate that the Sigma rules can detect the intended aspect of the threat, and against the Splunk Boss of the SOC) BOTS (data set and the Mordor data set .Both data sets contain a large volume of log entries from both normal operations and adversary emulation events .The rule is tested by converting the Sigma rule to a query and running the query on a SIEM) either Splunk or Elasticsearch(containing the large log data sets.

Recorded Future Priority	Impact	Service Level Objective Target
Priority 0	Critical	4 Hours
Priority 1	High	8 Hours
Priority 2	Medium	2 Business Days
Priority 3	Low	5 Business Days

Table 1: Incident priority table

Threat Analysis

Alerts generated by Sigma rules act as a starting point in the incident response cycle, but without a proceduralized way to triage, respond to, and remediate the alerts, the Sigma detection itself may end up in an incident queue for months. For every Sigma rule provided by Insikt Group, we have also assigned an initial priority rating that can be adjusted based on other contextual components such as risk of confidential information being exposed or incidents involving multiple hosts. The priority is determined by Insikt Group using the Recorded Future Incident Priority Matrix, as described in the section of this report titled "Recorded Future Incident Priority Matrix".

Figure 7 shows an example of how Sigma detections can be used in conjunction with an incident response procedure. First, alert logic is applied using one or multiple Sigma rules. Once the alert is fired, the security team will respond in accordance to the defined procedure associated with the higher-level use case the Sigma rule is tied to. In the "Response and Mitigations" section of this report, we have provided a high-level response and remediation procedure for Credential Harvesting alerts.

Recorded Future Incident Priority Matrix

Triage of an investigation includes identifying the number of affected assets and related incidents, potentially identifying lateral movement and pivoting of threats from a threat actor during an attack.

For each Sigma rule, Insikt Group has provided an initial priority rating of Priority 0, Priority 1, Priority 2 or Priority 3, as shown in Table 1. These ratings are the result of our evaluation of the technical aspects of the malware and the relevancy. The priority levels and associated impact are a derivative of the "Cyber Incident Severity Schema".

However, other modifiers should be considered, such as the scale of the incident and business impact:

- Incident Scale:
  - Is the number of systems affected greater than 5?
  - What is the spreadability of malware, or is there evidence of lateral movement?
- Business Impact:
  - Do the individuals have access to confidential or privileged information?
  - Are critical asset(s) involved?

Recorded Future has created the Incident Priority Matrix, below, to aid in the prioritization of incidents, this is a similar but simpler approach to that of the "CISA National Cyber Incident Scoring System". The "Recorded Future Priority" noted in the "Level" tag of the Sigma rule, is the priority that we have identified for our custom Sigma rules. This priority can be upgraded depending on the assessment of the business context modifiers "Critical Asset", Access to "Privileged Information" and "Widespread". A priority is upgraded 1 level (From a Priority 3 to a Priority 2), when any 1 or 2 of the modifiers are met. If all the modifiers are met, then the priority jumps 2 levels (From a Priority 3 to a Priority 1).

Recorded Future Priority	Incident Scale or Business Modifiers	Modified Priority
Priority 0	No priority adjustment needed	
Priority 1	Critical Asset and/or Access to Privileged Information and/or Widespread	Priority 0
Priority 2	Critical Asset and Access to Privileged Information and Widespread	Priority 0
	Critical Asset or Access to Privileged Information or Widespread	Priority 1
Priority 3	Critical Asset and Access to Privileged Information And Widespread	Priority 1
	Critical Asset or Access to Privileged Information or Widespread	Priority 2

Table 2: Recorded Future Incident Priority Matrix

The Incident Priority Matrix (Table 2) and Priority Table (Table 1) are derived from the Cybersecurity and Infrastructure Security Agency (CISA) and are used as a guideline. They may need to be customized for your specific use based on risk factors or Service Level Objectives that are already defined in your organization.

Credential Stealers

Mimikatz

Benjamin Delpy ,whose username on GitHub is gentilkiwi, created Mimikatz as an open source tool designed to target devices running Windows OS and can run pass-the-hash ,pass-the-ticket ,kerberoasting ,and more .Since its creation ,Mimikatz has been associated with many intrusions and is frequently updated and included in many penetration testing frameworks.

Sigma Rules

In the Sigma Rule Github Repository, we have identified 17 open-source Sigma rules created to detect Mimikatz in some capacity.

1. Mimikatz Use
2. Mimikatz In-Memory
3. Mimikatz Command Line
4. Mimikatz Detection LSASS Access
5. Mimikatz DC Sync
6. Mimikatz through Windows Remote Management
7. Antivirus Password Dumper Detection
8. DLL Load via LSASS
9. Successful Overpass the Hash Attempt
10. Possible Process Hollowing Image Loading
11. Quick Execution of a Series of Suspicious Commands
12. LSASS Access from Non System Account
13. Malicious PowerShell Keywords
14. CreateMiniDump Hacktool
15. Credential Dumping Tools Service Execution
16. Malicious Nishang PowerShell Commandlets
17. Malicious PowerShell Commandlet Names

We have identified 4 of these rules that will detect the most common uses of Mimikatz. We have provided the Sigma rules in Appendices A through D.

1. Mimikatz Use
2. Mimikatz Command Line
3. LSASS Access from Non System Account
4. Mimikatz In-Memory

LaZagne

The LaZagne project is an open source tool used to extract many different types of passwords on a host machine. LaZagne is developed in Python 2.7 and according to the developer can be executed entirely in memory with no disk artifacts. LaZagne collects passwords from sources such as the system (similar to Mimikatz), Web Browsers, Databases, Games, GIT (for Windows), Keepass, Wifi, and admin tools like CyberDuck, OpenVPN, WinSCP and FileZilla.

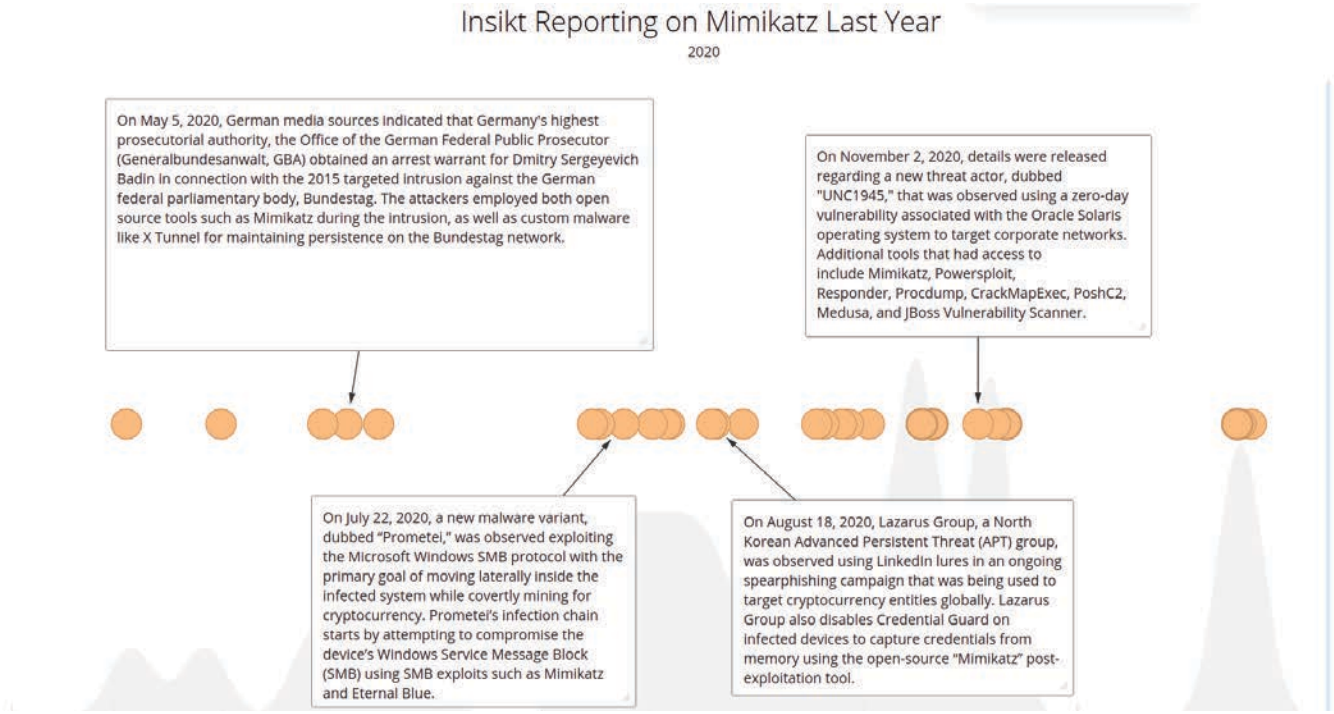

Figure 8: Recent use of Mimikatz (Source: Recorded Future)

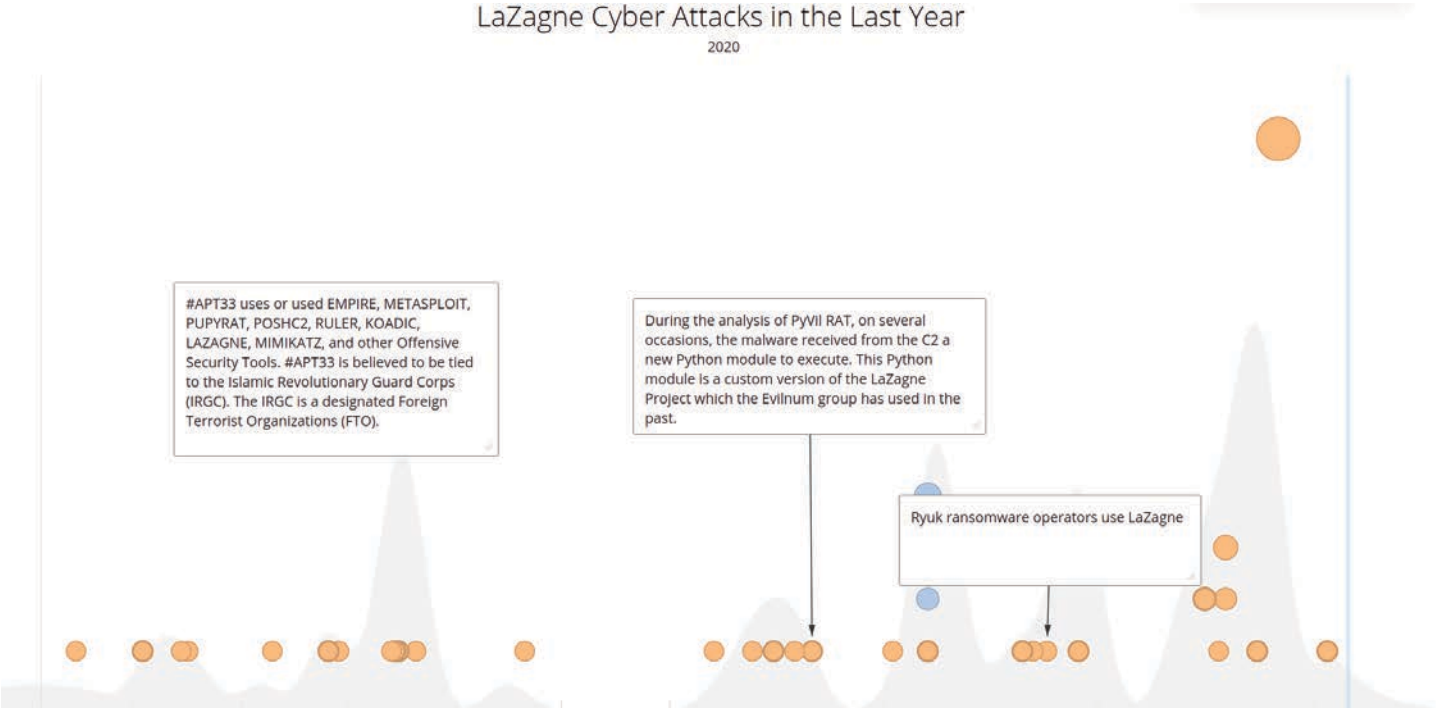

Figure 9: Recent use of LaZagne (Source: Recorded Future)

Sigma Rules

Available for our clients, we have created 2 Sigma Rules:

1. MAL_LaZagne_LSASS_Method
2. MAL_LaZagne_SQLite

The Sigma rule for "MAL_LaZagne_LSASS_Method" is similar to that of Mimikatz Sigma rules in that it is looking for abuse of LSASS.exe; however, because LaZagne is Python-based, the call trace within the event log will also contain the the string "Python27.dll". Additionally, we have also noticed that under default conditions, LaZagne modules are run from the "\AppData\Local\Temp\" directory.

The second rule, "MAL_LaZagne_SQLite", detects the loading of a specific SQLite3 Dynamic Link Library (DLL) that is included in the LaZagne package. The DLL itself is not malicious, but viewing the relations in VirusTotal, we can see this specific Sqlite3 DLL is commonly tied to the execution of LaZagne. The loading of this particular DLL in the "AppData\Local\Temp" directory correlates to a high possibility of LaZagne activity.

Credential Stealers Technical Assessment

Recorded Future's Technical Assessment, as described in the Recorded Future Incident Priority Matrix section identifies Mimikatz and LaZagne detections as a Priority 1 threat. This is because detections of both Mimikatz and LaZagne almost always lead to compromise of credentials, that can directly lead to lateral movement and credential leaks.

Both Mimikatz and LaZagne have been used in enterprise scale intrusions:

- Mimikatz has been reportedly used by Lazarus Group, UNC1945 and included in malware botnets such as Prometei.
- LaZagne has been reportedly used by APT33 and Ryuk operators.

While the initial priority is at a Priority 1 it should be upgraded to a Priority 0 if a security analyst determines the additional modifiers such as Critical Asset, Access to Privileged Information or Widespread exist.

Recorded Future Priority	Widespread or Business Modifierers	Modified Priority
Priority 0	Critical Asset and/or Access to Privileged Information and/or Widespread	Priority 0
Priority 1	Critical Asset and/or Access to Privileged Information and/or Widespread	Priority 0
Priority 2	Critical Asset and Access to Privileged Information and Widespread	Priority 0
	Critical Asset or Access to Privileged Information or Widespread	Priority 1
Priority 3	Critical Asset and Access to Privileged Information and Widespread	Priority 1
	Critical Asset or Access to Privileged Information or Widespread	Priority 2

Table 3 Credential Stealers Incident Priority Matrix

Infostealers

While Mimikatz and LaZagne focus on the collection of credentials on Windows systems ,specifically credentials found in the LSASS process memory ,Infostealer malware families collect credentials stored on a victim's computer from a variety of sources 2 .new Infostealer families that Recorded Future has identified in the last 6 months are T-Rat 2.0 and Babax/Osno Stealer.

T-Rat 2.0

T-RAT 2.0 (Intelligence Card) is a remote access tool for sale by threat actor D108 (Intelligence Card) on the dark web that uses legitimate Telegram APIs for command and control. Because the malware uses Telegram as a command and control (C2) service, buyers don't have to set up any custom infrastructure, they simply provide the threat actor selling T-RAT 2.0, D108, with a Telegram bot ID and bot token. T-RAT 2.0 provides a wide set of features including keylogging, password stealing, screen capture, hidden VNC, and RDP access.

Sigma Rules

At the time of this writing, T-RAT 2.0 attempts to obfuscate its execution by running the primary payload with the name sihost.exe. This is static across multiple deployments, even where the initial downloader had changed names. While there is a publicly available rule to detect suspicious use of sihost.exe, "System File Execution Location Anomaly" (Appendix E), Recorded Future notes that the rule, while effective, will capture a variety of threats that attempt to masquerade as a system utility and not specifically T-RAT 2.0.

For this reason, Recorded Future developed a second rule, MAL_TRAT_Initial_Check_In, that combines the process name of sihost.exe with the fact that T-Rat 2.0 will attempt to resolve Telegram's API domain name, api.telegram[.]com. This rule targets the specific threat posed by T-Rat 2.0. It detects the high level TTPs of using Telegram for C2 and masquerading as a system process.

Osno Stealer

Osno Stealer is based on the open source BABAX Stealer, which was released in June 2019. Osno stealer has the ability to capture sensitive data such as passwords and session information and exfiltrate that data via Telegram and Discord. Osno builds on the functionality of BABAX Stealer, adding a new module that claims to function as ransomware, but Recorded Future analysts found the function overwrites targeted files with random data, which would better be described as a wiper module.

Sigma Rules

During the execution of the Osno stealer, it gathers information about the computer it is executing on. To do this it issues a series of commands that do not change between payloads. These commands have a unique format to them that in our testing had no false positives. The full rule MAL_Babax_Stealer_Execution is available only to Recorded Future clients.

Information Stealers Technical Assessment

Recorded Future's Technical Assessment, as described in the Recorded Future Incident Priority Matrix section, identifies T-RAT 2.0 and Osno Stealer detections as a Priority 2 threat. While both information stealers pose a threat to credential loss, as of now they are not widely used by threat actors and execution of the information stealer does not always lead to account compromise. However, Information Stealer detections should be quickly escalated to a higher priority after assessment of the additional modifiers for Critical Asset, Access to Privileged Information or Lateral Movement.

Recorded Future Priority	Widespread or Business Modifierers	Modified Priority
Priority 0	Critical Asset and/or Access to Privileged Information and/or Widespread	Priority 0
Priority 1	Critical Asset and/or Access to Privileged Information and/or Widespread	Priority 0
Priority 2	Critical Asset and Access to Privileged Information and Widespread	Priority 0
	Critical Asset or Access to Privileged Information or Widespread	Priority 1
Priority 3	Critical Asset and Access to Privileged Information and Widespread	Priority 1
	Critical Asset or Access to Privileged Information or Widespread	Priority2

Table 4: Information stealers incident priority matrix

Response and Mitigations

The following workflow shows a best practice approach for maintaining content development activities inline with investigations, detection and alerting. The workflow is not exhaustive or detailed and should be used as a reference or overarching high level process flow alongside a more detailed process flow.

In general, when responding to alerts regarding credential harvesting your main objective is to determine if it is widespread and related to a bigger intrusion. For eradication, there are 2 methods that can be taken:

1. Reset the passwords for every compromised account.
2. Decommission compromised accounts.

The advantage of decommissioning accounts over resetting passwords is that reuse of the stolen credentials and any post-breach activity are able to be tracked by defenders.

Another action would be to record and store the hashes of the breached passwords .This will help determine if passwords found in data dumps have already been remediated or if they are new ones.

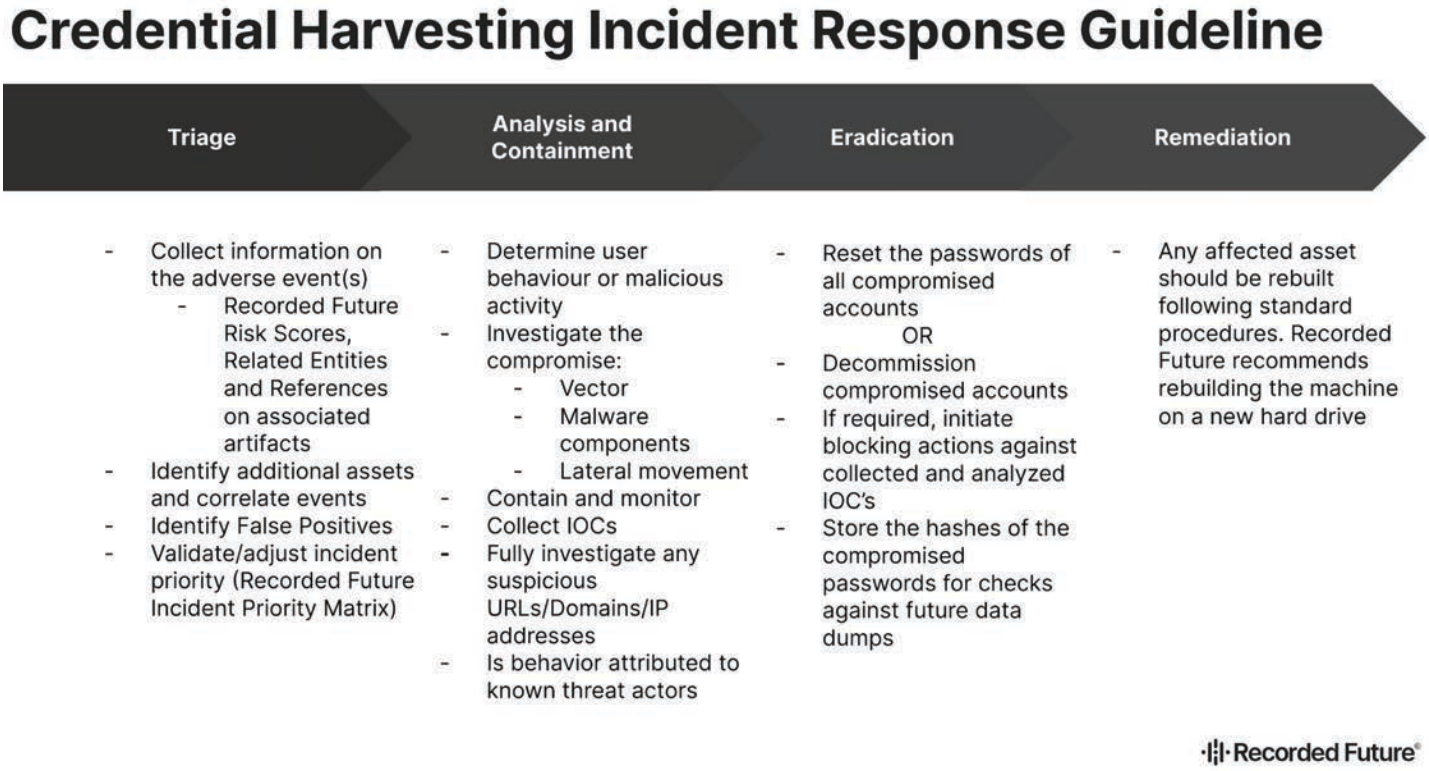

Figure 10: Credential harvesting incident response guideline

Outlook

Harvesting credentials is a crucial step for many threat actors and is often done early in the attack chain. Mimikatz in particular has been used by both nation-state threat actors conducting espionage and big-game ransomware threat actors. Detecting and responding to the use of the credential access techniques reported here is a force multiplier on preventing further incidents with more severe business impact, such as ransomware.

The Sigma rules provided by both the public repository and the custom rules developed by Recorded Future offer a powerful way to detect and respond to credential harvesting using already deployed SIEMs. When combined with properly configured host-based logging, using tools like Sysmon, Sigma rules can elevate the ability of an organization to detect and respond to threats.

Appendix A: Mimikatz Use (Sigma Rule)

```
    title :Mimikatz Use
    id06 :d71506-7beb4-f22-8888-e2e5e2ca7fd8
    description :This method detects mimikatz keywords in different Eventlogs) some of them only appear in older Mimikatz version
that are however still used by different threat groups(
    author :Florian Roth
    date2017/01/10 :
    modified2019/10/11 :
    tags:
     -    attack.s0002
     -    attack.t1003 #          an old one
     -    attack.lateral_movement
     -    attack.credential_access
     -    car2013-07-001.
     -    car2019-04-004.
     -    attack.t1003.002
     -    attack.t1003.004
     -    attack.t1003.001
     -    attack.t1003.006
    logsource:
       product :windows
    detection:
       keywords:
          Message:
     *" -            mimikatz"*
     *" -            mimilib"*
     3> *" -            eo.oe"*
     *" -            eo.oe.kiwi"*
     *" -            privilege::debug"*
     *" -            sekurlsa::logonpasswords"*
     *" -            lsadump::sam"*
     *" -            mimidrv.sys"*
     *" -            p::d"*
     *" -            s::l"*
       condition :keywords
    falsepositives:
     -    Naughty administrators
     -    Penetration test
    level :critical
```

Table 5: Sigma Rule — Mimikatz Use (Source: Sigma)

Appendix B: Mimikatz Command Line (Sigma Rule)

```
title :Mimikatz Command Line
id :a642964e-bead4-bed8910-1-bb4d63e3b4d
description :Detection well-known mimikatz command line arguments
author :Teymur Kheirkhabarov ,oscd.community
date2019/10/22 :
modified2020/09/01 :
references:
 -    https//:www.slideshare.net/heirhabarov/hunting-for-credentials-dumping-in-windows-environment
tags:
 -    attack.credential_access
 -    attack.t1003 #          an old one
 -    attack.t1003.001
 -    attack.t1003.002
 -    attack.t1003.004
 -    attack.t1003.005
 -    attack.t1003.006
logsource:
    category :process_creation
    product :windows
detection:
    selection:1_
        CommandLine|contains:
 -            DumpCreds
 -            invoke-mimikatz
    selection:2_
        CommandLine|contains:
 -            rpc
 -            token
 -            crypto
 -            dpapi
 -            sekurlsa
 -            kerberos
 -            lsadump
 -            privilege
 -            process
    selection:3_
        CommandLine|contains:
'::' -
    condition :selection 1_or selection 2_and selection3_
falsepositives:
 -    Legitimate Administrator using tool for password recovery
level :medium
status :experimental
```

Table 6: Sigma Rule — Mimikatz Command Line (Source: Sigma)

Appendix C: LSASS Access from Non System Account (Sigma Rule)

```
    title :LSASS Access from Non System Account
    id962 :fe167-e48d4-fd6-9974-11e5b9a5d6d1
    description :Detects potential mimikatz-like tools accessing LSASS from non system account
    status :experimental
    date2019/06/20 :
    modified2019/11/10 :
    author :Roberto Rodriguez@ Cyb3rWard0g
    references:
     -   https//:github.com/Cyb3rWard0g/ThreatHunter-Playbook/tree/master/playbooks/windows_06/credential_access/T1003_credential_
dumping/lsass_access_non_system_account.md
    tags:
     -    attack.credential_access
     -    attack.t1003 #          an old one
     -    attack.t1003.001
    logsource:
        product :windows
        service :security
    detection:
        selection:
            EventID:
    4663 -
    4656 -
            ObjectType' :Process`
            ObjectName|endswith\' :lsass.exe`
        filter:
            SubjectUserName|endswith`$' :
        condition :selection and not filter
    fields:
     -    ComputerName
     -    ObjectName
     -    SubjectUserName
     -    ProcessName
    falsepositives:
     -    Unknown
    level :critical
```

Table 7: Sigma Rule — LSASS Access from Non System Account (Source: Sigma)

Appendix D: Mimikatz In-Memory (Sigma Rule)

```
    title :Mimikatz In-Memory
    id :c0478ead5336-46-c2-bd5e-b4c84bc3a36e
    status :experimental
    description :Detects certain DLL loads when Mimikatz gets executed
    references:
     -    https//:securityriskadvisors.com/blog/post/detecting-in-memory-mimikatz/
    tags:
     -    attack.s0002
     -    attack.t1003
     -    attack.lateral_movement
     -    attack.credential_access
     -    car2019-04-004.
    logsource:
        category :image_load
        product :windows
    date2017/03/13 :
    detection:
        selector:
            Image' :C\:Windows\System32\rundll32.exe'
        dllload1:
            ImageLoaded\*' :vaultcli.dll'
        dllload2:
            ImageLoaded\*' :wlanapi.dll'
        exclusion:
            ImageLoaded:
    ' -             ntdsapi.dll'
    ' -             netapi32.dll'
    ' -             imm32.dll'
    ' -             samlib.dll'
    ' -             combase.dll'
    ' -             srvcli.dll'
    ' -             shcore.dll'
    ' -             ntasn1.dll'
    ' -             cryptdll.dll'
    ' -             logoncli.dll'
        timeframe30 :s
        condition :selector | near dllload1 and dllload2 and not exclusion
    falsepositives:
     -    unknown
```

Table 8: Sigma Rule — Mimikatz In-Memory (Source: Sigma)

Appendix E: System File Execution Location Anomaly (Sigma Rule)

```
title :System File Execution Location Anomaly
id :e4a6b256-3e47-40fc89-d2-7a477edd6915
status :experimental
description :Detects a Windows program executable started in a suspicious folder
references:
 -    https//:twitter.com/GelosSnake/status934900723426439170/
author :Florian Roth ,Patrick Bareiss
date2017/11/27 :
tags:
 -    attack.defense_evasion
 -    attack.t1036
logsource:
    category :process_creation
    product :windows
detection:
    selection:
        Image:
\*' -            svchost.exe'
\*' -            rundll32.exe'
\*' -            services.exe'
\*' -            powershell.exe'
\*' -            regsvr32.exe'
\*' -            spoolsv.exe'
\*' -            lsass.exe'
\*' -            smss.exe'
\*' -            csrss.exe'
\*' -            conhost.exe'
\*' -            wininit.exe'
\*' -            lsm.exe'
\*' -            winlogon.exe'
\*' -            explorer.exe'
\*' -            taskhost.exe'
\*' -            Taskmgr.exe'
\*' -            sihost.exe'
\*' -            RuntimeBroker.exe'
\*' -            smartscreen.exe'
\*' -            dllhost.exe'
\*' -            audiodg.exe'
\*' -            wlanext.exe'
    filter:
        Image:
' -            C\:Windows\System32'*\\
' -            C\:Windows\system32'*\\
' -            C\:Windows\SysWow64'*\\
' -            C\:Windows\SysWOW64'*\\
' -            C\:Windows\explorer.exe'
' -            C\:Windows\winsxs'*\\
' -            C\:Windows\WinSxS'*\\
\' -            SystemRoot\System32'*\\
    condition :selection and not filter
fields:
 -    ComputerName
 -    User
 -    Image
falsepositives:
 -    Exotic software
level :high
```

Table 9: Sigma Rule — System File Execution Location Anomaly (Source: Sigma)

About Recorded Future

Recorded Future is the world's largest provider of intelligence for enterprise security. By combining persistent and pervasive automated data collection and analytics with human analysis, Recorded Future delivers intelligence that is timely, accurate, and actionable. In a world of ever-increasing chaos and uncertainty, Recorded Future empowers organizations with the visibility they need to identify and detect threats faster; take proactive action to disrupt adversaries; and protect their people, systems, and assets, so business can be conducted with confidence. Recorded Future is trusted by more than 1,000 businesses and government organizations around the world.

Learn more at recordedfuture.com and follow us on Twitter at @RecordedFuture.