Quality Over Quantity: Comparison of Windows Security Audit Recommendations in Detecting Malware

GIAC (GCIH) Gold Certification

Author: Nicole JeNaye, nicole.jenaye@gmail.com
Advisor: *John Hally*

Accepted: *10/31/2021*

Abstract

More than seventy percent of businesses operate with at least one version of Microsoft Windows in their environment (Keizer, 2020). For security teams focused on detecting threats in these environments, the generation and collection of Windows security event logs is typically a significant component of the detection strategy. While the refrain "just log everything and sort it out later" is often heard, this is typically not feasible or prudent given resource limitations. Thus, triaging which events to collect becomes essential, leaving the security architect to determine what to collect and what to ignore. Microsoft provides Windows Security Audit Recommendations, which outline the Default, Baseline, and Stronger audit policy recommendations (Foulds, 2017), without providing specifics on what these templates are best suited for, or how they were developed. This research paper aims to provide security professionals with quantitative analysis to help determine which of the recommended audit policies detect malicious activity and what customizations may be helpful.

1. Introduction

For security analysts, the development of an effective strategy for log collection and analysis is often a challenge fraught with uncertainty. Many white papers discuss logging strategies and provide guidance on what to log, how long to retain those logs, how to utilize those logs, and so on. However, the core conundrum in logging is balancing the resources needed to collect, retain, and analyze logs with the urge to log everything out of fear that one uncollected log source will turn out to be critical. For Windows environments, the ability to collect a wide variety of security event logs can be a boon for security analysts, but only if the audit policies have been 'configured properly' to generate the necessary logs. But what does 'configured properly' mean?

Microsoft provides Audit Policy Recommendations (Foulds, 2017) for both workstations and servers, with three profiles for each: Default, Baseline, and Stronger. These recommendations indicate which security audit policy subcategories to enable and what inclusion settings to configure, whether for "Success", "Failure", or both. Security analysts typically reference Windows security events by IDs, but each audit policy subcategory generates multiple unique events when enabled. The recommendations provided by Microsoft provide a starting point for configuring security event log generation but provide little guidance on which strategy to select or why. For organizations that do not have unlimited resources for log collection, retention, and analysis, a more precise method for developing a security audit policy tailored for their organization is warranted.

This paper aims to provide a quantitative analysis of the Microsoft-provided audit policy recommendations for servers, including a statistical analysis of the differences in the volume of each event generated during a set of common attacks. The goal is to provide security teams with additional tools to tailor their Windows security event audit policy to meet their organization's resource and risk-based needs.

Nicole JeNaye, nicole.jenaye@gmail.com

2. Research Method

The comparison of Microsoft security audit recommendations was conducted using a quantitative, repeatable research method. The virtual lab consisted of networked virtual machines (VMs), including two Windows Server 2016 VMs and one Ubuntu VM running Splunk. For each policy configuration, a set of simple predefined tasks was run in the environment, followed by the detonation of malware samples. After the log collection, each detonation under each configuration was analyzed to identify differences in events generated and the volume of logs.

2.1. Lab Setup and Configuration

The virtual lab environment was built using VMWare Workstation Pro 16.1.2 as the host virtualization software, and HashiCorp Packer and Vagrant as the virtual machine provisioning and configuration tools. The Packer and Vagrant configuration scripts for the lab were adapted from the excellent DetectionLab project created by Chris Long available on Github (Long, 2017). Most of the security tools and monitoring features included in Long's DetectionLab were removed for this testing environment to simplify the environment and avoid generating atypical Windows events. However, the author utilized and modified some of the basic provisioning scripts to create a repeatable architecture for the virtual lab.

The virtual lab consisted of the following virtual machines (VMs):

- One Ubuntu 18.04.5 LTS VM running Splunk Enterprise 8.2.2.1 as the log aggregation and analysis tool, referred to in this paper as the "logger," with a Splunk Dev license installed to allow an adequate license volume of 10GB per day.
- One Windows Server 2016 VM was configured as the domain controller (DC) and is referred to in this paper as "DC." This instance had Windows Defender disabled. It also functioned as the DC for the lab environment and distributed Group Policy Object (GPO) updates to modify the security audit policy on the test VM.

- One Windows Server 2016 VM for testing, referred to throughout this paper as the "test server," with Windows Defender disabled, the Windows firewall completely disabled, and Microsoft Office 365, Google Chrome v. 94.0.4606.71 and 7Zip v. 19.00 installed. This server also had Splunk Universal Forwarder 8.1.0.1 installed, including the Splunk Add-On for Microsoft Windows 7.0.0, configured with the following inputs.conf stanza to collect all generated Windows security events from the test server and forward them to the Splunk logger:

```
[WinEventLog://Security]
disabled = 0
start_from = oldest
current_only = 0
evt_resolve_ad_obj = 1
checkpointInterval = 5
blacklist1 = EventCode="4662" Message="Object Type:(?!\s*groupPolicyContainer)"
blacklist2 = EventCode="566" Message="Object Type:(?!\s*groupPolicyContainer)"
renderXml=true
```

While the DC in this lab was configured with a Splunk Universal Forwarder, the events generated by the DC were not in scope for the analysis of Windows events included in this paper. The virtual lab environment was hardened and isolated from the host using best practices, including guidance provided by Lenny Zelster (Zelster, 2015) (Zelster, 2019), and utilized a custom virtual network to allow communication between the nodes and log collection by the logger. The virtual lab did not include anti-forensics modifications to prevent malware from detecting the virtual environment because the malware samples were not determined to have code for virtualization detection.

3. Test Methodology

To accurately measure the Windows security events generated for each audit policy, iterations of testing were conducted for each audit policy, with each iteration including a baseline of ‘normal activity,’ and detonation of malware samples. All Windows security events generated during these 28 tests were collected and analyzed, including those generated by normal Windows processes during testing.

3.1. Audit Policy Templates

Microsoft provides three security audit templates – Default, Baseline, and Stronger. In addition to these three templates, a fourth template, Maximum, was created with all audit settings fully enabled to generate a control dataset for comparison. The audit policies were distributed to the test server using a Group Policy Object (GPO) configured on the DC in the lab environment. The audit policies were installed on the DC from crafted CSV files using the PowerShell command `Import-GPO -BackupGpoName`, and distributed to the test Windows Server 2016 VM, followed by `gpupdate /force` on the test VM.

3.1.1. Maximum Security Audit Policy

Before testing each of the Microsoft-provided audit policy templates, a full iteration of testing was conducted using a security audit policy with all subcategories enabled and all inclusion settings configured to generate events for both "Success" and "Failure" conditions. This iteration captured a control dataset for all possible Windows security events for comparison and to identify subcategories not included in the Microsoft recommendations which were useful in identifying threat activity. Appendix A contains a complete list of Windows security audit subcategories for reference.

3.1.2. Default Security Audit Policy

The Microsoft Default security audit policy template has 11 subcategories enabled, with four subcategories configured with inclusion settings for Success or Failure, and seven configured with an inclusion setting only for Success. These 11 subcategories enable the potential generation of 97 unique Windows security events. Table 1 depicts the enabled subcategories for this template, with all other subcategories disabled.

Subcategory	Subcategory GUID	Inclusion Setting	Exclusion Setting	Setting Value
Audit User Account Management	{0cce9235-69ae-11d9-bed3-505054503030}	Success		1
Audit Account Lockout	{0cce9217-69ae-11d9-bed3-505054503030}	Success		1
Audit Logoff	{0cce9216-69ae-11d9-bed3-505054503030}	Success		1
Audit Logon	{0cce9215-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Network Policy Server	{0cce9243-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Special Logon	{0cce921b-69ae-11d9-bed3-505054503030}	Success		1
Audit Audit Policy Change	{0cce922f-69ae-11d9-bed3-505054503030}	Success		1
Audit Authentication Policy Change	{0cce9230-69ae-11d9-bed3-505054503030}	Success		1
Audit Other System Events	{0cce9214-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Security State Change	{0cce9210-69ae-11d9-bed3-505054503030}	Success		1
Audit System Integrity	{0cce9212-69ae-11d9-bed3-505054503030}	Success and Failure		3

Table 1. Default Security Audit Policy Settings

3.1.3. Baseline Security Audit Policy

The Microsoft recommended Baseline security audit policy has 15 subcategories enabled, with four subcategories configured with an inclusion setting only for “Success,” and the remaining 11 configured for “Success” and "Failure.” These 15 subcategories may result in the generation of 107 unique Windows security events. The Baseline template should not be considered as building on the Default policy, as the subcategories "Audit Account Lockout," "Audit Network Policy Server," and "Audit Other System Events" are included in the Default template but not in the Baseline template. Table 2 depicts the enabled subcategories and inclusion settings for this template.

Subcategory	Subcategory GUID	Inclusion Setting	Exclusion Setting	Setting Value
Audit Credential Validation	{0cce923f-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Computer Account Management	{0cce9236-69ae-11d9-bed3-505054503030}	Success		1
Audit Other Account Management Events	{0cce923a-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Security Group Management	{0cce9237-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit User Account Management	{0cce9235-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Process Creation	{0cce922b-69ae-11d9-bed3-505054503030}	Success		1
Audit Logoff	{0cce9216-69ae-11d9-bed3-505054503030}	Success		1
Audit Logon	{0cce9215-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Special Logon	{0cce921b-69ae-11d9-bed3-505054503030}	Success		1
Audit Audit Policy Change	{0cce922f-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Authentication Policy Change	{0cce9230-69ae-11d9-bed3-505054503030}	Success		1
Audit IPsec Driver	{0cce9213-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Security State Change	{0cce9210-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Security System Extension	{0cce9211-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit System Integrity	{0cce9212-69ae-11d9-bed3-505054503030}	Success and Failure		3

Table 2. Baseline Security Audit Policy Settings

3.1.4. Stronger Security Audit Policy

The Microsoft-recommended Stronger security audit policy has 22 subcategories enabled, with three configured to generate events on the condition of "Success," and the remaining 19 configured to generate events in the conditions of either “Success” or “Failure.” These 22 subcategories can generate 147 unique Windows security events. The

Stronger audit policy template builds on the Baseline template and includes all subcategories from the Baseline template. Table 3 depicts the enabled subcategories and inclusion settings for this template, with all unlisted subcategories disabled.

Subcategory	Subcategory GUID	Inclusion Setting	Exclusion Setting	Setting Value
Audit Credential Validation	{0cce923f-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Kerberos Authentication Service	{0cce9242-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Kerberos Service Ticket Operations	{0cce9240-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Other Account Logon Events	{0cce9241-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Computer Account Management	{0cce9236-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Other Account Management Events	{0cce923a-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Security Group Management	{0cce9237-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit User Account Management	{0cce9235-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit DPAPI Activity	{0cce922d-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Process Creation	{0cce922b-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Account Lockout	{0cce9217-69ae-11d9-bed3-505054503030}	Success		1
Audit Logoff	{0cce9216-69ae-11d9-bed3-505054503030}	Success		1
Audit Logon	{0cce9215-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Other Logon/Logoff Events	{0cce921c-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Special Logon	{0cce921b-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Audit Policy Change	{0cce922f-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Authentication Policy Change	{0cce9230-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit MPSSVC Rule-Level Policy Change	{0cce9232-69ae-11d9-bed3-505054503030}	Success		1
Audit IPsec Driver	{0cce9213-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Security State Change	{0cce9210-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Security System Extension	{0cce9211-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit System Integrity	{0cce9212-69ae-11d9-bed3-505054503030}	Success and Failure		3

Table 3. Stronger Security Audit Policy Settings

3.2. Control Data

After configuring the audit policy for each iteration, a snapshot of the test Windows Server 2016 VM was created, with the default user logged in locally. The VM was reverted to this snapshot after the baseline activity was recorded and after each sample of malware was tested. The following activities were conducted on the VM for each security audit policy to create a baseline of control data simulating user activity:

1. Open Microsoft Word, type ‘test’ in the new text document, and save to the Desktop with the filename “somedocs.docx”
2. Open Internet Explorer, access the SANS Student Services website (https://www.sans.edu/students/), and download the SANS Technology Institute Student Handbook (https://assets.contentstack.io/v3/assets/blt36c2e63521272fdc/bltecd2ce4eea04ac1a/SANS.edu-Student-Handbook.pdf) as a PDF to the default

Downloads folder, then open the file location, close Windows Explorer and close Internet Explorer.

3. Open Windows Explorer, navigate to the Desktop, and open the file 'somedocs.txt' created earlier in Word, then close Word.
4. Navigate to the Downloads folder, open the STI Student Handbook in Google Chrome, and close Google Chrome.
5. Navigate to the shared folder on the DC containing the compressed, password-protected malware samples.

3.3. Malware Samples

Six malware samples were tested for each of the four audit policies to evaluate the effectiveness of the Windows audit policies for detecting malware. These samples were selected from the list of "Most Seen Malware" as tallied on abuse.ch, operated by Roman Hüssy of the Institute for Cybersecurity Engineering (ICE) at Bern University (abuse.ch, 2021). They were picked to reflect different types of malicious activity and to avoid redundancy. For example, TrickBot was included, but QakBot was not, as both are banking Trojans frequently dropped by Emotet. This is not intended to indicate the unincluded malware is not a threat but rather to utilize a varied set of samples for analysis.

The following six malware samples were obtained from MalwareBazaar (https://bazaar.abuse.ch) and tested, listed below with a short description:

- AgentTelsa – a remote access trojan (RAT) and keylogger used to steal credentials and other sensitive information. Newer versions can steal the clipboard's contents and target a wide variety of applications (O'Donnell, 2021).
- CoinMiner XMRig – cryptojacking malware that consumes CPU and memory resources on the infected host to mine cryptocurrencies (Kessem, 2018).

- Dridex – a modular trojan originally designed to steal banking credentials, with modern versions also capable of deploying other malware, including ransomware (Didier, 2021).
- RedLineStealer – an information stealer identified in 2020 and available for sale on Russian underground forums, according to Proofpoint malware researchers (Proofpoint, 2020).
- Ryuk – ransomware frequently deployed after TrickBot and often the last step of an attack, resulting in encryption of critical data, with the decryption key ransomed to the victim (Gallagher, 2020)
- TrickBot - Began as a banking trojan that redirected users to fake sites to steal login credentials but has expanded to include many other functions, including webinjects for stealing banking credentials during legitimate logins, lateral propagation, and more (Seals, 2021).

During testing, the VM was reverted to a clean snapshot, then the malware compressed file (.zip) was copied from the shared folder on the lab DC to the desktop of the test VM, decompressed, and executed with administrative privileges. The sample was allowed to run until the generation of Windows events ceased, as monitored via Splunk.

4. Findings and Discussion

After the testing iterations, all Windows security events directly generated by Splunk or VMWare Tools were removed from the sample data to avoid skewing the results with lab-specific variables. The resulting clean dataset contained 10,228 Windows security events generated during testing. The average byte length was calculated for each EventID to determine potential licensing and storage resource consumption. Table 4 lists every EventID generated during the testing phase, with the count of events across all iterations and the average byte length of each EventID.

EventID	count	avg_bytes
4611	35	849
4624	35	1774
4625	14	1467
4656	4326	1696
4658	1021	1033
4663	207	1303
4670	31	1291
4672	35	1162
4673	58	1037
4688	656	1365
4689	144	898
4690	467	946
4698	1	3594
4797	69	960
4799	12	1049
4946	4	790
4948	4	790
4956	7	677
4957	8	808
5058	6	1294
5059	6	1118
5061	18	1118
5140	4	1050
5145	6	1184
5152	20	1017
5156	1828	1099
5157	20	1100
5158	955	902
5447	1750	2392

Table 4. EventID count and average byte length.

The volume of each EventID was used to calculate the resource ramifications of enabling specific Windows security audit. Note these Windows events were generated in XML format, so the average event sizes likely differ from Windows events generated in the conventional format. Also, note the average size of Windows events will vary somewhat, particularly those that contain full command line strings or other highly variable data.

4.1. Technical Data Review

To analyze the data, the events collected for each audit policy were analyzed and summarized for each activity tested.

4.1.1. Control Data

A set of activities simulating normal user activities was conducted for each audit policy and the events collected. This activity did not include any malicious activity or malware samples. Table 5 lists all unique EventIDs generated during this activity for each policy, as well as the count of events, and the total length of the events in bytes and kilobytes (KB).

Default Policy			
EventID	count	total_bytes	total_kb
5058	1	1294	1.26
5059	1	1118	1.09
5061	1	1118	1.09
Total	**3**	**3530**	**3.45**

Baseline Policy			
EventID	count	total_bytes	total_kb
4688	28	38220	37.32
Total	**28**	**38220**	**37.32**

Stronger Policy			
EventID	count	total_bytes	total_kb
4688	29	39585	38.66
4946	2	1580	1.54
4948	2	1580	1.54
Total	**33**	**42745**	**41.74**

Maximum Policy			
EventID	count	total_bytes	total_kb
4656	12	20352	19.88
4658	18	18594	18.16
4670	20	25820	25.21
4673	4	4148	4.05
4688	27	36855	35.99
4689	25	22450	21.92
4690	6	5676	5.54
4698	1	3594	3.51
4946	2	1580	1.54
4948	2	1580	1.54
5156	608	668192	652.53
5158	284	256168	250.16
5447	376	899392	878.31
Total	**1385**	**1964401**	**1918.36**

Table 5. Control activity data summary.

The Default policy generated only three security events for the control activity, one each for EventIDs 5058, 5059, and 5061. These events were generated under the "System Integrity" and "Other System Events" subcategories and relate to cryptography key operations. The Baseline policy generated 28 events, all of which were 4688 (Process Creation) events. These events contain detailed information on new processes, including the ParentProcessName, ProcessCommandLine, and the Creator and Target accounts, which can be critical data points for detecting or investigating threat activity. The Stronger policy also generated two of EventIDs 4946 and 4948, caused by changes to the Windows firewall. The Maximum policy generated 1385 events, with a total volume of 45 times greater the volume generated by the Stronger policy.

Nicole JeNaye, nicole.jenaye@gmail.com

4.1.2. AgentTesla

A sample of malware categorized as AgentTesla, with the SHA256 hash value 867ffa9dacdc4e809ac75fdbb1405f0b5b5e51e375901c838637ab6c36eb9356, was executed under each of the four policies. Table 6 summarizes the Windows security events generated during the execution of the AgentTesla sample.

Default Policy			
EventID	count	total_bytes	total_kb
4624	2	3548	3.46
4672	2	2324	2.27
4797	3	2880	2.81
5058	1	1294	1.26
5059	1	1118	1.09
5061	1	1118	1.09
Total	**10**	**12282**	**11.99**

Baseline Policy			
EventID	count	total_bytes	total_kb
4611	2	1698	1.66
4624	2	3548	3.46
4672	2	2324	2.27
4688	23	31395	30.66
4797	3	2880	2.81
5061	1	1118	1.09
Total	**33**	**42963**	**41.96**

Stronger Policy			
EventID	count	total_bytes	total_kb
4611	2	1698	1.66
4624	2	3548	3.46
4625	2	2934	2.87
4672	2	2324	2.27
4688	22	30030	29.33
4799	2	2098	2.05
5061	1	1118	1.09
Total	**33**	**43750**	**42.72**

Maximum Policy			
EventID	count	total_bytes	total_kb
4611	2	1698	1.66
4656	23	39008	38.09
4658	31	32023	31.27
4663	8	10424	10.18
4673	9	9333	9.11
4688	13	17745	17.33
4689	15	13470	13.15
4690	10	9460	9.24
4797	3	2880	2.81
5152	6	6102	5.96
5156	65	71435	69.76
5157	6	6600	6.45
5158	81	73062	71.35
Total	**272**	**293240**	**286.37**

Table 6. AgentTesla malware sample data summary.

The detonation of the AgentTesla sample generated ten events under the Default policy. EventIDs 5058, 5059, and 5061 were identical to those generated during the control activity. The generation of EventID 4797 ("An attempt was made to query the existence of a blank password for an account") appears in both the Default and Baseline

Policy logs, but this event is not associated with malicious activity (Franklin, n.d.). The Baseline and Stronger policies generated the same number of events (33). In contrast, the Stronger policy generated EventIDs 4799 ("A security-enabled local group membership was enumerated") and 4625 ("An account failed to log on"), in addition to those generated under the Baseline policy. However, neither of these EventIDs appear in the Maximum policy dataset, and a review of the logs does not indicate these events were caused by malicious activity. EventID 4611 appeared in all the datasets and was generated by 7Zip during decompression of the malware sample and is not associated with malicious activity.

The remaining EventID 4688 ("A new process has been created") provides critical details about the execution of malware in the environment. The volume of EventID 4688 events generated during normal activity can be prohibitive without additional filtering. For cleverly named malware, a manual review of EventID 4688 is unlikely to be an effective detection method.

The Maximum policy generated one notable event not generated under the other policies, EventID 4663 (“An attempt was made to access an object”). A review of the logs shows the targeted object was `lsass.exe`, also known as Local Security Authority Subsystem Service (LSASS), the Windows process that handles user logon authentication. This activity was directly associated with the malware sample execution and could be used for detecting malicious activity if enabled. According to Microsoft, EventID 4663 has "little to no security relevance" due to the volume and difficulty analyzing without additional context (Simpson, 2021a).

4.1.3. CoinMiner XMRig

A sample of malware categorized as CoinMiner identified by the SHA256 hash value 4d265a1ee6dd0bdccd7e31fce027ccd42f1e19c09a92e911fba7db7696698b4d was executed under each of the four policies. Table 7 describes the Windows security events generated during the execution of this sample.

Default Policy			
EventID	count	total_bytes	total_kb
4624	2	3548	3.46
4625	2	2934	2.87
4672	2	2324	2.27
4797	3	2880	2.81
5058	1	1294	1.26
5059	1	1118	1.09
5061	1	1118	1.09
Total	12	15216	14.86

Baseline Policy			
EventID	count	total_bytes	total_kb
4611	2	1698	1.66
4624	2	3548	3.46
4672	2	2324	2.27
4688	29	39585	38.66
4797	3	2880	2.81
5061	1	1118	1.09
Total	39	51153	49.95

Stronger Policy			
EventID	count	total_bytes	total_kb
4611	2	1698	1.66
4624	2	3548	3.46
4625	2	2934	2.87
4672	2	2324	2.27
4688	30	40950	39.99
4797	3	2880	2.81
4799	2	2098	2.05
5061	1	1118	1.09
Total	44	57550	56.2

Maximum Policy			
EventID	count	total_bytes	total_kb
4611	1	849	0.83
4656	5	8480	8.28
4658	10	10330	10.09
4663	1	1303	1.27
4670	3	3873	3.78
4673	4	4148	4.05
4688	7	9555	9.33
4689	11	9878	9.65
4690	5	4730	4.62
5156	51	56049	54.74
5158	31	27962	27.31
Total	129	137157	133.94

Table 7. CoinMiner malware sample data summary.

The profile of events generated during the CoinMiner detonation are similar to those generated during the AgentTesla detonation. The event count is slightly higher, but the EventID of utility is again 4688 ("A new process has been created"). The Maximum policy generated approximately 3 times the count and log volume but did not generate any additional EventIDs of utility.

4.1.4. Dridex

A sample of malware categorized as Dridex, with SHA256 hash value 347d827101bbb1fc114fe06e705670046350338683e3cea6643ab55b9e0f6558 was executed under each of the four audit policies. Table 8 summarizes the EventIDs generated during the test execution.

Default Policy			
EventID	count	total_bytes	total_kb
4624	2	3548	3.46
4672	2	2324	2.27
4797	3	2880	2.81
5058	1	1294	1.26
5059	1	1118	1.09
5061	1	1118	1.09
Total	10	12282	11.99

Baseline Policy			
EventID	count	total_bytes	total_kb
4611	2	1698	1.66
4624	2	3548	3.46
4672	2	2324	2.27
4688	13	17745	17.33
4797	3	2880	2.81
5061	1	1118	1.09
Total	23	29313	28.63

Stronger Policy			
EventID	count	total_bytes	total_kb
4611	1	849	0.83
4624	2	3548	3.46
4625	2	2934	2.87
4672	2	2324	2.27
4688	22	30030	29.33
4799	2	2098	2.05
5061	1	1118	1.09
Total	32	42901	41.9

Maximum Policy			
EventID	count	total_bytes	total_kb
4611	2	1698	1.66
4656	18	30528	29.81
4658	18	18594	18.16
4673	7	7259	7.09
4688	11	15015	14.66
4689	12	10776	10.52
4797	3	2880	2.81
5156	61	67039	65.47
5158	63	56826	55.49
5447	42	100464	98.11
Total	237	311079	303.79

Table 8. Dridex malware sample data summary.

The Windows security events generated during the Dridex malware execution maintain a similar profile to those generated under the AgentTesla and CoinMiner malware execution tests. The notably helpful EventID is again 4688, which details process creation, including malicious processes created with the Dridex executable as the parent process. Under the Maximum policy, there were 42 EventID 5447 ("A Windows Filtering Platform filter has been changed") events generated. According to Microsoft, this event is typically generated during Group Policy Object (GPO) updates but has "little to no security relevance" (Simpson, 2021b). However, the appearance of this event outside of a GPO update may be used for detecting changes made to the Windows Filtering Platform by malware.

4.1.5. RedlineStealer

A malware sample categorized as RedlineStealer, with SHA256 hash value 6505008c814246965748bdbfe7c034fcab75cc435a66b6ccfdd366927befb6ed, was executed under each of the four audit policies. Table 9 displays a summary of the EventIDs generated during this test.

Default Policy			
EventID	count	total_bytes	total_kb
4624	2	3548	3.46
4672	2	2324	2.27
4797	3	2880	2.81
5058	1	1294	1.26
5059	1	1118	1.09
5061	1	1118	1.09
Total	10	12282	11.99

Baseline Policy			
EventID	count	total_bytes	total_kb
4611	2	1698	1.66
4624	2	3548	3.46
4672	2	2324	2.27
4688	23	31395	30.66
4797	3	2880	2.81
5061	1	1118	1.09
Total	33	42963	41.96

Stronger Policy			
EventID	count	total_bytes	total_kb
4611	2	1698	1.66
4624	2	3548	3.46
4625	2	2934	2.87
4672	2	2324	2.27
4688	37	50505	49.32
4797	3	2880	2.81
4799	2	2098	2.05
5061	1	1118	1.09
Total	51	67105	65.53

Maximum Policy			
EventID	count	total_bytes	total_kb
4611	2	1698	1.66
4656	15	25440	24.84
4658	15	15495	15.13
4673	8	8296	8.1
4688	13	17745	17.33
4689	14	12572	12.28
4797	3	2880	2.81
5156	74	81326	79.42
5158	65	58630	57.26
5447	42	100464	98.11
Total	251	324546	316.94

Table 9. RedlineStealer malware sample data summary

The volume of events and EventIDs generated for the RedlineStealer malware was consistent with the previously discussed samples. The Maximum policy includes EventIDs 5156 (“The Windows Filtering Platform has allowed a connection”) and 5158 (“The Windows Filtering Platform has permitted a bind to a local port”) that could potentially be used for identifying connections to malicious IPs. Still, Windows generates a high volume of these events for routine processes, so collection would be resource intensive.

4.1.6. Ryuk

A sample of Ryuk malware with SHA256 hash value 180f82bbedb03dc29328e32e054069870a1e65078b78b2120a84c96aaed7d843 was executed under each of the four audit policies. Table 10 summarizes the EventIDs generated during this test.

Default Policy			
EventID	count	total_bytes	total_kb
4688	16	21840	21.33
4797	3	2880	2.81
Total	19	24720	24.14

Baseline Policy			
EventID	count	total_bytes	total_kb
4611	1	849	0.83
4624	2	3548	3.46
4672	2	2324	2.27
4688	57	77805	75.98
Total	62	84526	82.54

Stronger Policy			
EventID	count	total_bytes	total_kb
4611	1	849	0.83
4624	1	1774	1.73
4672	1	1162	1.13
4688	32	43680	42.66
Total	35	47465	46.35

Maximum Policy			
EventID	count	total_bytes	total_kb
4611	1	849	0.83
4656	4162	7058752	6893.31
4658	826	853258	833.26
4663	198	257994	251.95
4670	8	10328	10.09
4673	4	4148	4.05
4688	24	32760	31.99
4689	21	18858	18.42
4690	434	410564	400.94
5140	4	4200	4.1
5145	6	7104	6.94
5152	2	2034	1.99
5156	577	634123	619.26
5157	2	2200	2.15
5158	37	33374	32.59
Total	6306	9330546	9111.86

Table 10. Ryuk malware sample data summary

The profile of events generated during the Ryuk malware detonation is notably smaller than for other malware samples under the three recommended audit policies.

Excluding the events generated by the decompression and execution of the malware sample consistently seen in all datasets, there are essentially no events generated that could be used to detect Ryuk ransomware execution in the environment other than EventID 4688. However, for the Maximum audit policy, the volume of events is enormous, nearly 200 times the volume and count generated under the Stronger policy. Most of this volume results from EventIDs 4656 ("A handle to an object was requested"), followed by 4658 ("The handle to an object was closed"), 5156 ("The Windows Filtering Platform has allowed a connection"), and 4690 ("An attempt was made to duplicate a handle to an object").

These events are not included any of the three Microsoft security audit policy recommendations but a spike in volume was observed immediately after the Ryuk malware sample was detonated, before it was apparent a ransomware attack was occurring. According to Microsoft, EventID 4656 also has "little to no security relevance" (Simpson, 2021c) due to the volume and difficulty in parsing. However, particularly with this sample, where the three recommended audit policies generated no security events of use, a nascent ransomware infection could potentially be detected early through the identification of a sudden and excessive spike in the volume of 4656 events.

4.1.7. TrickBot

A sample of TrickBot malware with SHA256 hash value 3cdd741fb186596e2c6654737c86f6143b6c130a3f51333346e252f74dafc78e was executed under each of the four audit policies. Table 11 lists the EventIDs generated during this execution.

Default Policy			
EventID	count	total_bytes	total_kb
4624	2	3548	3.46
4672	2	2324	2.27
4797	3	2880	2.81
Total	7	8752	8.55

Baseline Policy			
EventID	count	total_bytes	total_kb
4611	2	1698	1.66
4624	2	3548	3.46
4672	2	2324	2.27
4688	24	32760	31.99
4797	3	2880	2.81
5061	1	1118	1.09
Total	34	44328	43.29

Stronger Policy			
EventID	count	total_bytes	total_kb
4611	2	1698	1.66
4624	2	3548	3.46
4625	2	2934	2.87
4672	2	2324	2.27
4688	38	51870	50.65
4797	3	2880	2.81
4799	2	2098	2.05
5061	1	1118	1.09
Total	52	68470	66.87

Maximum Policy			
EventID	count	total_bytes	total_kb
4611	2	1698	1.66
4656	27	45792	44.72
4658	35	36155	35.31
4673	7	7259	7.09
4688	15	20475	20
4689	17	15266	14.91
4690	8	7568	7.39
4797	3	2880	2.81
4956	2	1354	1.32
4957	4	3232	3.16
5152	3	3051	2.98
5156	152	167048	163.13
5157	3	3300	3.22
5158	143	128986	125.96
5447	624	1492608	1457.63
Total	1045	1936672	1891.28

Table 11. Trickbot malware sample data summary.

For this sample, the Default policy did not generate any events related to the malware activity after execution. In contrast, the Baseline and Default policies generated a similar profile of events to previous malware samples. As seen in two other samples, under the Maximum policy, a significant volume of EventID 5447 events were generated outside of a GPO update, which could be used to detect malware processes making changes to the Windows firewall.

Nicole JeNaye, nicole.jenaye@gmail.com

4.1.8. Analysis Summary

The 28 tests conducted (six malware samples and one control activity for each of the four evaluated policies) resulted in the collection of 10,228 Windows security events. Table 12 summarizes the total count and volume of events generated for the Microsoft-provided Default, Baseline, and Stronger audit policies.

Default Policy		
EventID	count	total_kb
4624	10	17.32
4625	2	2.87
4672	10	11.35
4688	16	21.33
4797	18	16.88
5058	5	6.32
5059	5	5.46
5061	5	5.46
Total	**71**	**86.99**

Baseline Policy		
EventID	count	total_kb
4611	11	9.12
4624	12	20.79
4672	12	13.62
4688	197	262.6
4797	15	14.06
5061	5	5.46
Total	**252**	**325.65**

Stronger Policy		
EventID	count	total_kb
4611	10	8.29
4624	11	19.06
4625	10	14.33
4672	11	12.48
4688	210	279.93
4797	9	8.44
4799	10	10.24
4946	2	1.54
4948	2	1.54
5061	5	5.46
Total	**280**	**361.31**

Table 12. Summary of events collected for Default, Baseline, and Stronger audit policies

The Default policy generates between 25% of the number and byte volume of events generated by the Baseline and Stronger policies but did not provide demonstrable value in detecting or investigating the malware samples tested in this research. While the volume of logs generated is lower, the minimal value added by this policy does not justify even the lower resource usage. The Baseline and Stronger audit policies generated similar counts and volumes of events, with the Stronger policy generating approximately 10% more count and byte volume of events.

The Maximum policy was tested to provide a baseline for comparison, and Table 13 summarizes the event counts and volumes generated under this policy. As expected, the count and volume of events generated under this policy are significantly higher than the other policies. The total count and byte volume of events are approximately 35 times more than was generated under the Stronger policy, which had the following highest amount.

The bulk of these events were generated during the Ryuk ransomware test, as the ransomware was requesting handles (EventID 4656 and EventID 4658) while encrypting the contents of the drive.

Maximum Policy					
EventID	count	total_kb	EventID	count	total_kb
4611	10	8.29	4946	2	1.54
4656	4262	7058.94	4948	2	1.54
4658	953	961.38	4956	2	1.32
4663	207	263.4	4957	4	3.16
4670	31	39.08	5140	4	4.1
4673	43	43.55	5145	6	6.94
4688	110	146.63	5152	11	10.92
4689	115	100.85	5156	1588	1704.31
4690	463	427.73	5157	11	11.82
4698	1	3.51	5158	704	620.13
4797	12	11.25	5447	1084	2532.16
Total				**9625**	**13962.55**

Table 13. Summary of events collected under Maximum audit policy

5. Recommendations and Implications

The Microsoft-provided Baseline and Stronger Audit Policy Recommendations provide a good starting point for crafting a logging strategy. Still, based on the testing conducted for this paper, these policy templates can be improved upon to generate additional Windows security events that may be useful for detecting and investigating malicious activity in a Windows environment.

The Default policy did not produce meaningful events that could be used to detect or investigate any of the six malware samples tested for this paper and was determined to consume logging resources, albeit reduced, without providing value. If Windows events are to be generated and collected, it is recommended to apply any broader audit policy to gain value from Windows security events, rather than rely on the Default policy.

The Baseline and Stronger policies performed similarly during testing in generating events useful for direct detection of malicious activity, with the primary event of interest being EventID 4688 ("A new process was created"). Given the nominal difference in license and storage requirements, the Stronger policy is recommended over the Baseline, as in some cases, the additional events may provide helpful context for detections and investigations.

Despite the oft-heard refrain "log all the things," the volume of Windows security events generated under the Maximum policy highlights the impracticality of that strategy.

However, several events logged under the Maximum policy were not included in the Microsoft audit recommendations that may be considered for logging.

5.1. Recommendations for Practice

Based on the research performed, the events generated under the Default policy were not useful in detecting or investigating any of the six malware samples tested. Therefore, security professionals should not rely on the Default security audit recommendations in a live Windows environment. Given the nominal difference in resource requirements between the Baseline and Stronger templates, the Stronger template is recommended as a starting point. Regardless of which template is used, it is strongly recommended EventID 4688 is enabled. This event was identified in each sample test as providing critical information about malicious activity in the environment. The average byte length of EventID 4688 was 1365, so it is a relatively large event compared with others collecting during this research. However, the value of the information in this event justifies the additional the resource consumption. To generate EventID 4688, enable audit subcategory “Process Creation”, which also results in the generation of EventID 4696 (“A primary token was assigned to the process”).

In addition to the subcategories enabled in the Stronger template, it is recommended that security professionals consider enabling EventID 4656 or EventID 4663. These events can be monitored for early detection of ransomware in the environment. The average byte length of EventID 4656 in the collected datasets was 1696 bytes, while for EventID 4663, the average byte length was 1303, both of which are relatively large. However, since the detection relies on identifying extreme spikes in the number of events generated, not on the content of the event messages, a filtered or truncated version of these events could be collected for statistical analysis only. The method for filtering for volume reduction depends on the logging solution used and is beyond the scope of this paper.

The generation of EventID 4656 requires enabling audit subcategory "Handle Manipulation," which will also generate EventIDs 4658 and 4690. The generation of EventID 4663 requires enabling audit subcategory “Kernel”, which also enables generation of EventIDs 4659 (“A handle to an object was requested with intent to

delete"), 4660 ("An object was deleted"), and 4661 ("A handle to an object was requested").

It should also be noted the Maximum audit policy tested in this research included enabling the File System auditing subcategory, but this subcategory does not generate file system audit events unless the policy has been applied to specific files and folders. The malware samples tested in this research did not lend themselves to monitoring individual files, so auditing settings were not applied to specific files or folders. If the File System auditing subcategory is enabled and applied to specific files or folders, the volume of file system auditing events, including EventIDs 4656, 4658, 4660 ("An object was deleted"), and 4663 would increase relative to the number of files and folders being monitored and the frequency of access by processes and users.

5.2. Implications for Future Research

The scope of this research paper included six malware samples and a relatively small virtual test environment. The research results provide a starting point and a methodology for determining what security events can be used for identifying malicious activity and thus how to customize the audit policy for a specific environment. The variety of current threats and variability in environments limits the applicability of the results of this research. For security professionals, conducting similar testing in a lab environment that simulates the organization they are tasked with securing would assist in tailoring a security audit policy best suited for their environment, rather than relying on static recommendations.

6. Conclusion

There are various log sources available to security professionals, but in a Windows environment, the ability to generate Windows security events is a convenient and valuable option for monitoring endpoints. For security professionals concerned with using logs to detect and investigate threats, taking the time to test and customize the Windows security audit policy can transform a logging license and storage glut into an invaluable security resource.

References

Didier, A. (2021, April 21). *Dridex*. Red Canary. Retrieved September 17, 2021 from https://redcanary.com/threat-detection-report/threats/dridex/

Foulds, I., et al (2017, May 31), *Audit policy recommendations*. Microsoft Docs. Retrieved August 15, 2021, from https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/audit-policy-recommendations

Franklin, R. (n.d.) *Windows security log event ID 4797 - An attempt was made to query the existence of a blank password for an account.* (n.d.). Randy Franklin Smith's Ultimate Windows Security. Retrieved September 29, 2021 from https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/event.aspx?eventid=4797

Gallagher, S. (2020, October 28). *They're back: Inside a new Ryuk ransomware attack.* Sophos News. https://news.sophos.com/en-us/2020/10/14/inside-a-new-ryuk-ransomware-attack/

Keizer, G. (2020, November 4). *Windows by the numbers: Windows 10 rolls on past 70%.* ComputerWorld. Retrieved August 11, 2021 from https://www.computerworld.com/article/3199373/windows-by-the-numbers-windows-10-rolls-on-past-70.html

Kessem, L., (2018, February 26). *XMRig: Father Zeus of Cryptocurrency Mining Malware?* SecurityIntelligence. Retrieved September 8, 2021 from https://securityintelligence.com/xmrig-father-zeus-of-cryptocurrency-mining-malware/

Nicole JeNaye, nicole.jenaye@gmail.com

O'Donnell, L. (2021, February 2). *Agent Tesla Trojan 'Kneecaps' Microsoft's anti-malware interface*. Threatpost. Retrieved September 9, 2021 from https://threatpost.com/agent-tesla-microsoft-asmi/163581/

Proofpoint (2020, August 26). *New RedLine password stealer virus insights*. Retrieved September 18, 2021 from https://www.proofpoint.com/us/blog/threat-insight/new-redline-stealer-distributed-using-coronavirus-themed-email-campaign

Seals, T. (2021, July 2). *TrickBot spruces up its banking Trojan module*. Threatpost. https://threatpost.com/trickbot-banking-trojan-module/167521/

Simpson, D. (2021, September 7). *4663(S) an attempt was made to access an object.* Microsoft Docs. Retrieved October 5, 2021 from https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4663

Simpson, D., et al (2021, September 7). *4656(S, F) a handle to an object was requested.* Microsoft Docs. Retrieved October 6, 2021 from https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4656

Simpson, D., et al (2021, September 8). *5447(s) a windows filtering platform filter has been changed*. Microsoft Docs. Retrieved October 5, 2021, from https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-5447

Zelster, L. (2015, February 9) *Virtualized Network Isolation for a Malware Analysis Lab.* Retrieved August 13, 2021 from https://zeltser.com/vmware-network-isolation-for-malware-analysis/

Zelster, L. (2019, March 4) *How to get and set up a free Windows VM for malware analysis*. Retrieved August 13, 2021 from https://zeltser.com/free-malware-analysis-windows-vm/

Nicole JeNaye, nicole.jenaye@gmail.com

Appendix A
Complete List of Windows Audit Security Subcategories

Subcategory	Subcategory GUID	Inclusion Setting	Exclusion Setting	Setting Value
Audit Credential Validation	{0cce923f-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Kerberos Authentication Service	{0cce9242-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Kerberos Service Ticket Operations	{0cce9240-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Other Account Logon Events	{0cce9241-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Computer Account Management	{0cce9236-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Distribution Group Management	{0cce9238-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Other Account Management Events	{0cce923a-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Security Group Management	{0cce9237-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit User Account Management	{0cce9235-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit DPAPI Activity	{0cce922d-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Process Creation	{0cce922b-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Process Termination	{0cce922c-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit RPC Events	{0cce921b-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Detailed Directory Service Replication	{0cce923e-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Directory Service Access	{0cce923b-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Directory Service Changes	{0cce923c-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Directory Service Replication	{0cce923d-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Account Lockout	{0cce9217-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit User / Device Claims	{0cce9247-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Logoff	{0cce9216-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Logon	{0cce9215-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Network Policy Server	{0cce9243-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Other Logon/Logoff Events	{0cce921c-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Special Logon	{0cce921b-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Application Generated	{0cce9222-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Certification Services	{0cce9221-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Detailed File Share	{0cce9244-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit File Share	{0cce9224-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit File System	{0cce921d-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Filtering Platform Connection	{0cce9226-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Filtering Platform Packet Drop	{0cce9225-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Handle Manipulation	{0cce9223-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Kernel Object	{0cce921f-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Other Object Access Events	{0cce9227-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Registry	{0cce921e-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Removable Storage	{0cce9245-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit SAM	{0cce9220-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Central Access Policy Staging	{0cce9246-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Audit Policy Change	{0cce922f-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Authentication Policy Change	{0cce9230-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Authorization Policy Change	{0cce9231-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Filtering Platform Policy Change	{0cce9233-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit MPSSVC Rule-Level Policy Change	{0cce9232-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Other Policy Change Events	{0cce9234-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Non Sensitive Privilege Use	{0cce9229-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Other Privilege Use Events	{0cce922a-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Sensitive Privilege Use	{0cce9228-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit IPsec Driver	{0cce9213-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Other System Events	{0cce9214-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Security State Change	{0cce9210-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit Security System Extension	{0cce9211-69ae-11d9-bed3-505054503030}	Success and Failure		3
Audit System Integrity	{0cce9212-69ae-11d9-bed3-505054503030}	Success and Failure		3

Nicole JeNaye, nicole.jenaye@gmail.com