Intrusion Detection Using Indicators of Compromise Based on Best Practices and Windows Event Logs

María del Carmen Prudente Tixteco, Lidia Prudente Tixteco, Gabriel Sánchez Pérez, Linda Karina Toscano Medina
Instituto Politécnico Nacional
Sección de Estudios de Posgrado e Investigación ESIME Culhuacan
Santa Ana 1000, San Francisco Culhuacán, Coyoacán, D. F., México
mprudentet0900@alumno.ipn.mx, lprudente@ipn.mx, gasanchezp@ipn.mx, ltoscano@ipn.mx

***Abstract*— Nowadays computer attacks and intrusions have become more common affecting confidentiality, integrity or the availability of computer systems. They are more sophisticated making the job of the information security analysts more complicated, mainly because of the attacking vectors are more robust and complex to identify. One of the main resources that information security people have on their disposition are *Indicators of Compromise* (*IOCs*), which allow the identification of potentially malicious activity on a system or network. Usually IOCs are made off virus signatures, IP addresses, URLs or domains and some others elements, which are not sufficient to detect an intrusion or malicious activity on a computer system. The *Windows event logs* register different activities in a Windows® operating system that are valuable elements in a forensic analysis process. *IOCs* can be generated using *Windows event logs* for intrusion detection, improving *Incident Response* (*IR*) and forensic analysis processes. This paper presents a procedure to generate *IOCs* using *Windows event logs* to achieve a more efficient diagnostic computer system for *IR*.**

Keywords-indicators of compromise; windows event logs; intrusion detection.

I. INTRODUCTION

In the process of *IR* and forensic analysis, determining that a computer system is compromised could mean success or failure to mitigate a security incident. Attack vectors have increased their complexity, making more difficult for information security analysts to identify their presence. Different tools have been developed to facilitate their identification; one of these tools are the *IOCs*.

IOCs are pieces of forensic data, that could be found in log entries or system files, which can help to identify potentially malicious activity on a system or network [1].

In order to reduce the number of false-positive results over time, it is required to improve *IR* and forensic analysis procedures that collaborate directly with the *Computer Incident Response Team* (*CIRT*).

As a part of a forensic analysis recommendation of *Computer Emergency Response Teams* (*CERTs*), is of vital importance that a computer system intrusion is promptly identified, in order to perform the actions that will avoid further damages within the information system infrastructure [6].

Security Windows Event Logs Codes can be mainly used to describe malicious activity [4], and these can be added to antivirus signatures, IP addresses, URLs or domains to make *IOCs* more robust and specific to determine if a computer system could be compromised.

The *containment step* for the *IR* process is considered the most important; because it allows us to know if changes were made in the system [2], e.g., changes in privileges within registry keys, relocation of .dll system files or any other manipulation of processes and/or files during the intrusion on a computer system.

Knowing the behavior of the attacker is of great importance, it helps to achieve a better analysis stage for *IR* and forensic analysis processes, applying proactive actions and preventing future intrusions, e.g., computer system hardening.

Event logs should not be considered as isolated events; they have to be considered as a whole, that happen over a period of time and occur commonly, i.e., to obtain an *IOC* a failed login is not enough; this event must be associated with some others to determine a compromised computer system event.

This article presents a procedure to use *Windows event logs* to generate *IOCs*, achieving the detection of a computer system intrusion and help the information security people or a *CIRT* to take quickly and effective decisions to defend the information system infrastructure.

This paper is organized as follows. Section II presents state of the art. Section III presents an explanation of Windows event logs. Section IV describes the functionality of indicators of compromise. Section V describes the development of this research. Section VI presents the results of this research and Section VII conclusion and future work.

II. STATE OF THE ART

An incident is a compromise or a security violation in an organization. Preparation of the *CIRT* through planning, communication, and practice of the *IR* process will provide the experience needed to perform actions timely should an incident occur [2].

There are six basic steps when an *IR* occurs: preparation, identification, containment, eradication, recovery and lessons

 ISBN: 978-1-61208-475-6

learned; they provide the basic foundation to create and perform their own *IR* plan. Specifically, there is one task within the containment phase that should be done: the system back-up. This provides the information about how the events happened, resulting in an incident from a malicious activity, or to be used for observing how the system was compromised during the phase of lessons learned [2].

In order for an *IR* to be considered successful, the *CIRT* should know the steps followed by an attacker when computer system is being compromised. There are seven steps: reconnaissance, weaponize, deliver, exploitation, installation, command and control (C2), actions on objectives, that the adversaries usually follow when attempting an intrusion, these leave a trail behind them [3]. This process is also known as the *Kill Chain Life Cycle* shown in Figure 1.

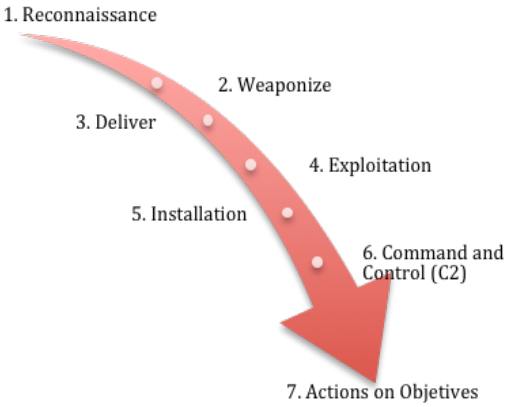

Figure 1. Kill Chain Life Cycle.

Microsoft Windows® Active Directory's best practices consider different signs to identify and evaluate a compromised computer system by monitoring and the use of alerts, through a proper configuration of Windows auditing settings. These signs can help to detect a malicious activity in a computer system early and timely.

The following security events can be considered as part of the event monitoring to detect possible signs of computer system intrusion within Windows® operating system [4]:

- Account Logon Events
- Account Management
- Directory Service Access
- Logon Events
- Object Access
- Policy Change
- Privilege Use
- Process Tracking
- System Events Audit

There are certain recommendations made by *CERTs* to help identify a compromised computer system. However, these have to be done by expert analysts [6]. When looking for signs of intrusion, most of the time if one host has been compromised, others on the same network have also been compromised. These are some of the signs to look for in a possible compromised computer system review [6]:

- Examine log files.
- Check for odd user accounts and groups.
- Check all groups for unexpected user membership.
- Look for unauthorized user rights.
- Check for unauthorized applications that start up automatically.
- Check for unauthorized processes.
- Check for altered permissions on files or registry keys.
- Check for changes in user or computer policies.
- Audit for intrusion detection.

On the other hand Hun-Ya Lock [5] presents the benefits of using OpenIOC framework as common syntax to describe the results of malware analysis; this work describes tools and techniques used during analysis but not in the reporting of results. The document emphasizes that reporting of the results is as important as the results themselves and if the results can be reported in a consistent well-structured manner that is easily understood by man and machine, then it becomes possible to automate some of the processes in the detection, prevention and reporting of malware infections.

IOC Editor is a free editor tool for *IOCs*. The *IOCs* are XML documents that support incident responders capturing different information about threats including malicious files attributes, changes in registries and artifacts in memory. IOC Editor provides an interface to manage data within these *IOCs* [7].

IOC Editor can:

- Manipulate the logical structures that define the *IOC*.
- Apply meta-information to *IOC* including detailed descriptions or arbitrary labels.
- Convert *IOCs* into XPath filters.
- Manage lists of terms that are used within *IOC*.

SANS [8] published a checklist to review critical logs during an *IR*. It can also be used for a log review routine.

The general approach is:

1) Identify which log sources and/or automated tools can be used during the analysis.

2) Copy log records over to a single location for later review.

3) Minimize noise by removing routinely and repetitive log entries after confirming that they are benign.

4) Determine whether logs' timestamps are reliable; considering the different time zone differences.

5) Focus on recent changes, failures, errors, status changes, access and administration events, and other unusual events in the environment.

6) Go back in time to reproduce the scenario before and after the incident.

7) Correlate activities across different logs to get a more comprehensive picture.

8) Develop theories about what occurred; explore logs to confirm or disprove them.

Some potential security log sources that can be found are [8]:

- Server and workstation operating system logs.

 ISBN: 978-1-61208-475-6

- Application logs.
- Security tool logs.
- Outbound proxy logs and end-user application logs.
- Other non-log sources for security events.

Finally, this document recommends what to look for on a Windows® OS like user logon/logoff events, user account changes, password changes, startup or stopping of a service, object access denial and other kind of resources [8].

Windows event logs can be used to generate *IOCs* and help identify should a computer system is compromised, detecting possible intrusions or strange behaviors, and make timely decisions accordantly.

None of the above references use *IOCs* in conjunction with *Windows event logs*.

This paper presents a procedure to fix this situation.

The efficiency of these *IOCs* could be improved if correlation tools are implemented to analyze attack vectors promptly.

III. WINDOWS EVENT LOGS

This following section describes key concepts about the Windows event logs.

A. Log

A log is a record of events occurring within an organization's systems and networks. Logs are composed of entries and they have evolved to contain information related to many different kinds of events [9].

B. Operating Systems Logs

Usually, Operating Systems (OS) for computers log a variety of information related to security. The most common types of security-related OS data are the following:

- System Events. System events are operational actions performed by OS components. Many OS allow administrators to specify what type of events will be logged and what kind of details register, such as timestamp, status and error codes, service name and user or system account associated with each event.
- Audit Records. Audit records contain security event information such as successful and/or failed authentication attempts, file accesses, security policy changes, account changes, and use of privileges [9].

OS logs are the most beneficial to identify or investigate suspicious activity involving a particular host. After suspicious activity like attacks, frauds, and inappropriate usage, OS' logs can be consulted to obtain more information on the activity and type of situation. Commonly they contain detailed information about each activity. Other logs can contain information less detailed and are helpful only to correlate events recorded in the primary log types [9].

C. Windows Event Log

Windows event log is a record of events that happen on a computer system, generating alerts and notifications. Microsoft® [10] defines an event as "any significant occurrence in the system or in a program that requires users to be notified, or an entry added to a log".

Some Windows event logs categories are [11]:

- Application. Events in this log are classified as error, warning, or information, depending on the severity of the event. An error is a significant problem. A warning is an event that is not necessarily significant, but might indicate a possible future problem. An information event describes the successful operation of a program, driver, or service.
- Security. This log contains security-related events, which are called audit events, and are described as successful or failed, depending on the event.
- Setup. Computers that are configured will have additional logs displayed here.
- System. System events are sent to this log by Windows and Windows system services, and are classified as error, warning, or information.
- Forwarded Events. Events are forwarded to this log by other computers.

Windows operating system classifies events by type as:

- Information event. Describes the successful completion of a task.
- Warning event. Notifies the administrator of a potential problem.
- Error message. Describes a significant problem that may result in a loss of functionality.
- Success audit event. Indicates the completion of an audited security event.
- Failure audit event. Describes an audited security event that did not complete successfully, such as an end-user locking himself out by entering incorrect passwords [10].

Each event in a log entry contains the following information:

- Date. The date the event occurred.
- Time. The time the event occurred.
- User. The user name of the user who was logged on when the event occurred.
- Computer. The name of the computer.
- Event ID. A Windows identification number that specifies the event type.
- Source. The program or component that caused the event.
- Type. The type of event [10].

For the purpose of this research different event logs of each category were selected and organized in the following tables. These events are the most representative of each of the categories being registered in a computer system and can help to generate an *IOC*. Tables I to VIII show the association of current Windows *Event ID* with its *Event Summary*, retrieved from appendix of Events to Monitor by Microsoft [13]:

TABLE I. EVENTS OF ACCOUNT LOGON CATEGORY.

Event ID	Event Summary
4774	An account was mapped for logon.
4776	The domain controller attempted to validate the credentials for an account.

TABLE II. EVENTS OF ACCOUNT MANAGEMENT CATEGORY.

Event ID	Event Summary
4783	A basic application group was created.
4785	A member was added to a basic application group.
4741	A computer account was created.
4742	A computer account was changed.
4727	A security-disabled global group was created.
4728	A member was added to a security-disabled global group.
4720	A user account was created.
4722	A user account was enabled.
4724	An attempt was made to reset an account's password.
4738	A user account was changed.
4740	A user account was locked out.

TABLE III. EVENTS OF PROCESS TRACKING CATEGORY.

Event ID	Event Summary
4688	A new process has been created.
4689	A process has exited.

TABLE IV. EVENTS OF LOGON CATEGORY.

Event ID	Event Summary
4634	An account was logged off.
4647	User initiated logoff.
4624	An account was successfully logged on.
4625	An account failed to log on.
4649	A replay attack was detected. May be a harmless false positive due to misconfiguration error.
4778	A session was reconnected to a Window Station.
4801	The workstation was unlocked.
4964	Special groups have been assigned to a new logon.

TABLE V. EVENTS OF OBJECT ACCESS CATEGORY.

Event ID	Event Summary
4665	An attempt was made to create an application client context.
4668	An application was initialized.
4664	An attempt was made to create a hard link.
4985	The state of a transaction has changed.
5051	A file was virtualized.
4691	Indirect access to an object was requested.
4698	A scheduled task was created.
4700	A scheduled task was enabled.
4702	A scheduled task was updated.
4657	A registry value was modified.
4660	An object was deleted.

TABLE VI. EVENTS OF POLICY CHANGE CATEGORY.

Event ID	Event Summary
4719	System audit policy was changed.
4905	An attempt was made to unregister a security event source.
4907	Auditing settings on object were changed.
4912	Per User Audit Policy was changed.
4704	A user right was assigned.
4946	A change has been made to Windows Firewall exception list. A rule was added.
4947	A change has been made to Windows Firewall exception list. A rule was modified.
4948	A change has been made to Windows Firewall exception list. A rule was deleted.
4949	Windows Firewall settings were restored to the default values.
4950	A Windows Firewall setting has changed.
4670	Permissions on an object were changed.

TABLE VII. EVENTS OF PRIVILEGE USE CATEGORY.

Event ID	Event Summary
4672	Special privileges assigned to new logon.
4673	A privileged service was called.

TABLE VIII. EVENTS OF SYSTEM AUDIT CATEGORY.

Event ID	Event Summary
5025	The Windows Firewall Service has been stopped.
5034	The Windows Firewall Driver has been stopped.
4697	Attempt to install a service
4618	A monitored security event pattern has occurred.

The events review from Windows logs can help trace the activities, *IR* and keep computer systems secure. Configuring these logs properly can help to manage the logs efficiently to identify and diagnose the current source of system problems and predict future ones.

IV. INDICATORS OF COMPROMISE

It is necessary to know the terminology used for indicators of compromise.

- Expression. The definition of a condition which, when true, suggests that intrusion activity is present.
- Simple Expression. An expression that can be defined without using "AND" or "OR" logic operators.
- Complex Expression. An expression that combines multiple simple expressions using "AND" or "OR" logic operators.

 ISBN: 978-1-61208-475-6

- Indicator of Compromise (*IOC*). A mix of expressions (simple, complex, or both), usually grouped together for the purpose of describing a single piece of malicious activity [7].

Figure 2 shows the common *IOC* structure.

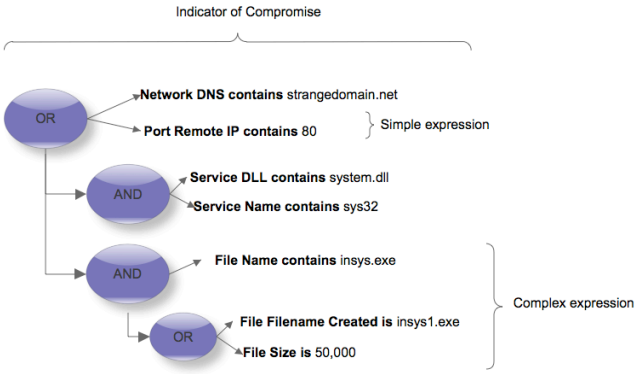

Figure 2. IOC structure.

A. IOC Editor Logic

An *IOC* can be represented by expressions on a logic tree. The logic tree starts out with a top-level "OR" structure. When expressions are added, an *IOC* will be triggered as long as one of the expressions describes a true circumstance. Sometimes an *IOC* will consist of a collection of simple expressions listed in the top-level "OR" structure without the need of a more complex logic tree [7], as shown in Figure 3.

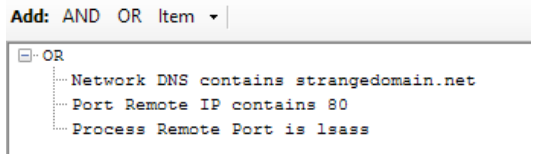

Figure 3. IOC by Simple logic.

When required, logic branches can be built with "AND" and "OR" substructures to form complex expressions. Each "AND" and "OR" applies to the branches only in its substructure [7], as is shown in Figure 4.

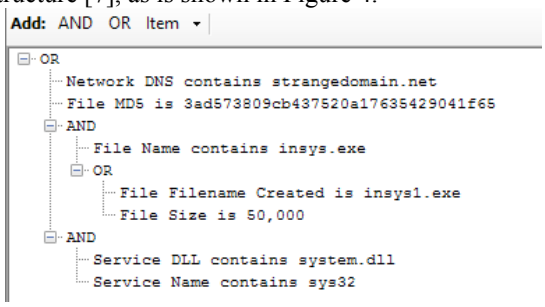

Figure 4. IOC by Logic branch.

OpenIOC is an open source framework developed by Mandiant® for sharing threat intelligence [12]. It can be used to improve the reliability and repeatability of forensic analysis, to support the investigations of incidents or suspicious malicious activity in the IT operations of an organization.

V. DEVELOPMENT

This section describes the steps followed to determine some *IOCs* using *Windows event logs* that allow early detection of malicious activity.

Using the classification made by Microsoft [4], only eight out of nine of the categories were used (shown in Figure 5). These categories can be associated with general events, and used to detect malicious intrusion events on a computer system.

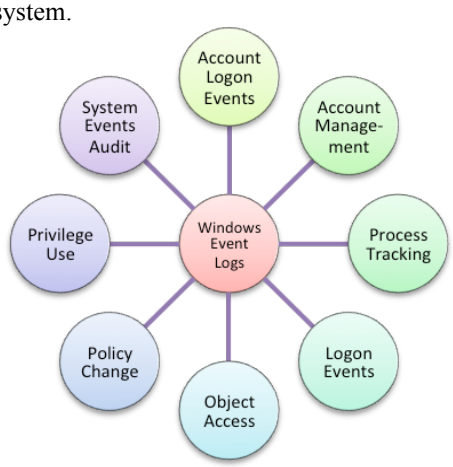

Figure 5. Categories used of *Windows event logs*.

In order to generate *IOCs* using *Windows event logs*, the main events (as described by Microsoft [4], and Anton Chuvakin et al. [8]) were reviewed, these are user-related, system access, granting of permissions or privileges, and activities or services modified in the system whilst an intrusion occurred.

The following sentences describe some of the possible actions that an external agent would perform upon an intrusion, and how this is reflected in the *Windows event logs*:

1) If an unauthorized user tries to access the system with invalid credentials, Windows event log provides the information regarding the number of attempts to access the computer system and the user credentials that are being used on a specific period of time. Examples of the information provided are: "*authentication failure*" or "*failed to log on*".

2) While the anomalous agent has valid credentials, successful login events will be also considered.

Then the anomalous agent, within the system and depending on the level of privileges the account has, will try:

3) Privileges Escalation, administrative privileges could be granted to ordinary accounts in order to create other user and/or system accounts; execute actions amongst user groups that could be reflected on events related to creation or modification of accounts and eventually taking total control over the system and be able to disable or delete existing accounts.

4) Change in *Windows Security Audit*, preventing modifications made within the system to be identified.

 ISBN: 978-1-61208-475-6

5) Create or modify files and/or system objects belonging to the installation process itself.

6) Disable or change *Windows Firewall Settings* to allow communications between malicious domains.

7) Change policies for network connections.

8) Install applications to create new services or change services already running in the system.

9) Delete audit events to prevent register activities in the system.

10) Log out session.

Having acquired all possible actions made by an intruder, we considered the odds of each event happening continuously over a determined period of time to classify them in the following activities: user, audit, services and objects (shown in Figure 6).

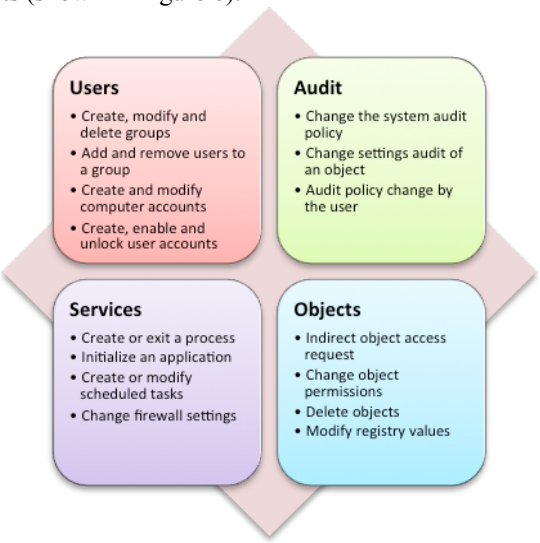

Figure 6. Classification of events by their activity.

Considering that events treated individually, might indicate a normal behavior on a computer system, it is necessary to generate a different kind of *IOC*, which could handle different events in conjunction that could determine malicious activity.

The following is a case study that represents the procedure to generate an *IOC* using *Windows event logs*:

"Peter" is a valid user (the User from now on) in a computer system with Windows® 8 OS. Peter's account has read-only user privileges.

After a security incident, the following activities were detected:

1) Two attempts to login as the User were executed.

2) User session started successfully.

3) Special privileges were assigned to User's account.

4) A new user account was created, named "Jame".

5) A global group with security-disabled settings was created.

6) An explorer process has been created.

7) An attempt to unregister a security event source was executed.

8) Jame's account was enabled.

9) The auditing settings on access-control object were changed.

10) Peter´s account session was closed.

The diagram in Figure 7 shows the activity sequence of an *IOC* considering the possible events achieved by an intruder using Peter's account.

Figure 7. Simple event sequence by malicious activity of a case study.

The next step on this procedure is to identify the event ID corresponding to the activities registered on the sequence considered malicious. Figure 8 shows the sequence of current Windows event ID for this case study.

Then an IOC could be generated to analyze the malicious activity on future occasions. It could be described with the following *simple expression* using Windows event ID:

AND (4625, 4625, 4624, 4672, 4720, 4728, 4688, 4905, 4722, 4907, 4634)

 ISBN: 978-1-61208-475-6

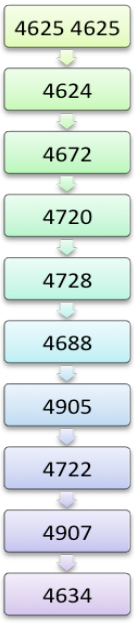

Figure 8. Event ID sequence by malicious activity of a case study.

Through the use of the IOC Editor tool, different indicators of compromise can be developed for the purpose of a timely identification of possible intrusions on Windows® OS. The *simple expression* for the case study developed using IOC Editor can be represented as shown in Figure 9.

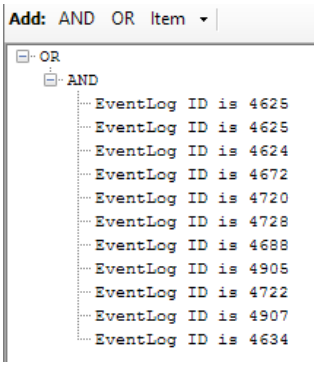

Figure 9. IOC of simple expression using IOC Editor.

Different *IOCs* could be generated using statements that represented different sequences of malicious activities previously registered. Figure 10 shows another example; taking into account the case study previously explained and considering other event sequences, a more efficiently *IOC* can be generated, reducing the false-positive behavior for a forensic analysis.

Considering the different options of events described for this new example, Figure 11 shows the current Windows event ID for current activities sequences.

To represent sequences identified in the latest example, simple expressions are combined using AND and OR logical operators, to generate an *IOC* of *complex expression*.

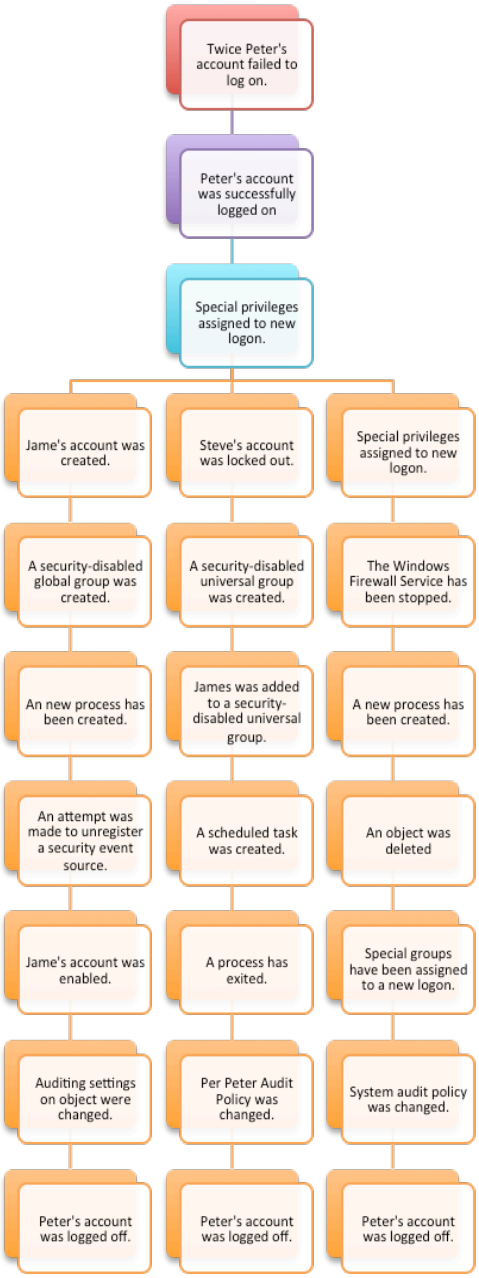

Figure 10. Branch event sequences by malicious activity.

The sentence describing the *IOC* generated using logical operators is:

AND (4625, 4625, 4624, 4672, OR (AND (4720, 4728, 4688, 4905, 4722, 4907, 4634)), OR (AND (4740, 4759, 4761, 4698, 4689, 4912, 4634)), OR (AND (4672, 5025, 4688, 4660, 4964, 4719, 4634)))

Figure 12 shows the representation of this sentence using the IOC Editor tool, where the structure of the *IOC* is displayed using simple expressions to create complex expressions using logical operators.

 ISBN: 978-1-61208-475-6

Figure 11. Branch event ID sequence by malicious activity.

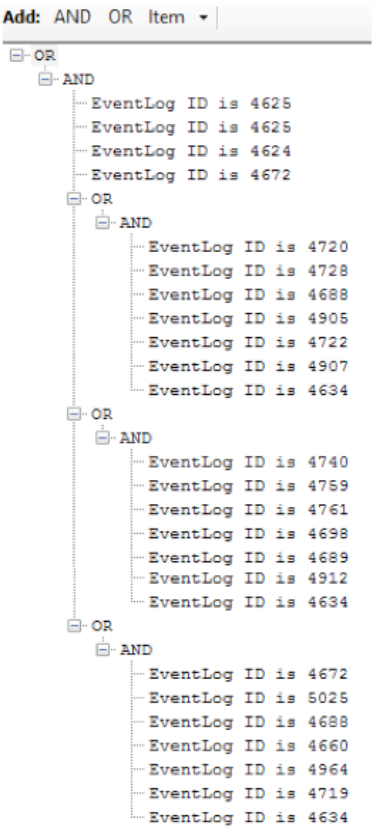

Figure 12. Indicator of Compromise of complex expression.

In order to have an overview of the activities during the intrusion on a computer system, the *IOCs* can be generated with the combination of user events, auditing, system and objects.

The general description made above shows the procedure developed to generate *IOCs* using *Windows event logs* to detect intrusions that could result in a security incident on a computer system.

VI. RESULTS

According to the previously described procedure, a list of the most representative *Windows event logs* categories considering how critical the events to generate *IOCs* of complex expressions are; to identify patrons of abnormal behavior that could result in an intrusion, were selected.

An example of a *Windows event log* for a compromised computer system, which describes the creation of the user account "Jame" using Peter´s user account, is shown in Figure 13.

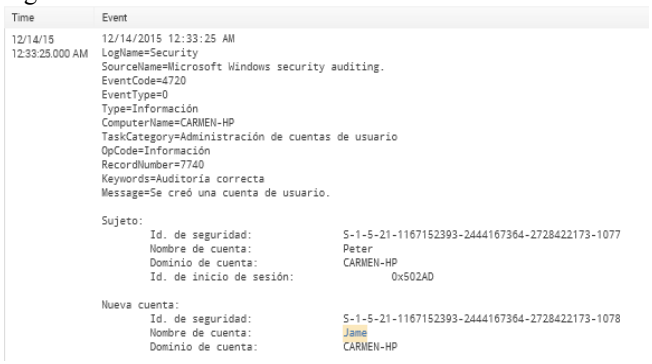

Figure 13. Windows event log structure example.

As previously described in Section V Development, a single *Windows event log* is not sufficient to detect an intrusion; events should relate to other events in the computer system to generate the *IOC* representing that malicious or anomalous activity.

To validate whether the *IOCs* generated are functional or not, these were tested in a correlational tool, for the case study presented above. A free version of the Splunk® tool was used [14], which is a *Security Information and Event Management Software* [9] that allows us to detect intrusion events in real time.

Figure 14 shows an example of an intrusion detection using a generated *IOC*.

In a sampling made on thirty computer systems over an hour, 9996 events were registered, about 50% of these belonging to the categories of User Account Management, Logon and Logoff.

Other registered events that represent 1% are:

- 4735 A security-enabled local group was changed
- 4625 An account failed to log on
- 4738 A user account was changed
- 4634 An account was logged off and
- 4732 A member was added to a security-enabled local group

 ISBN: 978-1-61208-475-6

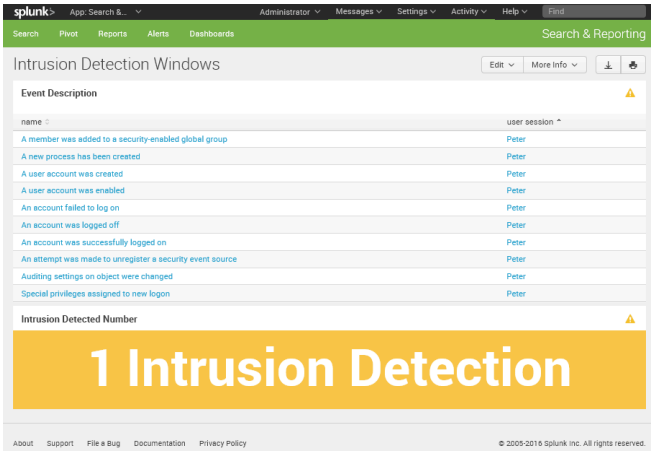

Figure 14. Test of intrusion detection using a IOC.

These events are combined with others of higher occurrence or lower occurrence. This helps generate IOCs of simple or complex expressions, which can reduced by more than 80% the rate of false positives.

There are now about 50 *IOCs* to detect intrusion on computers running Windows® 7 OS an up; Table IX shows examples of some of them. They come mainly from forensic analysis.

TABLE IX. EXAMPLES OF IOCS FOR WINDOWS OS.

ID IOC	Logic Description of Events used
1	AND [4618, 4912, OR (4907, 4660,4670,4691), OR (4964, 4767, 4760, 4758, 4757, 4753, 4750, 4743, 4740), OR (5025, 5034, 4950, 4949,)]
2	AND [4649, 4912, 4618, OR (4907, 4660,4670,4691), OR (4964, 4767, 4760, 4758, 4757, 4753, 4750, 4743, 4740), OR (5025, 5034, 4950, 4949,)]
3	AND [4912, 4618, OR (4907, 4660,4670,4691), OR (4964, 4767, 4760, 4758, 4757, 4753, 4750, 4743, 4740), OR (5025, 5034, 4950, 4949,)]
4	AND [4649, 4912, 4618]
5	AND [4618, 4912, 4618]
6	AND [6273, 4618, OR (4907, 4660,4670,4691), OR (4964, 4767, 4760, 4758, 4757, 4753, 4750, 4743, 4740), OR (5025, 5034, 4950, 4949,)]
7	AND [4719, 4618, OR (4907, 4660,4670,4691), OR (4964, 4767, 4760, 4758, 4757, 4753, 4750, 4743, 4740), OR (5025, 5034, 4950, 4949,)]

Most of the generated IOCs are complex expressions and involve more than 15 events of different categories each. 85% of simple expressions can be used to generate different IOCs of complex expressions, which represent different ways that an attack vector may consist of.

The most repeated IOCs events are sometimes considered as non-critical but associated with other events may represent an intrusion or malicious activity.

VII. CONCLUSION AND FUTURE WORK

To summarize, *IOCs* can be generated by the combination and sequence of different *Windows event logs*, describing a possible malicious activity on a computer system, adding them to the techniques known to generate *IOCs*, which provides to *CIRTs*, another way to detect an intrusion in a computer systems.

The use of *IOCs* with *Windows event logs* improves *IR* and forensic analysis processes, allowing to know the activities of an intruder on a computer system, and managing to make proper actions to prevent future malicious activities with the same attack vector in the system.

As future work, we propose to analyze what other events can be added to improve the *IOCs* generated and perform the same procedure on other OS such as Linux.

ACKNOWLEDGMENT

Thanks to Instituto Politécnico Nacional for the support granted during the development of this research.

REFERENCES

[1] <http://searchsecurity.techtarget.com/definition/Indicators-of-Compromise-IOC> 2015.09.23

[2] P. Kral, "The Incident Handlers Handbook," SANS Institute InfoSec Reading Room, 2011.

[3] T. Sager, "Killing Advanced Threats in Their Tracks: An Intelligent Approach to Attack Prevention," SANS Institute InfoSec Reading Room, 2014.

[4] <https://technet.microsoft.com/en-us/library/dn487458.aspx> 2015.11.07

[5] H. Lock, "Using IOC (Indicators of Compromise) in Malware Forensics," SANS Institute InfoSec Reading Room, 2013.

[6] <https://www.auscert.org.au/render.html?it=4323&template=1> 2015.12.16

[7] <https://www.fireeye.com/content/dam/fireeye-www/services/freeware/ug-ioc-editor.pdf> 2015.09.25

[8] A. Chuvakin and L. Zeltser, "Critical Log Review Checklist For Security Incidents," Cheat sheet version 1.0, SANS.

[9] K. Kent and M. Souppaya, "Guide to Computer Security Log Management, NIST Special Publication 800-92," Recommendations of the National Institute of Standards and Technology, USA 2006.

[10] <http://searchwindowsserver.techtarget.com/definition/Windows-event-log> 2016.01.04

[11] <https://technet.microsoft.com/en-us/library/cc722385%28v=ws.10%29.aspx> 2015.11.10

[12] <http://openioc.org/resources/An_Introduction_to_OpenIOC.pdf> 2015.12.05

[13] <https://technet.microsoft.com/en-us/library/dn535498.aspx> 2015.11.12

[14] <http://www.splunk.com/> 2015.10.10

 ISBN: 978-1-61208-475-6