Global Information Assurance Certification Paper

Attackers Inside the Walls: Detecting Malicious Activity

GIAC GCIA Gold Certification

Author: Sean D. Goodwin, SeanGoodwin@protonmail.ch
Advisor: *Johannes Ullrich*
Accepted: June 7, 2019

Abstract

Small and medium-sized businesses (SMBs) do not always have the budget for an advanced intrusion detection system (IDS) technology. Open-source software can fill this gap, but these free solutions may not provide full coverage for known attacks, especially once the attacker is inside the perimeter. This paper investigates the IDS capabilities of a stand-alone Security Onion device when combined with built-in event logging in a small Windows environment to detect malicious actors on the internal network.

1. Introduction

Small and medium-sized businesses are frequently targeted for computer and data breach attacks due to limited resources and defensive expertise. According to the 2018 Verizon Data Breach Investigations Report, 50% of breach victims were categorized as small businesses (*2018 Data Breach Investigations Report*). Additionally, 68% of breaches took "months or longer to discover" (*2018 Data Breach Investigations Report*). To make matters worse, a large percentage of small and medium-sized businesses (SMBs) identify restricted budgets as the greatest challenge to security (Untangle, n.d.). Another significant concern identified in the survey was not having enough staff to "monitor and manage security" (Untangle, n.d.).

Many organizations have invested significant efforts into the traditional border defensive controls, but have ignored the possibility of a malicious actor existing inside of that trusted boundary. According to the "Defending Against the Wrong Enemy: 2017 SANS Insider Threat Survey", the majority of those surveyed do not believe their processes are effective enough to detect an insider threat (Cole, 2017). This opinion is important, as this insider threat concern primarily focuses on compromising sensitive data such as PHI, PII, or privileged credentials (Cole, 2017). These are the same data types we have seen targeted in attacks from outsiders. In 2018, the top four data types compromised in attacks included: personal, payment, medical, and credentials (*2018 Data Breach Investigations*). These statistics combine to paint a bleak picture for the hopes of securing an SMB environment.

Identifying a toolset that minimizes cost and complexity while providing actionable alerts will enable an SMB to reduce the time required to identify a breach. Security Onion is "a free and open source Linux distribution for intrusion detection, enterprise security monitoring, and log management" (Security Onion, n.d.). The focus of the Security Onion device in this case study is to collect and analyze the host alert data coming from Windows Event Logs. This distribution also includes tools and capabilities that go beyond the scope of this paper.

The lab environment used for this research focused on a Windows domain environment due to Windows holding over 80% of the market share (Operating System Market Share, n.d.). The Wazuh (forked from OSSEC) agent was installed on all Windows hosts to transmit the logs to the Security Onion device (Wazuh - The Open Source Security Platform, n.d.). This utility is used over other options due to its easy installation and native support with Security Onion.

Sean D. Goodwin, SeanGoodwin@protonmail.ch

Microsoft Sysmon was also installed to ensure the appropriate event log details were provided for analysis (Sysmon - Windows Sysinternals, 2019). An installation procedure document can be found in this GitHub repository (https://github.com/0xSeanG/ISE5501). This configuration was chosen so that it aligns with the theory of simplifying the process to detect and respond to threats inside the network. These references were included as best practice starting guides and were implemented as such.

In addition to Microsoft Sysmon, the Windows Advanced Audit Policy was implemented via Group Policy Objects (GPOs). The scope of logging was based on the Malware Archeology *Windows Logging Cheat Sheet* (WINDOWS LOGGING CHEAT SHEET - Win 7 thru Win 2019, 2019). This Audit Policy configuration was chosen, as the creators have mapped the MITRE ATT&CK framework to their proposed logging scope (*WINDOWS ATT&CK LOGGING CHEAT SHEET - Win 7 - Win 2012, 2018*).

A high-level network diagram of the environment used for testing is shown in Figure 1 below. The Caldera tool from MITRE was used to simulate various attack techniques, which is discussed further in Section 3, "Malicious Actor Activity," below.

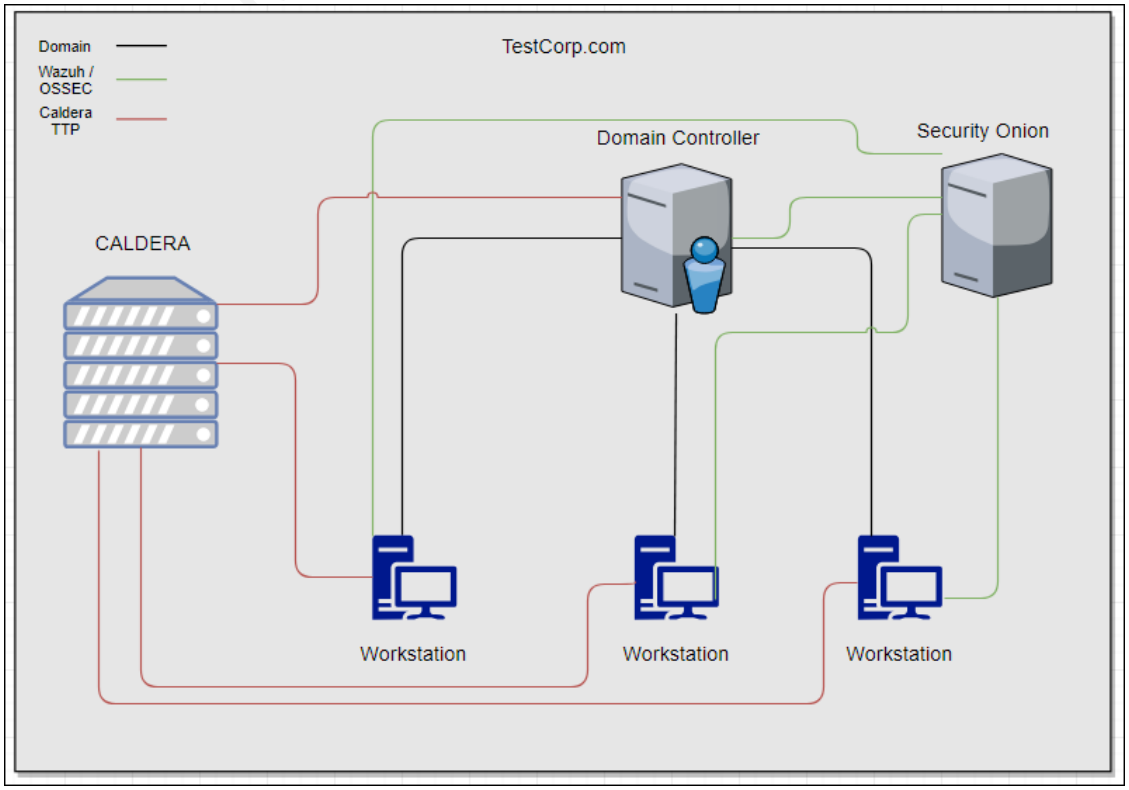


Figure 1- Lab Network Diagram

Two other no-cost event log forwarding utilities were considered for this project: Logstash (Logstash, n.d.) and NXLog (NXLog, n.d.). Logstash is a utility created by Elastic, who also maintains some of the tools included in the Security Onion distribution. NXLog was considered the utility for collecting and centralizing the Windows event logs. While there may be benefits in a more complex environment, the configuration of these two utilities was more complicated than Sysmon and Wazuh. This paper is focused on maintaining simplicity due to the limited resources of SMBs.

This project seeks to determine if it is possible to tune Security Onion and Windows logging events to effectively detect the Tactics, Techniques, and Procedures (TTPs) of attackers inside a network, without introducing unnecessary complexity to an SMB network.

2. Obtaining the necessary data

The first step in detecting these attackers is implementing an appropriate Audit Policy for the TestCorp domain. We want to capture evidence of the attacker, but we do not want to lose these indicators in the noise of an overly-verbose Audit Policy. Malware Archaeology, LLC has released a recommended configuration for the Windows Advanced Audit Policy (WINDOWS LOGGING CHEAT SHEET - Win 7 thru Win 2019, 2019) as well as a mapping of Windows Event IDs to the MITRE ATT&CK Framework (*WINDOWS ATT&CK LOGGING CHEAT SHEET - Win 7 - Win 2012, 2018).* These settings were deployed to the TestCorp domain via Group Policy Object (GPO).

To capture the appropriate level of detail in event logs, Sysmon was installed on all Windows hosts (Sysmon - Windows Sysinternals, 2019). By following the Security Onion installation directions (Sysmon¶, n.d.), the SwiftOnSecurity Sysmon configuration file was used (Sysmon-Config, n.d.). Sysmon was not customized to the TestCorp network since we are working with limited resource availability. This configuration was used “as-is” in the hopes of being sufficient to aid in detecting malicious activity even if it is not perfect for TestCorp without significant investment of time in customizing the rules. This will aid in testing the configuration as an easy solution for SMBs to use out-of-the-box.

Security Onion brings many tools to the table when it comes to detecting malicious activity in the network, many of which are outside the scope of this paper. The major toolset

excluded from this research is the various Network Intrusion Detection (NIDS) capabilities. A full listing of the Kibana Dashboards is shown in Figure 2.

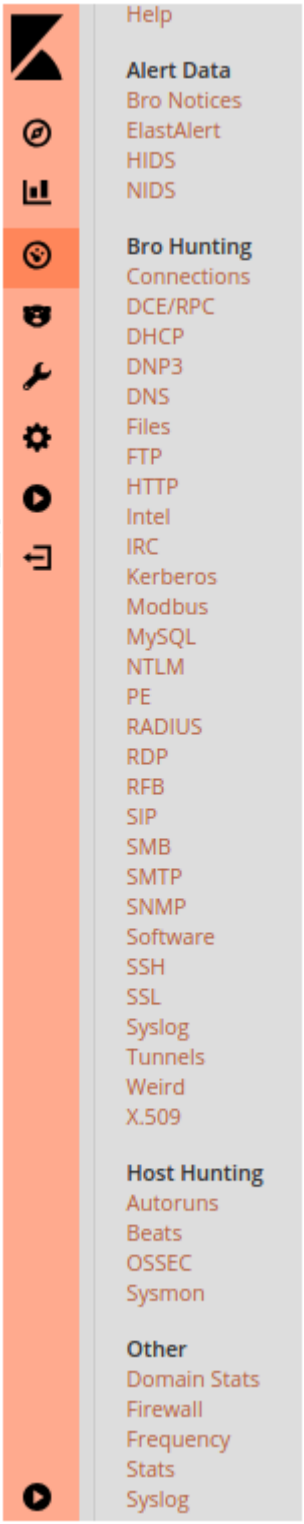


Figure 2 - Kibana Dashboards

This paper focuses on the host-based dashboards, as we are only analyzing Windows Event Logs. The crucial dashboard, in this case, is “HIDS” under the “Alert Data” category. This dashboard, shown in Figure 3, analyzes the host-based logs (Windows Event Logs in this case) against a set of rules. The default installation includes 114 rules, which are included in the /var/ossec/rules/ directory. This directory contains a file named “local_rules.xml” where an organization can create custom alert criteria.

If Security Onion does not generate alerts with the default rule configurations, the event logs will be reviewed at a high level to determine if the data is being captured and if it would be available for investigation after malicious activity was detected. In cases where the simulated attacks do not trigger an alert, but there is sufficient data contained in the event logs being captured, suggestions will be made to identify filters that can be used to create a tile on the dashboard or a recommendation to create a custom rule to add to the “local_rules.xml” file.

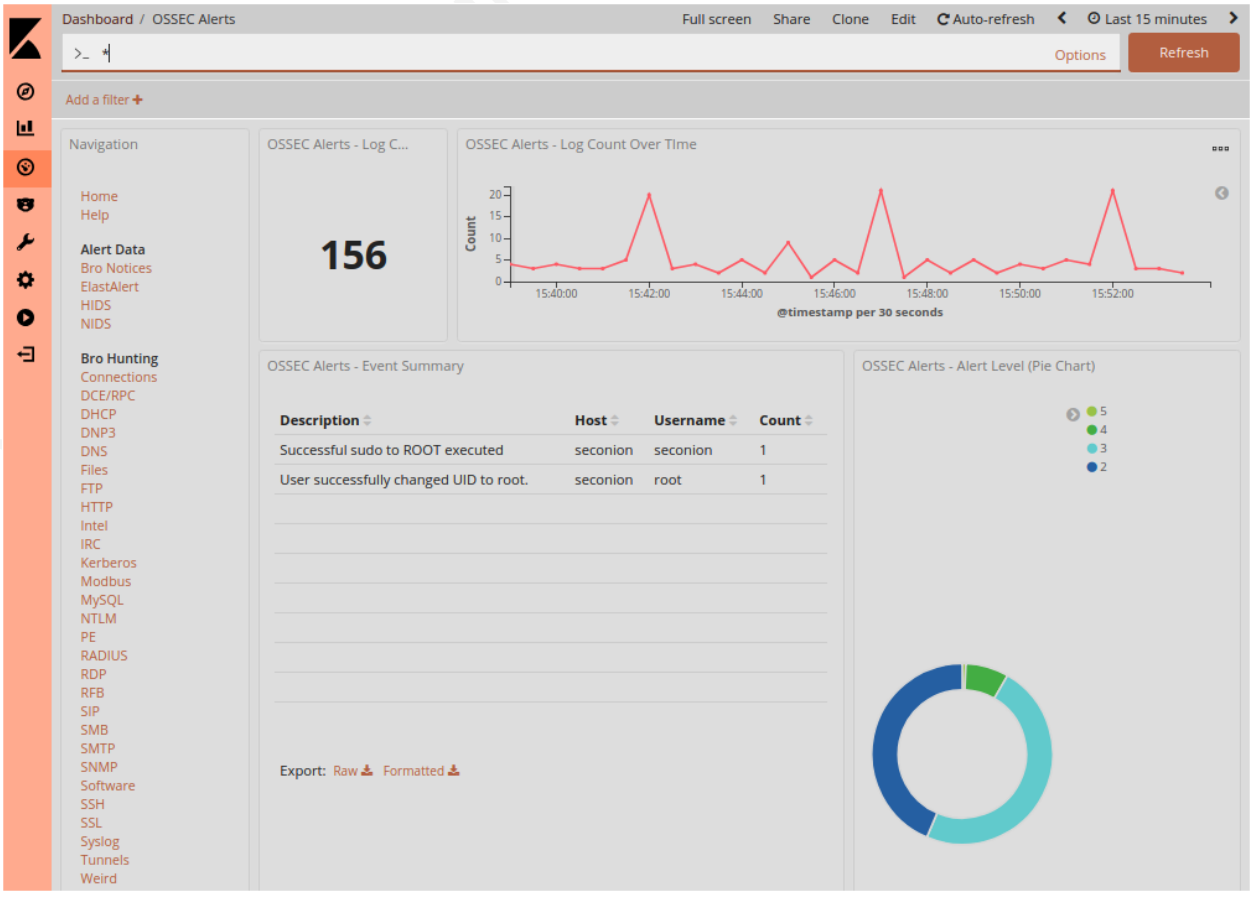


Figure 3 - OSSEC Alert Dashboard

The “OSSEC” dashboard, under the “Host Hunting” category is the second dashboard to investigate. OSSEC Alerts provides access to all log files imported to Security Onion via Wazuh agents and is used for further investigation. Before performing any targeted searching, the dashboard presents a summary of recent OSSEC logs, shown in Figure 4.

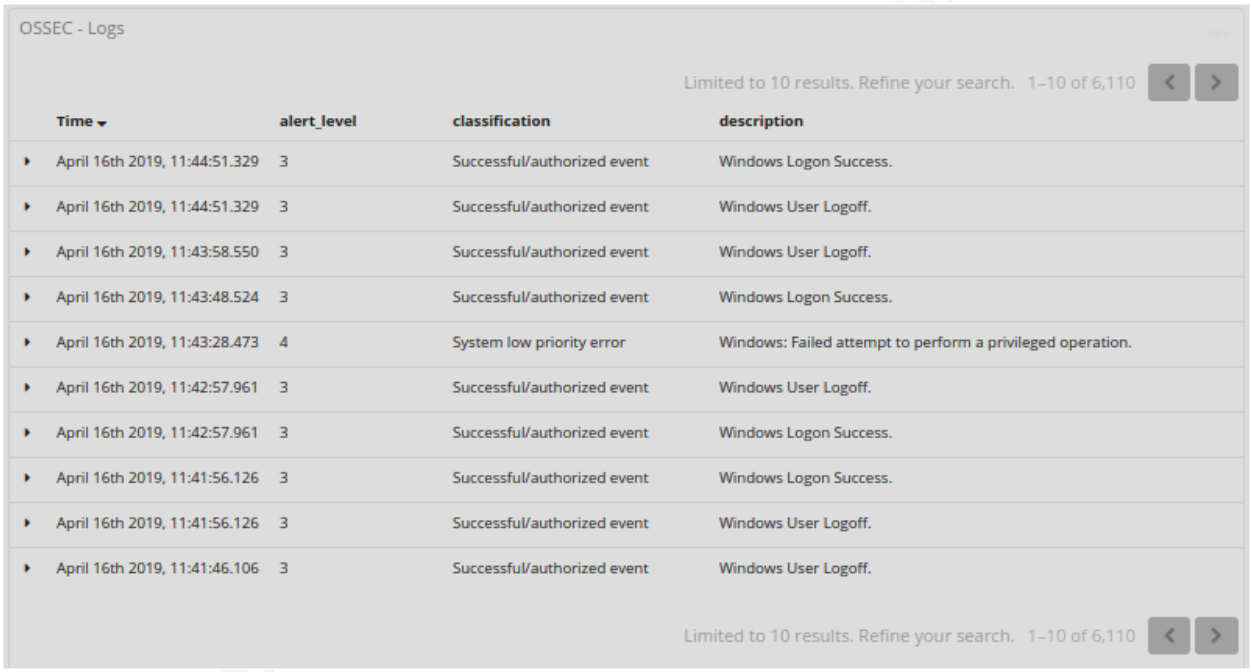


Figure 4 - OSSEC Alert Dashboard Detail

Security Onion includes many other useful tools that are not within the scope of this paper. However, it is worth noting the benefits of these supplementary tools, especially the network traffic alerting capabilities provided by Zeek (formerly Bro) (The Zeek Network Security Monitor, n.d.). Mature environments will benefit from the ingestion of network traffic as a means to detect malicious activity or indicators of future malicious activity. Examples of this type of information include Zeek alerting based on predefined rules, custom rules, and the ability to hunt for threats based on traffic characteristics such as the service. Zeek is focused on identifying abnormal activity by chaining different events and network packets together.

The second component worth noting is Suricata as a Network Intrusion Detection System (NIDS). Suricata functions like a traditional NIDS by looking at the flow of network traffic but adds the ability to extract files and certificates for analysis. Suricata also includes an engine that will process these artifacts against a set of rules (Suricata, n.d.), similar to the host-based alerting

this paper investigates. Organizations can alter these alerting capabilities with custom rules written in the Lua language.

3. Malicious Actor Activity

This paper used CALDERA to simulate three different attack scenarios to evaluate event log collection and review processes. These simulated attacks include using TTPS such as PsExec, pass-the-hash, and file transfers. These attacks cover the expected actions of a malicious actor, such as enumerating the environment, moving laterally to reach high-value systems, and collecting files for exfiltration.

3.1 Assumptions

Several assumptions were made about the environment to allow for the lab network to represent a standard SMB network. Some of the non-administrator user accounts were granted local administrator access to their workstations. Additionally, some of the users with Domain Admin rights do not maintain a separate low-privilege user account for standard daily activity. Lastly, there is no significant network segmentation in place.

The first assumption made was that some users would have local administrator rights on their workstations. The reason behind this is again related to resource constraints. If users can install software, the IT staff can focus efforts on more critical projects. While this provides a level of convenience for the end user, there are security risks introduced by this elevated access, particularly if that user's session is compromised. As evidence of this scenario, one workstation was configured to have a domain user account that had local admin rights logged in during the simulated attacks.

The second assumption made was that at least one user with Domain Admin rights was using a single high privileged account as their only user account. This situation presents additional concerns around user session and credential compromise due to the high value of a Domain Admin user account. As evidence of this scenario, one workstation was configured to have a Domain Admin user account that logged in during the simulated attacks.

The third assumption made was a lack of network segmentation. There were no network controls put in place to restrict traffic between workstations and servers. The network was one

"flat" segment. This network design represents a simple network design, likely to match that of many SMBs.

These assumptions are worth noting, as each assumption may be reversed and used as a recommendation for securing the environment. The latest *Microsoft Vulnerabilities Report* from Beyond Trust shows this is an ongoing issue, noting that of the 189 critical vulnerabilities announced, 154 of the vulnerabilities required local administrator rights (Microsoft Vulnerabilities Report 2019: An Analysis of Microsoft Security Updates in 2018). Additionally, organizations commonly struggle with implementing adequate traffic restrictions between network segments largely due to the increased complexity of managing these configurations (Jaworski 2017). Outside of the detection capabilities tested in this research, organizations would be well served to restrict local administrator rights on endpoints, operate with the principle of least privilege with user accounts, and allow only necessary network activity.

3.2 Lateral Movement - PsExec

The first TTP this paper examines is using PsExec to install and initiate a Remote Access Trojan (RAT) on a remote host, allowing the attackers to move laterally through the environment. PsExec is a legitimate system administration utility released as part of Windows Systinternals, which can make detection of malicious use difficult (PsExec - Windows Sysinternals, n.d.). Abuse of these tools, known as "adminware" remains a prevalent attack vector (*2018 Data Breach Investigations Report*). This TTP in CALERA performs the following actions, as shown in Figure 5 below:

- Enumerates the domain to which the initial system is connected.
- Enumerates the privileged accounts of the domain to which the initial system is connected.
- Checks the system for credentials stored in memory.
- Executes PsExec using privileged credentials to start a RAT on another domain-joined machine.

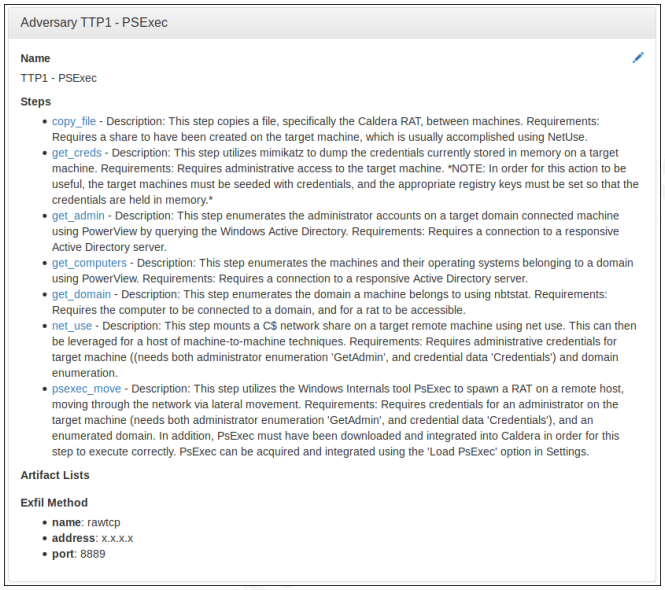


Figure 5 - TTP1 PsExec

3.3 Lateral Movement - Pass the Hash

The second TTP this paper examined was lateral movement via pass-the-hash techniques. This includes extracting privileged credentials from memory of the initial host to use the password hash, without having to brute-force the plaintext password (CWE-836: Use of Password Hash Instead of Password for Authentication, n.d.). These password hashes will be used to open a remote network share and transfer a file, which is not the standard workflow an administrative user would follow. This TTP in CALERA performs the following actions, as shown in Figure 6 below:

- Enumerates the domain to which the initial system is connected.
- Enumerates the privileged accounts of the domain to which the initial system is connected.
- Checks the system for credentials stored in memory.
- Uses the hashed password retrieved from memory to transfer a file to a remote host.
- Uses the hashed password retrieved from memory to start a Windows service to transfer a file to a remote host.

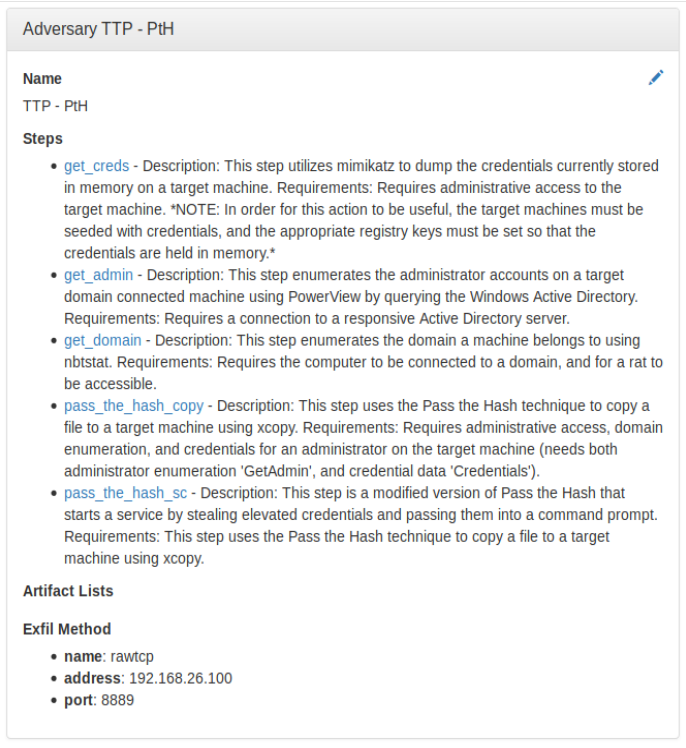


Figure 6 - TTP2 Pass the Hash

3.4 Collecting and Centralizing Files - xcopy

The third TTP this paper examined was lateral file transfers, as an early indication of files being collected in a central location before exfiltration. The process of copying files from workstation to workstation is not a standard work-flow and should stand out as questionable activity. This TTP in CALERA performs the following actions, as shown in Figure 7 below:

- Enumerates the domain to which the initial system is connected.
- Enumerates the privileged accounts of the domain to which the initial system is connected.
- Mounts a remote network share from a second machine.
- Transfers a CALDERA RAT.
- Transfers a local file to the mounted network share.

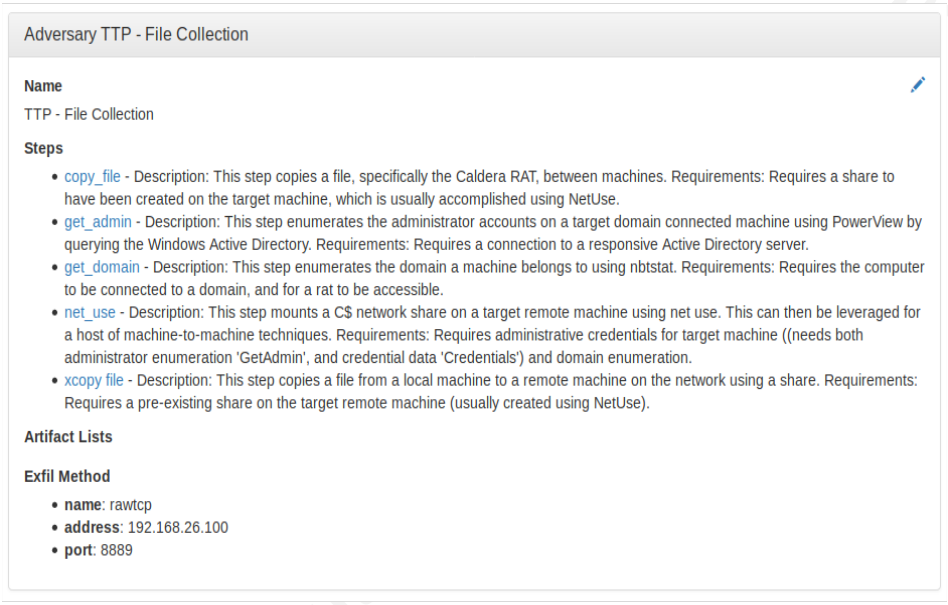


Figure 7 - TTP3 File Collection

Each of these attack techniques match actions documented in the MITRE Attack Framework as techniques used by the Lazarus Group, documented as Group ID: G0032 (MITRE Attack).

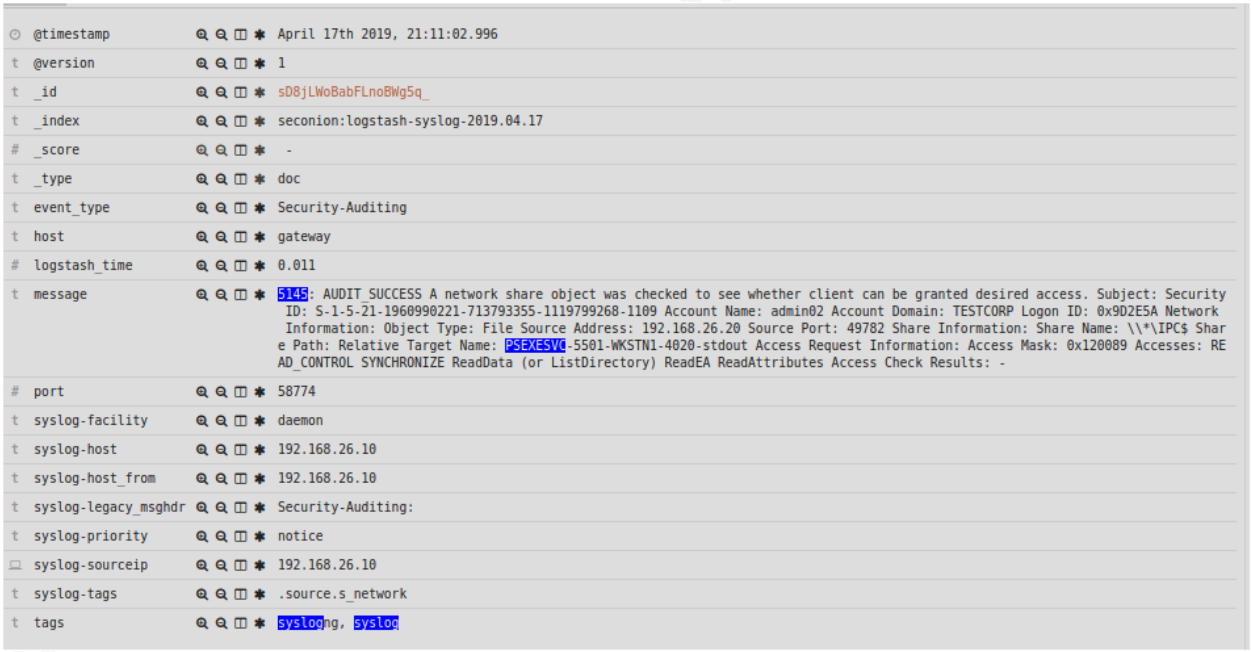
4. Detecting Threats

A series of each attack scenario was run to allow for sufficient log volume generation to make hunting for threats in Security Onion more realistic, and avoid hunting for only event logs generated during the few minutes of a simulated attack. SMBs with limited resources may not have the ability to implement a threat hunting program to parse through large volumes of event logs. These organizations need a solution that will make detection and recognition of potential threats stand out among the noise.

4.1 Detecting Lateral Movement – PsExec

The execution steps included in TTP1 – PsExec provide several artifacts we can look for when hunting for malicious use of PsExec as a means of lateral movement through our environment. Specifically, the PsExec tool creates a distinct log entry in the file share access request.

Due to the Audit Policy, events that fall under Object Access – Detailed File Share (Success) (Event ID 5145) will be recorded. This will log any successful mapping of file shares. This can capture plenty of legitimate uses, so to filter this down, we want to look for instances of the inclusion of "PSEXECSVC" in the log data. This traffic did not generate any alerts in our monitoring dashboards with the default rules in place. To ensure the traffic was processed by Security Onion, a search was performed in the Log Stash database to highlight any records that contained "5145" and "PSEXECSVC" in the message content as shown in Figure 8 below.



@timestamp		April 17th 2019, 21:11:02.996
@version		1
_id		sD8jLWoBabFLnoBWg5q_
_index		seconion:logstash-syslog-2019.04.17
_score		-
_type		doc
event_type		Security-Auditing
host		gateway
logstash_time		0.011
message		5145: AUDIT_SUCCESS A network share object was checked to see whether client can be granted desired access. Subject: Security ID: S-1-5-21-1960990221-713793355-1119799268-1109 Account Name: admin02 Account Domain: TESTCORP Logon ID: 0x9D2E5A Network Information: Object Type: File Source Address: 192.168.26.20 Source Port: 49782 Share Information: Share Name: *\IPC$ Share Path: Relative Target Name: PSEXESVC-5501-WKSTN1-4020-stdout Access Request Information: Access Mask: 0x120089 Accesses: READ_CONTROL SYNCHRONIZE ReadData (or ListDirectory) ReadEA ReadAttributes Access Check Results: -
port		58774
syslog-facility		daemon
syslog-host		192.168.26.10
syslog-host_from		192.168.26.10
syslog-legacy_msghdr		Security-Auditing:
syslog-priority		notice
syslog-sourceip		192.168.26.10
syslog-tags		.source.s_network
tags		syslogng, syslog

Figure 8 - PsExec Event Log Detail

The details of this event log show that this is classified as priority "notice" by syslog, as this event is a legitimate administration tool being used by a legitimate user. This log represents standard business activity. Organizations may write custom rules to raise the priority level within the syslog dashboards, or to trigger an alert in the HIDS dashboard. This customization of rules goes against the hypothesis of an easy solution for detecting malicious activity but does provide a means of entry-level threat hunting.

The second artifact in this scenario that should trigger an alert is the use of Mimikatz through the "get_creds" step. Mimikatz is a well-known post-exploitation tool that extracts credentials from memory (Porup, 2019). When this scenario was run under privileged credentials, no alerts were generated. This shortfall is due to the lack of command line process

auditing (Command line process auditing, 2017). Some organizations may not want to enable this setting due to the risk of capturing credentials entered as script arguments in event logs.

When this TTP executes under the context of a standard user with no administrative rights on the local machine, multiple OSSEC alerts trigger due to privilege escalation failures (Figure 9). OSSEC alert level 10 events are defined as: “Multiple user generated errors - They include multiple bad passwords, multiple failed logins, etc. They may indicate an attack or may just be that a user just forgot his [credentials]” (Rules Classification¶, n.d.).

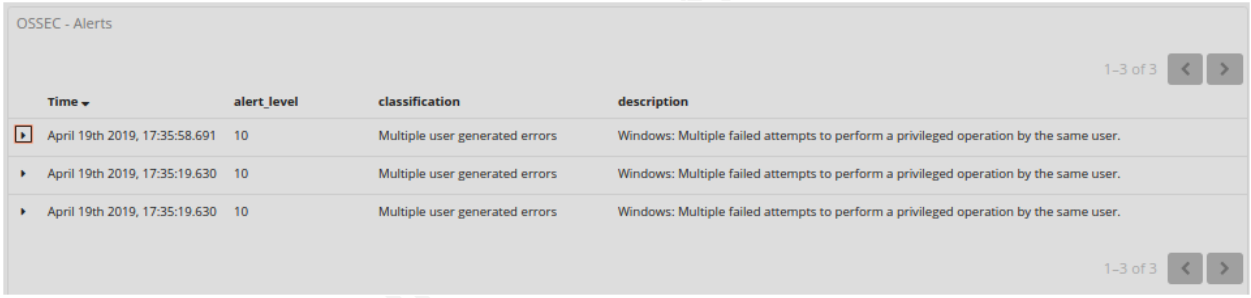


Figure 9 - OSSEC Alert Admin Failure

These alerts can be expanded to view further details, including the content of the full event log, as shown in Figure 10.

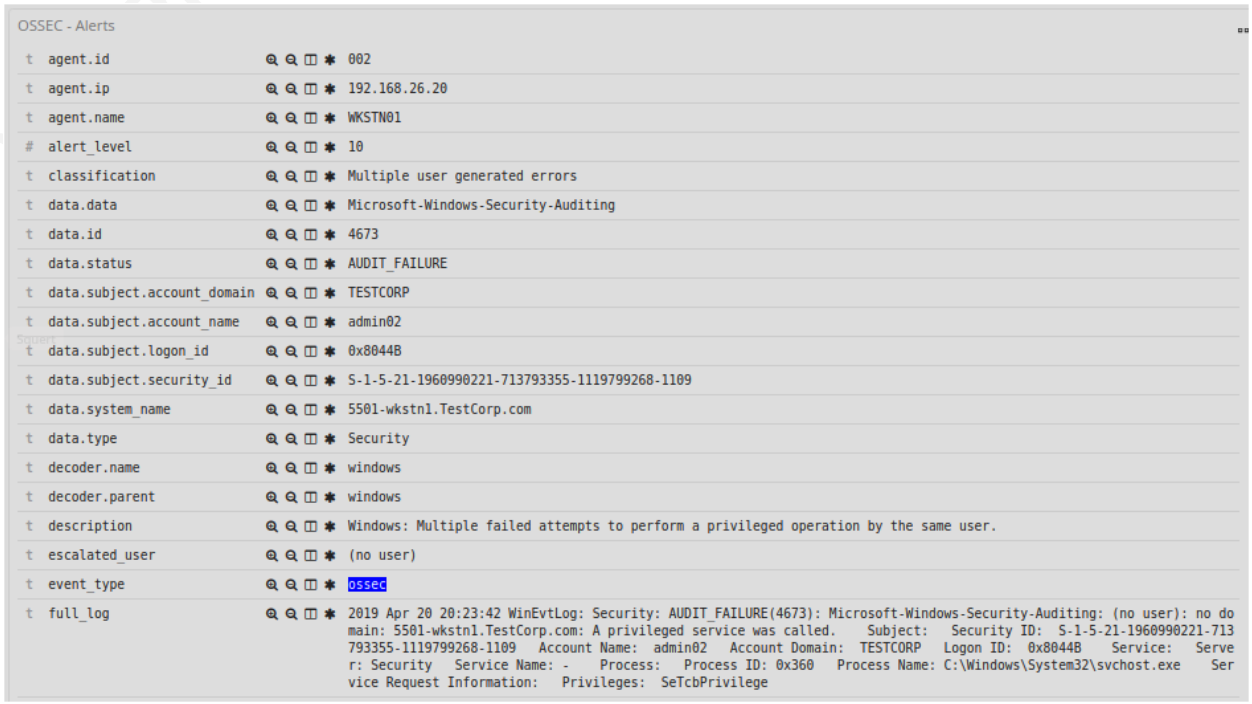


Figure 10 - OSSEC Alert Admin Failure Detail

Sean D. Goodwin, SeanGoodwin@protonmail.ch

4.2 Detecting Lateral Movement - Pass the Hash

The execution steps included in TTP2 – Pass the Hash present similar detection struggles to that of TTP1 – PsExec. Specifically, the use of Mimikatz to extract credentials (password hashes in this case) was not detected by the default configuration and alert rules. Script block logging, as discussed above, may provide additional detection capabilities in this situation.

The second indication of malicious activity to look for is evidence of login via Pass the Hash. This attack is another example of attackers taking advantage of legitimate processes. Figure 11 below shows the Sysmon alert dashboard indicating there are no alerts. The two filters shown are limiting alerts to those coming from the machine to which the hash was passed, and which contain the source port the attack was sent from.

Figure 11 - Sysmon PtH Alert Dashboard

Although there were no alerts triggered for this attack, the Log Stash database contains the necessary information to search for suspected Pass the Hash attacks. David Kennedy provides a set of data fields and values that can be used to filter a large number of authentication logs down to a manageable set for investigation (Kennedy, 2016). Specifically, Kennedy recommends filtering to display:

- Windows Event ID 4624
- Logon Type = 3
- Logon Process = NtLmSsP
- Key Length = 0

Kennedy's blog post goes on to state the Security ID should be null, which is reflected in Figure 12 below as "Security ID: S-1-0-0". While this filtering identifies the attack traffic, this

will also present an analyst with potential false positives (valid authentication logs). To reduce the number of these false positive records, and analyst would need to identify any legitimate uses of Pass the Hash activity in the environment and create additional filters. The filters identified by Kennedy are shown as highlights in Figure 12, indicating this event is a potential Pass the Hash attack, and is worth further review.

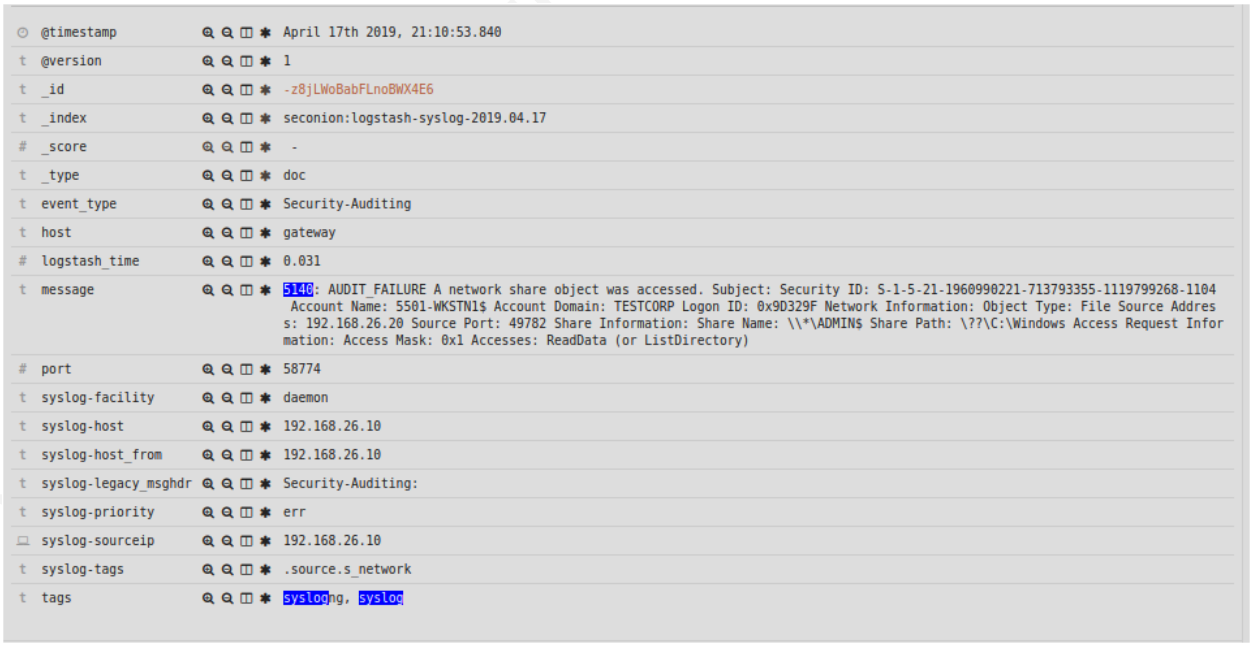
Figure 12 - Pass the Hash Event Log

Detecting the use of Pass the Hash attacks by malicious actors will require significant tuning to each environment. For environments where this tuning is not feasible, the mitigating recommendations outlined in *Pass-the-hash attacks: Tools and Mitigation* (Ewaida, 2010) should be implemented to reduce the likelihood of a successful Pass the Hash attack.

4.3 Detecting Collecting and Centralizing Files - xcopy

The execution steps included in TTP3 – Collecting and Centralizing Files - xcopy presents another detection challenge, as users accessing files over the network is a typical workflow. The default configuration and alert rules did not alert on the attack traffic that accessed these file shares and copied files to a central location.

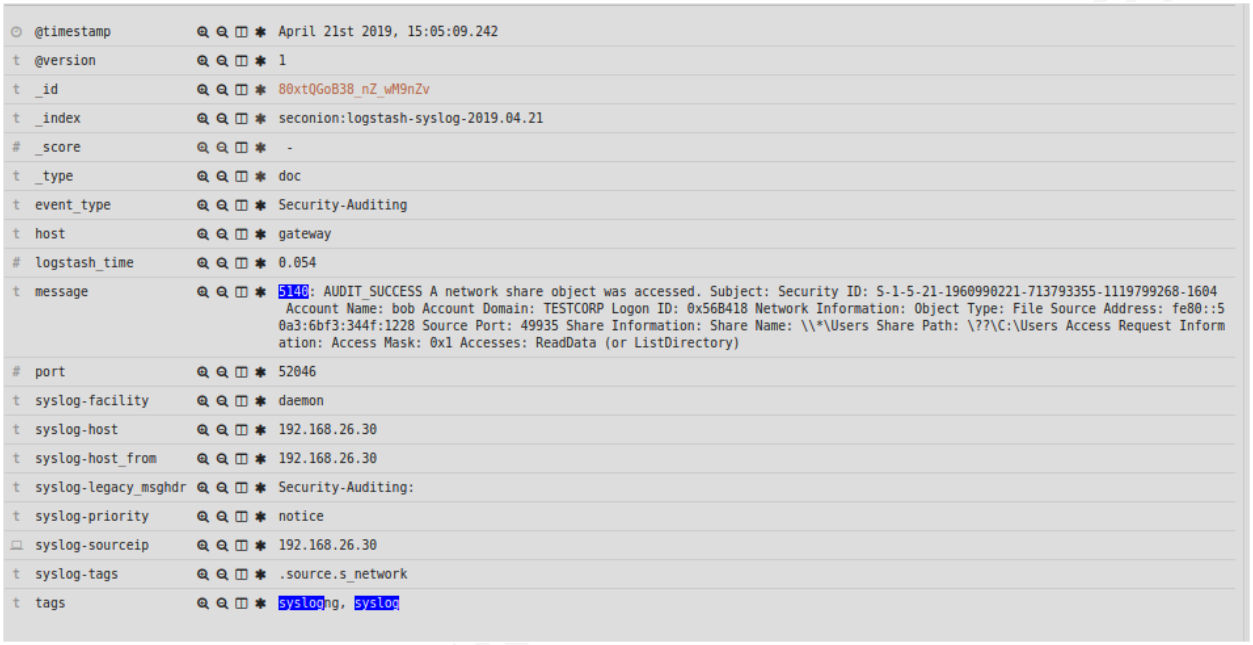
The Log Stash database again provides an analyst with the opportunity to hunt for suspicious file share access. Windows Event ID 5140 shows the first time a network share is accessed – successfully or unsuccessfully, both of which are captured in our Audit Policy. The analyst could then apply a filter to show the alert coming only from workstation hosts, to exclude file shares where this Event ID is expected as typical activity (e.g. users accessing a departmental file share). In this lab environment, the file share is hosted on 192.168.26.10, so the event log shown in Figure 13 may represent a typical share request, depending on the files and folders being requested. This access requested was an audit failure, meaning the user was not authorized to access the requested file or share. Upon further investigation, the share being requested (*\ADMIN$) provides us with a second potential indication of malicious activity.



@timestamp		April 17th 2019, 21:10:53.840
@version		1
_id		-z8jLWoBabFLnoBWX4E6
_index		seconion:logstash-syslog-2019.04.17
_score		-
_type		doc
event_type		Security-Auditing
host		gateway
logstash_time		0.031
message		5140: AUDIT_FAILURE A network share object was accessed. Subject: Security ID: S-1-5-21-1960990221-713793355-1119799268-1104 Account Name: 5501-WKSTN1$ Account Domain: TESTCORP Logon ID: 0x9D329F Network Information: Object Type: File Source Address: 192.168.26.20 Source Port: 49782 Share Information: Share Name: *\ADMIN$ Share Path: \??\C:\Windows Access Request Information: Access Mask: 0x1 Accesses: ReadData (or ListDirectory)
port		58774
syslog-facility		daemon
syslog-host		192.168.26.10
syslog-host_from		192.168.26.10
syslog-legacy_msghdr		Security-Auditing:
syslog-priority		err
syslog-sourceip		192.168.26.10
syslog-tags		.source.s_network
tags		syslogng, syslog

Figure 13 - File Share Access Failure

Figure 14 shows an example event log that raises more suspicion as it shows a workstation browsing the "Users" directory of another workstation. A typical process flow for collaborating on files in a corporate environment is workstations accessing files on a dedicated file server, not a workstation accessing a file stored on another workstation. In this typical configuration, files being moved directly between workstations may present an interesting event worthy of further scrutiny.



@timestamp		April 21st 2019, 15:05:09.242
@version		1
_id		80xtQGoB38_nZ_wM9nZv
_index		seconion:logstash-syslog-2019.04.21
_score		-
_type		doc
event_type		Security-Auditing
host		gateway
logstash_time		0.054
message		5140: AUDIT_SUCCESS A network share object was accessed. Subject: Security ID: S-1-5-21-1960990221-713793355-1119799268-1604 Account Name: bob Account Domain: TESTCORP Logon ID: 0x56B418 Network Information: Object Type: File Source Address: fe80::50a3:6bf3:344f:1228 Source Port: 49935 Share Information: Share Name: *\Users Share Path: \??\C:\Users Access Request Information: Access Mask: 0x1 Accesses: ReadData (or ListDirectory)
port		52046
syslog-facility		daemon
syslog-host		192.168.26.30
syslog-host_from		192.168.26.30
syslog-legacy_msghdr		Security-Auditing:
syslog-priority		notice
syslog-sourceip		192.168.26.30
syslog-tags		.source.s_network
tags		syslogng, syslog

Figure 14 - File Share Access Request

Another indication to search for is Windows Event ID 5142. This event log is created when a new network share object is created. If an attacker is more concerned with easy access from other hosts, a new network share with access rights granted to the "Everyone" group provides a simple repository for centralizing files before exfiltration offsite. Figure 15 provides an example of the "C:\Users\admin02\Documents" folder shared with the network.

This level of filtering is time-consuming, as filters must be written to only show suspicious connections, which are likely to be buried in the white noise of valid file share access on the SMB network.

@timestamp	April 21st 2019, 15:22:07.319
@version	1
_id	g1R9QGoB38_nZ_wMf3Hn
_index	seconion:logstash-syslog-2019.04.21
_score	-
_type	doc
event_type	Security-Auditing
host	gateway
logstash_time	0.024
message	5142: AUDIT_SUCCESS A network share object was added. Subject: Security ID: S-1-5-21-1960990221-713793355-1119799268-1109 Account Name: admin02 Account Domain: TESTCORP Logon ID: 0x260B9F Share Information: Share Name: *\Documents Share Path: C:\Users\admin02\Documents
port	52046
syslog-facility	daemon
syslog-host	192.168.26.20
syslog-host_from	192.168.26.20
syslog-legacy_msghdr	Security-Auditing:
syslog-priority	notice
syslog-sourceip	192.168.26.20
syslog-tags	.source.s_network
tags	syslogng, syslog

Figure 15 - Network Share Creation

5. Potential Enhancements

Several potential enhancements noted in the investigations above require further explanation. These include enabling command line process auditing to capture the full text of any command line arguments entered, developing a set of custom alert rules, and following recommended configuration changes to mitigate the likelihood of these attack techniques.

As noted in the investigations above, the default rule configurations are not designed to trigger alerts for every attack scenario. In order to make this toolset more efficient, analysts must create rules tailored to their environment designed to highlight activities the analyst deems worthy of immediate investigation.

An example of a custom rule is shown below in Figure 16. This rule has been written to trigger an alert for events matching the Pass the Hash indicators that were identified by various search filters in the event log database. If an organization would prefer active alerts, with the potential for false positives, over the legwork required to search for these events, this rule may prove useful. This chain of rules performs the same filtering and analysis discussed in Kennedy's post that were used in manual filtering to identify these attacks. A level 7 alert is generated when these criteria are met.

```
<!-- Built-in rule 1807 is used as our initial trigger. Below each of the search criteria discussed in the paper
are presented in a chain. If all search criteria are met, a level 7 alert is raised for further investigation.

<rule id="18107" level="3">
        <if_sid>18104</if_sid>
         <id>^528$|^540$|^673$|^4624$|^4769$</id>
        <description>Windows Logon Success.</description>
        <group>authentication_success,pci_dss_10.2.5,gpg13_7.1,gpg13_7.2,gdpr_IV_32.2,</group>
</rule>
-->

<!-- This rule searches any entries that match rule 18107 for the string 'S-1-0-0' (Security ID) -->
<rule id="100002" level="1">
        <if_sid>18107</if_sid>
        <match>S-1-0-0</match>
        <description>S-1-0-0 successful Auth</description>
</rule>

<!-- This rule searches any entries that match rule 100002 for the string 'Logon Type:  3' -->
<rule id="100003" level="1">
        <if_sid>100002</if_sid>
        <match>Logon Type:  3</match>
        <description>Logon Type:  3</description>
</rule>

<!-- This rule searches any entries that match rule 100003 for the string 'Logon Process: NtLmSsp' -->
<rule id="100004" level="1">
        <if_sid>100003</if_sid>
        <match>Logon Process: NtLmSsp</match>
        <description>Logon Process: NtLmSsp</description>
</rule>

<!-- This rule searches any entries that match rule 100002 for the string 'Key Length:  0'. If this rule is
triggered, all search criteria have been met, and this event warrants further investigation -->
<rule id="100005" level="7">
        <if_sid>100004</if_sid>
        <match>Key Length:  0</match>
        <description>Potential Pass the Hash!</description>
</rule>
```

Figure 16 - Custom PtH Rule PoC

6. Conclusion

The technologies examined in this paper contain much of the information needed to investigate for potential threats, but do not provide for a plug-and-play alerting mechanism with the default configurations. This is due in part to the custom activity representing common traffic in each environment but is also due to the selected attacks leveraging legitimate Windows utilities. SMBs may benefit from disabling some of these utilities outright if they are not being used for system administration. Other risk mitigation steps, such as performing daily tasks with a non-administrative user account can reduce the likelihood of attack success and increase the visibility of attempted attacks.

Analysts or custom rules are needed to determine if the usage of tools such as PsExec or “net use” should be considered legitimate or potentially malicious. The data provided through Windows Event Logs, Sysmon rules, and Security Onion parsing of the data aids in efficient parsing of large event log volumes. SMBs can rely on this toolset to identify some common attack techniques, but should not see this as a “set it and forget it” solution to trigger alerts for all attackers.

Sean D. Goodwin, SeanGoodwin@protonmail.ch

References

2018 Data Breach Investigations Report (Rep.). (n.d.). Verizon.

Cole, E., PhD. (n.d.). Defending Against the Wrong Enemy: 2017 SANS Insider Threat Survey (Tech.). SANS Institute. doi:https://www.sans.org/reading-room/whitepapers/analyst/defending-wrong-enemy-2017-insider-threat-survey-37890

Command line process auditing. (2017, May 30). Retrieved from https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/manage/component-updates/command-line-process-auditing

CWE-836: Use of Password Hash Instead of Password for Authentication. (n.d.). Retrieved from http://cwe.mitre.org/data/definitions/836.html

Ewaida, S. (2010, January 21). Pass-the-hash attacks: Tools and Mitigation [Scholarly project]. In SANS Reading Room. Retrieved from https://www.sans.org/reading-room/whitepapers/testing/pass-the-hash-attacks-tools-mitigation-33283

Hosburgh, M. (n.d.). Offensive Intrusion Analysis: Uncovering Insiders with Threat Hunting and Active Defense. Retrieved February 20, 2019, from https://www.sans.org/reading-room/whitepapers/threathunting/offensive-intrusion-analysis-uncovering-insiders-threat-hunting-active-defense-37885

Jaworski, S. (2017, June). Does Network Micro-segmentation Provide Additional Security? (Tech.). Retrieved https://www.sans.org/reading-room/whitepapers/networksecurity/network-micro-segmentation-provide-additional-security-38030

Kennedy, D. (2016, October 12). Reliably Detecting Pass the Hash Through Event Log Analysis. Retrieved from https://blog.binarydefense.com/reliably-detecting-pass-the-hash-through-event-log-analysis

Logstash. (n.d.). Retrieved from https://www.elastic.co/products/logstash

Managing Agents¶. (n.d.). Retrieved from https://ossec-docs.readthedocs.io/en/latest/manual/agent/agent-management.html#managing-agents

Microsoft Vulnerabilities Report 2019: An Analysis of Microsoft Security Updates in 2018 (Rep.). (n.d.). Retrieved https://beyondtrust-bomgar12.netdna-ssl.com/assets/documents/Microsoft-Vulnerabilities-Report-2019.pdf

Sean D. Goodwin, SeanGoodwin@protonmail.ch

MITRE ATT&CK™. (n.d.). Retrieved from https://attack.mitre.org/

Mitre. (n.d.). Mitre/caldera. Retrieved from https://github.com/mitre/caldera

NXLog. (n.d.). Retrieved from https://nxlog.co/

Operating System Market Share. (n.d.). Retrieved from https://netmarketshare.com/operating-system-market-share

PsExec - Windows Sysinternals. (n.d.). Retrieved from https://docs.microsoft.com/en-us/sysinternals/downloads/psexec

Porup, J. (2019, March 05). What is Mimikatz? And how this password-stealing tool works. Retrieved from https://www.csoonline.com/article/3353416/what-is-mimikatz-and-how-to-defend-against-this-password-stealing-tool.html

Rules Classification¶. (n.d.). Retrieved from https://ossec-docs.readthedocs.io/en/latest/manual/rules-decoders/rule-levels.html

Security Onion. (n.d.). Retrieved from https://securityonion.net/

Suricata. (n.d.). Retrieved from https://suricata-ids.org/

Sysmon¶. (n.d.). Retrieved from https://securityonion.readthedocs.io/en/latest/sysmon.html

Sysmon-Config [Brochure]. (n.d.). Retrieved from https://github.com/SwiftOnSecurity/sysmon-config

Sysmon - Windows Sysinternals. (2019, February 18). Retrieved from https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon

The Zeek Network Security Monitor. (n.d.). Retrieved from https://www.zeek.org/

Untangle. (n.d.). 2018 SMB It Security Report. Retrieved from https://www.untangle.com/2018-smb-it-security-report/

Wazuh - The Open Source Security Platform. (n.d.). Retrieved from https://wazuh.com/

WINDOWS ATT&CK LOGGING CHEAT SHEET - Win 7 - Win 2012 [Brochure]. (n.d.). Retrieved from https://www.malwarearchaeology.com/cheat-sheets

Sean D. Goodwin, SeanGoodwin@protonmail.ch

Version 1.0

WINDOWS LOGGING CHEAT SHEET - Win 7 thru Win 2019 [Brochure]. (n.d.). Retrieved from https://www.malwarearchaeology.com/cheat-sheets

Version 2.3

Sean D. Goodwin, SeanGoodwin@protonmail.ch